Iterative algorithm for weighted total least squares adjustment

S. Jazaeri¹,², A. R. Amiri-Simkooei³,⁴ and M. A. Sharifi¹

In this contribution, an iterative algorithm is developed for parameter estimation in a nonlinear measurement error model $y - e = (A - E_A)x$, which is based on the complete description of the variance–covariance matrices of the observation errors $e$ and of the coefficient matrix errors $E_A$ without any restriction, e.g. in the case that there are correlations among observations. This paper derives the weighted total least squares solution without applying Lagrange multipliers in a straightforward manner. The algorithm is simple in the concept, easy in the implementation, and fast in the convergence. The final exact solution can be achieved through iteration. Based on the similarity between the proposed algorithm and the ordinary least squares method, the estimate for the covariance matrix of the unknown parameters can be analogously computed by using the error propagation law. The efficacy of the proposed WTLS algorithm is demonstrated by solving three WTLS problems, i.e. a linear regression model, a planar similarity transformation and two-dimensional affine transformation in the case of diagonal and fully populated covariance matrices in both start and transformed coordinate systems.

Keywords: Least squares, Weighted total least squares (WTLS), Weight matrix, Straight line fit, Similarity transformation, Affine transformation

Introduction

Consider the design matrix $A \in \mathbb{R}^{n \times m}$ with $n > m$ and an observation vector $y \in \mathbb{R}^n$ in the linear model $y = Ax$. In the standard least squares (LS) problem, only the observation vector is contaminated by random errors and the least squares solution $\hat{s}_0$ is obtained using the following optimisation problem

$$e^T Q_e^{-1} e = \min$$

Subject to $y - e = Ax$

where $e$ and $Q_e$ are the $n \times 1$ vector of random observation errors and the $n \times n$ variance–covariance (VC) matrix of observations respectively.

Total least squares (TLS) is a technique that solves the LS problem for an errors-in-variables (EIV) model in which both the observation vector and the design matrix are contaminated by random errors, without linearisation. Although the terminology ‘total least squares’ appeared in the field of numerical analysis, only recently are the weights and the least squares solution $\hat{s}_0$ is obtained using the following optimisation problem

$$e^T Q_e^{-1} e = \min$$

Subject to $y - e = Ax$

where $e$ and $Q_e$ are the $n \times 1$ vector of random observation errors and the $n \times n$ variance–covariance (VC) matrix of observations respectively.

In this case its inverse multiplied by the unknown variance factor is known as the weight matrix. The ordinary TLS solution selects identity weight matrices for the observation vector and the design matrix which are supposedly uncorrelated. However, the so called weighted TLS (WTLS) approach is a generalisation of the ordinary TLS approach where both the observation vector and the design matrix have different weight matrices and may ever be correlated. In the absence of such correlation between $y$ and $A$, the weighted total least squares (WTLS) problem seeks to solve the following optimization problem [20]

$$e^T Q_e^{-1} e + vec(E_A)^T Q_E^{-1} vec(E_A) = \min$$

subject to $y - e = (A - E_A)x$

with $E_A$ and $Q_E$ as $n \times m$ random error matrix and as $nm \times nm$ variance–covariance matrix of the corresponding elements of the vectorised design matrix $A$ respectively. The symbol ‘vec’ denotes the operator that converts a matrix to a column vector by stacking the columns, one column underneath the other. Special structure of the variance–covariance matrix $Q_E$ results in special WTLS problems. Special optimisation methods for the element-wise weighted total least squares problem have been proposed by van Huffel and Vandewalle [24], de Moor [3], Wentzell et al. [25], Premoli and Rastello [16], Markovsky et al. [12] and Manton et al. [10] among others.

¹Department of Surveying and Geomatics Engineering, College of Engineering, University of Tehran, Tehran, Iran
²Department of Surveying Engineering, National Iranian South Oil Company (NISOC), Ahwaz, Iran
³School of Geodesy, Department of Surveying Engineering, Faculty of Engineering, University of Isfahan, 81746-73441 Isfahan, Iran
⁴Chair Acoustic, Faculty of Aerospace Engineering, Delft University of Technology, Kruisweg 1, 2629 HS Delft, The Netherlands

Corresponding author, email jazayeri@ut.ac.ir

© 2014 Survey Review Ltd.
Received 30 August 2012; accepted 26 May 2013
Schaffrin and Wieser [20] are the first who formulated WTLS adjustment by following the geodetic tradition using fairly general non-diagonal variance–covariance matrices and introduced it to the Geodetic science community. This study was then followed by related research, e.g. in [15] Neitzel showed that the TLS solution within an errors-in-variables (EIV) model can also be identified as a special case of the method of least-squares within an iteratively linearised Gauss–Helmert model (GHM) where weight matrices can then be introduced without any limitations. Amiri-Simkooei and Jazaeri [2] have recently formulated the WTLS problem based on the standard least-squares theory in which a linearly structured Gauss-Markov model (GMM) is iteratively solved instead of a nonlinear GHM. The algorithm takes the complete structure for the covariance matrix of the coefficient matrix into consideration. Shen et al. [23] proposed an iterative method to perform a WTLS adjustment for an EIV model based on the Gauss–Newton approach of non-linear weighted least squares adjustment. Xu et al. [26] formulated WTLS as a nonlinear adjustment model without constraints and further extended it to a partial EIV model.

The main motivation of this paper is to derive a novel algorithm for solving the WTLS problem by considering the complete and perfect description of the variance–covariance matrices without any limitations and without Lagrange multipliers. This algorithm is simple in the concept and easy to implement. Another contribution of the paper is the covariance matrix derivation for the estimated parameters. The remainder of this paper is organized as follows: In the next section an iterative algorithm to solve the WTLS problem is given in detail. In the third section, the proposed WTLS algorithm is applied to the example of a straight line fit problem, to a 2D planar similarity transformation and to an affine transformation. Conclusions are presented in the fourth section.

### Iterative algorithm for solving general WTLS problem

In this paper we consider the optimisation problem (2), i.e. the WTLS problem without correlation between $y$ and $A$. Let $\mathbf{y} = E(y) = y - e$ and $\mathbf{f} = E(A) = A - E_A$ be the true but unknown observation vector and design matrix respectively. The observation equation in (2) will obviously fulfill $\mathbf{y} = \mathbf{Ax}$, in which all of the terms are unknown. In practice one has to be satisfied with a least squares estimate for each of the above terms, namely $^\wedge \mathbf{y} = ^\wedge \mathbf{Ax}$, or for notation convenience as $^\wedge \mathbf{y} = ^\wedge \mathbf{A}x$, where the symbol is used to indicate estimated parameters. In this section we derive an iterative algorithm to solve this equation. The cost function of the WTLS problem in equation (2) is then given by

$$
\Phi(\mathbf{Ax}, x) = e^T Q_e^{-1} e + \text{vec}(E_A^T) Q_e^{-1} \text{vec}(E_A)$$

$$= (y - E(y))^T Q_e^{-1} (y - E(y)) + \text{vec}(A - A)^T Q_e^{-1} \text{vec}(A - A)$$

$$= (y - Ax)^T Q_e^{-1} (y - Ax) + \text{vec}(A - A)^T Q_e^{-1} \text{vec}(A - A)$$

(3)

which is to be minimised over $\mathbf{A}$ and $x$. To obtain the unknown $\mathbf{A}$ and $x$ quantities, namely the expected design matrix $E\{A\} = \mathbf{A}$ and the parameter vector $x$, we take the partial derivative of equation (3) with respect to vec($A$) and $x$ and set the results equal to zero

$$\frac{\partial \Phi}{\partial x} = 0, \quad \frac{\partial \Phi}{\partial \text{vec}(A)} = 0$$

(4)

Equation $\frac{\partial \Phi}{\partial x} = 0$ results in

$$\frac{\partial}{\partial x} [(y - Ax)^T Q_e^{-1} (y - Ax)]$$

$$= -2y^T Q_e^{-1} 2Ax + 2x^T A^T Q_e^{-1} A = 0$$

(5)

which yields the total least squares solution as

$$\hat{x} = (\hat{A}^T Q_e^{-1} \hat{A})^{-1} \hat{A}^T Q_e^{-1} y$$

(6)

which resembles the standard least-squares solution except that $\hat{A} = ^\wedge \mathbf{A}$ is now a random matrix still to be determined.

To exploit the second equation of (4), the cost function (3) is rewritten as

$$\Phi(E\{A\}, x) = (y - I_e \mathbf{Ax})^T Q_e^{-1} (y - I_e \mathbf{Ax}) + \text{vec}(A - \mathbf{A})^T Q_e^{-1} \text{vec}(A - \mathbf{A})$$

This, with the following property of the vec operator

$$\text{vec}(ABC) = (C^T \otimes A)\text{vec}(B)$$

where $\otimes$ is the Kronecker product operator of two matrices, yields

$$\Phi(\mathbf{Ax}, x) = (y - (x^T \otimes I_e) \text{vec}(\mathbf{A}))^T Q_e^{-1} (y - (x^T \otimes I_e) \text{vec}(\mathbf{A})) + \text{vec}(A - \mathbf{A})^T Q_e^{-1} \text{vec}(A - \mathbf{A})$$

(8)

where $I_e$ denotes an $n \times n$ identity matrix. The partial derivative of equation (8) with respect to vec($\mathbf{A}$), namely

$$\frac{\partial \Phi}{\partial \text{vec}(\mathbf{A})} = 0,$$

leads to

$$[Q_e^{-1} + (\hat{X} \otimes I_e) Q_e^{-1} (\hat{X}^T \otimes I_e)] \text{vec}(\hat{A}) - [y^T Q_e^{-1} (\hat{X}^T \otimes I_e)] - [\text{vec}(A)^T Q_e^{-1}] = 0$$

(9)

For the sake of brevity, in the following, we denote $\hat{X} = (\hat{X} \otimes I_e)$ and $\hat{X}^T = (\hat{X}^T \otimes I_e)$. From equation (9), vec($\hat{A}$) can be obtained as

$$\text{vec}(\hat{A}) = (Q_e^{-1} + \hat{X} Q_e^{-1} \hat{X}^T)^{-1} [\hat{X} Q_e^{-1} y + Q_e^{-1} \text{vec}(A)]$$

(10)

We note that there might exist repeated elements in positive or negative form and/or fixed elements in the coefficient matrix $\mathbf{A}$. This means that, when constructing $Q_e$, one should use zero variances corresponding to the fixed elements, and positive or negative entries for the perfect covariances of the repeated elements in positive or negative form in the matrix $\mathbf{A}$, respectively. To construct the variance–covariance matrix $Q_e$, we use the linear error propagation law [13]. Owing to the fixed
elements that have zero variances and covariances, and the repeated elements (either positive or negative) that have the same values for their variances and covariances (correlation coefficient of +1 or −1), $Q_e$ may suffer from two kinds of rank deficiencies. This latter issue is called duplication of the elements. Therefore the inverse of the variance–covariance matrix $Q_e$ may not necessarily exist. We then have to rewrite equation (10) based on the variance–covariance matrix $Q_e$, rather than the weight matrix $Q_{e}^{-1}$. To achieve this goal, we use the following identity for the inverse of a matrix $[9]
\begin{equation}
(A^{-1} - UD^{-1}V)^{-1} = A + AU(D - VAU)^{-1}VA
\end{equation}

(11)

Considering $A^{-1} = Q_{e}^{-1}$, $U = (\hat{x} \otimes I_n)$, $D = Q_e$ and $V = (\hat{x}^T \otimes I_n) = \hat{x}^T$, one has

\begin{equation}
(Q_{e}^{-1} + \hat{x}Q_{e}^{-1}\hat{x}^T)^{-1} = Q_e - Q_e\hat{x}(Q_e + \hat{x}^TQ_e\hat{x})^{-1}\hat{x}^TQ_e
\end{equation}

(12)

Taking equation (12), the $\text{vec}(\hat{A})$ in equation (10) obtains the form

\begin{equation}
\text{vec}(\hat{A}) = \text{vec}(A) + Q_e\hat{x}\left\{Q_{e}^{-1}y - (Q_e + \hat{x}^TQ_e\hat{x})^{-1}\hat{x}^T\left[Q_e\hat{x}Q_{e}^{-1}y + \text{vec}(A)\right]\right\}
\end{equation}

\begin{equation}
= \text{vec}(A) + Q_e\hat{x}(Q_e + \hat{x}^TQ_e\hat{x})^{-1}\left\{Q_{e}^{-1}y - \hat{x}^TQ_e\hat{x}Q_{e}^{-1}y - \hat{x}^T\text{vec}(A)\right\}
\end{equation}

(13)

This immediately yields

\begin{equation}
\text{vec}(\hat{A}) = \text{vec}(A) + Q_e\hat{x}(Q_e + \hat{x}^TQ_e\hat{x})^{-1}(y - A\hat{\lambda})
\end{equation}

\begin{equation}
= \text{vec}(A) - \text{vec}(\hat{E}_A)
\end{equation}

(14)

which is based on the variance–covariance of random error design matrix ($Q_e$) rather than its inverse $Q_{e}^{-1}$. Equation (14) coincides with the system of Fang [4] after setting $Q_1 = Q_e + \hat{x}^TQ_e\hat{x}$, $\hat{\lambda} = Q_{e}^{-1}(y - A\hat{\lambda})$ and $\text{vec}(\hat{E}_A) = -Q_e(\hat{x} \otimes \hat{\lambda})$.

After estimating the unknown parameters ($\hat{\lambda}$) and the expected coefficient matrix ($\hat{A}$), the residual vector of observations ($\hat{\varepsilon}$) and the corresponding residual matrix ($\hat{E}$) can be determined. The vectorised residual matrix belonging to the coefficient matrix, reads $\text{vec}(\hat{E}_A) = \text{vec}(A) - \text{vec}(\hat{A})$. This with equation (14) gives

\begin{equation}
\text{vec}(\hat{E}_A) = -Q_e\hat{x}(Q_e + \hat{x}^TQ_e\hat{x})^{-1}(y - A\hat{\lambda})
\end{equation}

(15)

in agreement with Fang [4]. The estimated residual vector of the observation vector reads

\begin{equation}
\hat{\varepsilon} = y - \hat{\varepsilon} = y - \hat{\lambda} \hat{\lambda} = y - A\hat{\lambda} + \hat{E}_A \hat{\lambda} = y - A\hat{\lambda} + \hat{x}^T\text{vec}(\hat{E}_A)
\end{equation}

This, with equation (15), yields

\begin{equation}
\hat{\varepsilon} = (I - \hat{x}^TQ_e\hat{x})(Q_e + \hat{x}^TQ_e\hat{x})^{-1}(y - A\hat{\lambda})
\end{equation}

which can be simplified to

\begin{equation}
\hat{\varepsilon} = Q_{e}^{-1}(Q_e + \hat{x}^TQ_e\hat{x})^{-1}(y - A\hat{\lambda})
\end{equation}

(16)

Note that with equation (15), equation (6) can be further modified to

\begin{equation}
\begin{bmatrix}
Q_e \\
\hat{A}^T
\end{bmatrix}
\begin{bmatrix}
\hat{\lambda} \\
\hat{\lambda}
\end{bmatrix}
= \begin{bmatrix}
y \\
0
\end{bmatrix}
\end{equation}

(17)

So that eventually the equivalent system by Fang [4] is obtained

\begin{equation}
\begin{bmatrix}
Q_e \\
\hat{A}^T
\end{bmatrix}
\begin{bmatrix}
\hat{\lambda} \\
\hat{\lambda}
\end{bmatrix}
= \begin{bmatrix}
y + \hat{x}^TQ_e\hat{x} - \hat{\lambda} \\
0
\end{bmatrix}
\end{equation}

(18)

Owing to the identity

\begin{equation}
\hat{\lambda} = Q_e^{-1}(y - A\hat{\lambda} + \hat{E}_A \hat{\lambda})
\end{equation}

\begin{equation}
= Q_e^{-1}(Q_e + \hat{x}^TQ_e\hat{x} - \hat{x}^TQ_e\hat{x})Q_{e}^{-1}(y - A\hat{\lambda})
\end{equation}

\begin{equation}
= Q_{e}^{-1}(y - A\hat{\lambda})
\end{equation}

In the above derivation, we assumed that the variance–covariance matrices (i.e. all variance and covariance elements) are known a priori. In many engineering applications, however, the variance–covariance matrices are only known to a certain degree. In practice, the true scale of the variance–covariance matrices is assumed to be unknown and thus must be estimated. This leads to having the correct parameters along with their covariance matrix. In the simplest form, one may consider one scale factor to modify all variances and covariances the so-called variance component. Therefore, the variance–covariance matrices of the observations, i.e. $Q_e$ and $Q_e$ must be modified as $\sigma^2_0Q_e$ and $\sigma^2_0Q_e$ respectively, where $\sigma^2_0$ indicates just one variance component to rescale the original covariance matrices.

In the ordinary LS adjustment, Hamilton [8] showed that the variance component is unbiasedly estimated by

\begin{equation}
\hat{\sigma}_{0}^2 = \frac{\hat{\varepsilon}^T P \hat{\varepsilon}}{n - m}
\end{equation}

(19)

where $\hat{\varepsilon}$ and $P$ are the residual vector and weight matrix of observations respectively. Considering

\begin{equation}
\hat{\varepsilon} = \begin{bmatrix}
\hat{\varepsilon} \\
\text{vec}(\hat{E})
\end{bmatrix}
\end{equation}

and

\begin{equation}
P = \begin{bmatrix}
Q_e & 0 \\
0 & Q_e
\end{bmatrix}
\end{equation}

the variance component is estimated as follows in the case of WTLS adjustment

\begin{equation}
\hat{\sigma}_{0}^2 = \frac{\hat{\varepsilon}^T P \hat{\varepsilon} + \text{vec}(\hat{E}_A)^T Q_e^{-1} \text{vec}(\hat{E}_A)}{n - m}
\end{equation}

(20)
Substitution of $\hat{\lambda}$ from equation (16) and $\vec{E}_A$ from equation (15) into equation (20), yields

$$\hat{\lambda} = \frac{(y-A\hat{\lambda})^T(Q_e+\hat{\lambda}^TQ_e\hat{\lambda})^{-1}(y-A\hat{\lambda})}{n-m}$$

(21)

in agreement with Amiri-Simkooei and Jazaeri [2] and Fang [4].

Our numerical evaluations show that the WTLS estimate in equation (6) is rather of low convergence rate. We then have to further reformulate the estimate $\hat{\lambda}$ expressed in equation (6) such that the convergence rate increases. At the point of convergence, one has

$$y = \hat{\lambda} + \tilde{y} = \hat{\lambda} + \hat{\lambda}$$

(22)

which with equation (16) yields

$$y = A\hat{\lambda} + Q_e(x_e + \hat{\lambda}^TQ_e\hat{\lambda})^{-1}(y-A\hat{\lambda})$$

(23)

Substitution of equation (23) into equation (6) gives

$$\hat{\lambda} = (A^TQ_e^{-1}A)^{-1}A^TQ_e^{-1}(y-A\hat{\lambda})^{-1}$$

(24)

which simplifies to

$$\hat{\lambda}^T(Q_e+\hat{\lambda}^TQ_e\hat{\lambda})^{-1}(y-A\hat{\lambda}) = 0$$

(25)

again in agreement with Amiri-Simkooei and Jazaeri [2] and Fang [4].

Substituting for $A$ from $A = \hat{E}_A + \hat{\lambda}$ into equation (25) reformulates the least squares estimate $\hat{\lambda}$ as

$$\hat{\lambda} = \hat{\lambda}^T(Q_e+\hat{\lambda}^TQ_e\hat{\lambda})^{-1}y$$

(26)

which is the original formula.

Although apparently formula (26) is similar to the Shen’s derivation [24], we note that our presented algorithm is different from the Shen’s algorithm. Equation (26) is obtained without linearisation and without applying the Lagrange multipliers. Numerical results indicate that the presented algorithm converged faster than Shen’s algorithm (see next section). Application of the variance-covariance propagation law to equation (26) approximates the variance-covariance matrix of the WTLS solution, $C_\xi$ as (note that $D(y-E_A\hat{\lambda}) = Q_e + \lambda^TQ_e\lambda$)

$$C_\xi = E^{\lambda}(Q_e+\hat{\lambda}^TQ_e\hat{\lambda})^{-1}$$

(27)

This approximation is due to the intrinsic nonlinearity of the problem in which the unknown vector $x$ has to be estimated by $\hat{\lambda}$. The above formula has already been presented by Amiri-Simkooei and Jazaeri [2].

Finally, with the above derivations, Algorithm 1 provides the steps involved to solve a WTLS problem.

In this section an efficient algorithm was developed to solve the WTLS problem in the same way as Amiri-Simkooei and Jazaeri [2] and Fang [4]. Because there exist repeated and/or fixed elements in the design matrix $A$, the corresponding covariance matrix, i.e. $Q_e$, derived by the error propagation law, is likely to be singular. Therefore, the derived algorithm has been rewritten based on the covariance matrix $Q_e$ instead of its inverse $Q_e^{-1}$. In the following section, three numerical examples are presented to evaluate the efficiency and the high convergence rate of the proposed algorithm as described above.

**Numerical results: three applications**

**Example 1.** As a first example, we consider a straight line fit problem where both variables are observed and thus EIV is involved. It is a linear regression where the goal is to find the intercept and slope regression parameters $\xi_1$ and $\xi_2$ of the regression line

$$y_i - e_\xi = \xi_2(x_i - e_\lambda) + \xi_1$$

(28)

using the proposed WTLS algorithm. In this section to prove the high efficiency of the proposed algorithm to solve all WTLS problems along with estimates of the variance-covariance matrix of the estimated parameters, as the first example, we consider the straight line fitting problem. The observed data and their corresponding weights, provided by Neri et al. (14), are listed in Table 1.

Equation (28) can be written in matrix notation as follows

$$y_i - e_\xi = \begin{bmatrix} x_i - e_\lambda \end{bmatrix} \begin{bmatrix} \xi_2 \\ \xi_1 \end{bmatrix}$$

(29)

<table>
<thead>
<tr>
<th>Point no.</th>
<th>$y$</th>
<th>$x$</th>
<th>$W_x$</th>
<th>$W_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.9</td>
<td>0.0</td>
<td>1.0</td>
<td>1000.0</td>
</tr>
<tr>
<td>2</td>
<td>5.4</td>
<td>0.9</td>
<td>1.8</td>
<td>1000.0</td>
</tr>
<tr>
<td>3</td>
<td>4.4</td>
<td>1.8</td>
<td>4.0</td>
<td>500.0</td>
</tr>
<tr>
<td>4</td>
<td>4.6</td>
<td>2.6</td>
<td>8.0</td>
<td>800.0</td>
</tr>
<tr>
<td>5</td>
<td>3.5</td>
<td>3.3</td>
<td>20.0</td>
<td>200.0</td>
</tr>
<tr>
<td>6</td>
<td>3.7</td>
<td>4.4</td>
<td>20.0</td>
<td>800.0</td>
</tr>
<tr>
<td>7</td>
<td>2.8</td>
<td>5.2</td>
<td>7.0</td>
<td>60.0</td>
</tr>
<tr>
<td>8</td>
<td>2.8</td>
<td>6.1</td>
<td>70.0</td>
<td>20.0</td>
</tr>
<tr>
<td>9</td>
<td>2.4</td>
<td>6.5</td>
<td>100.0</td>
<td>1.8</td>
</tr>
<tr>
<td>10</td>
<td>1.5</td>
<td>7.4</td>
<td>500.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**Table 1** Observed points and corresponding weights according to Neri et al. (1989)
Table 2 Results of straight line fit to observed data of Table 1, including their variances and covariances according to equation (25)

<table>
<thead>
<tr>
<th>Parameter estimate</th>
<th>Exact solution (Neri et al.)</th>
<th>WTLS (this paper)</th>
<th>$\sigma^2$</th>
<th>$\sigma^2_{\delta x^2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_1$</td>
<td>5.47991022</td>
<td>5.4799102240</td>
<td>0.0049872225</td>
<td>-0.0244336291</td>
</tr>
<tr>
<td>$\theta_2$</td>
<td>-0.480533407</td>
<td>-0.480533407</td>
<td>0.129058064</td>
<td></td>
</tr>
</tbody>
</table>

The design matrix $A$ and the observation vector $y$ are

$$
A = \begin{bmatrix}
x_1 & 1 & \vdots & \vdots & \vdots & x_{10} & 1 \\
y_1 & y_2 & \vdots & \vdots & \vdots & y_{10} & \end{bmatrix} (30)
$$

The covariance matrix of the random observation error $e$, $Q_e$, and the covariance matrix of $\text{vec}(E)$, $Q_E$, are given as

$$
Q_e = \text{Diag}(W_{x1}^{-1}, W_{x2}^{-1}, \ldots, W_{x10}^{-1})
$$

$$
Q_E = \text{Diag}(W_{x1}^{-1}W_{x2}^{-1}, \ldots, W_{x10}^{-1}0,0,\ldots,0)
$$

where ‘$\text{Diag}$’ is an operator that converts a vector into a diagonal matrix. We note that only the first column of the design matrix $A$ is associated with random errors, while all the values in the second column are fixed. Therefore, all elements associated with these fixed elements of $Q_E$ have zero variances. This clearly makes $Q_E$ rank-deficient. A threshold of $\epsilon=10^{-10}$ is chosen here. The regression parameters have been computed using the proposed algorithm, which converges after 7 iterations. Shen’s algorithm converges after 13 iterations. The results are presented in Table 2. The results indicate that the estimated line parameters are the same and coincide with the exact solutions as reported by Neri et al. (1989) and Schaffrin and Wieser (19); see also Schaffrin et al. (22) and Schaffrin et al. (17) for various ordinary TLS solution for the straight-line parameters.

The value of the objective function, the estimate of the variance component (equation (19)), and the corresponding residuals as obtained from the proposed algorithm can be found in Table 3.

Example 2. One of the common applications in geodesy, photogrammetry, mapping, engineering surveying, computer vision and geographic information science (GIS) employs the same form of transformations. Specific TLS solutions for coordinate transformation have been presented earlier in geodetic literature for example by Felus and Schaffrin ([5]), Akylma [([1]), Schaffrin ([18]) and Schaffrin and Felus ([19]) and Schaffrin et al. ([22]). In the second example we consider the similarity transformation in the 2D plane. Consider the start coordinates and the transformed coordinates of an arbitrary point $K_i$, to be denoted by $(x_i, y_i)$ and $(X_i, Y_i)$ respectively. Then, the planar similarity transformation can be described in geometric terms as

$$
\begin{bmatrix}
x_i \\
y_i
\end{bmatrix}
\approx
s\begin{bmatrix}
\cos \alpha - \sin \alpha \\
\sin \alpha \cos \alpha
\end{bmatrix}
\begin{bmatrix}
x_i \\
y_i
\end{bmatrix}
+ \begin{bmatrix}
\xi \\
\eta
\end{bmatrix}
$$

where $s$ is the scale factor, $\alpha$ is the rotation angle, and $\xi$ and $\eta$ are the two shift parameters of the origin.

By multiplying the corresponding expressions, equation (32) will have the following form per point

$$
X_i \approx (s \cos \alpha)x_i - (s \sin \alpha)y_i + \xi
$$

$$
Y_i \approx (s \sin \alpha)x_i + (s \cos \alpha)y_i + \eta
$$

Substituting the physical parameters $s$ and $\alpha$ by the synthetic parameters

$\lambda_1 : = s \cos \alpha$, $\lambda_2 : = s \sin \alpha$

results in the following linear system of equations

$$
\begin{bmatrix}
X_i \\
Y_i
\end{bmatrix}
\approx
\begin{bmatrix}
\lambda_1 & -\lambda_2 & \xi \\
\lambda_2 & \lambda_1 & \eta
\end{bmatrix}
\begin{bmatrix}
x_i \\
y_i \\
x_0
\end{bmatrix}
$$

Equation (35) may be rewritten in vector form per point as

$$
E \left( \begin{bmatrix}
X_i \\
Y_i
\end{bmatrix} \right) = E \left( \begin{bmatrix}
x_i \\
y_i \\
x_0
\end{bmatrix} \right) \begin{bmatrix}
\lambda_1 \\
\lambda_2 \\
\xi \\
\eta
\end{bmatrix}
$$

Table 3 Estimates for least squares residuals of observed $x$- and $y$-values along with value of objective function, and estimated variance component

<table>
<thead>
<tr>
<th>Point no.</th>
<th>$\hat{\delta}_x$, [m]</th>
<th>$\hat{\delta}_y$, [m]</th>
<th>$\delta$ TSSR* / $8 = 14832941492$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4199927944</td>
<td>0.0002018206</td>
<td>TSSR = 11.8663531941</td>
</tr>
<tr>
<td>2</td>
<td>0.3524233606</td>
<td>0.0003048322</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-0.2145537457</td>
<td>-0.0008248019</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.3686254336</td>
<td>0.0017713684</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-0.3852959889</td>
<td>-0.0185127412</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.3161840668</td>
<td>0.0379842518</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>-0.1426948374</td>
<td>-0.0799979092</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.1390025995</td>
<td>0.2337838747</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.0031498020</td>
<td>0.0840880607</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>-0.0036405369</td>
<td>-0.0746997931</td>
<td>TSSR = 11.8663531941</td>
</tr>
</tbody>
</table>

* TSSR: total sum of (weighted) squared residuals.
where the symbol $E$ again denotes the expectation operator. Once the four parameters $\lambda_1$, $\lambda_2$, $\xi$ and $\eta$ have been estimated, the rotation angle and scale factor can easily be derived by

$$\hat{\xi} = \arctan \left( \frac{\hat{\lambda}_2}{\hat{\lambda}_1} \right)$$  \hspace{1cm} (37)

$$\hat{s} = \left( \frac{\hat{\lambda}_2^2 + \hat{\lambda}_1^2}{2} \right)^{1/2}$$  \hspace{1cm} (38)

To check the results of the above WTLS algorithm, a transformation of weighted coordinates is considered. The coordinate estimation of the transformed system, $X_i$, $Y_i$, and in the start system, $x_i$, $y_i$, are listed in Table 4.

Diagonal weight matrices associated with the start and transformed system, denoted by $P_{xy}$ and $P_{XY}$ respectively, are given below


$$P_{XY} = \text{Diag}(9.8361|5.5357|12.7369|12.0099|10.181|11.3661|11.1475|5.8834|9.8322|7.5678)$$  \hspace{1cm} (39)

The goal is thus to find best estimates for the transformation parameters $s$, $\alpha$, $\xi$ and $\eta$, with the assumption that the coordinates in both start and transformed systems are contaminated by random errors.

To assist in the understanding of the proposed algorithm, the design matrix $A$, the covariance matrix of $\text{vec}(E)$, $Q_E$, the observation vector $y$, and its corresponding covariance matrix, $Q_y$, are given for this example by

$$A_{10 \times 4} = \begin{bmatrix} x_1 - y_1 & 1 & 0 \\ y_1 & x_1 & 0 \\ \vdots & \vdots & \vdots \\ x_5 - y_5 & 1 & 0 \\ y_5 & x_5 & 0 \end{bmatrix}$$  \hspace{1cm} (40)

$$Q_{E_{10 \times 10}} = \begin{bmatrix} \sigma_{X_1}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{Y_1}^2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{X_5}^2 \\ 0 & 0 & \cdots & \sigma_{Y_5}^2 \end{bmatrix}$$  \hspace{1cm} (41)

The corresponding WTLS residuals are presented in Table 6. The estimated covariance matrix of the estimated parameters using the proposed WTLS algorithm is included in Table 7. Both the proposed algorithm and Shen’s algorithm converge after three iterations.

Numerical examples are easy to implement and demonstrate the efficiency and high convergence rate of the proposed algorithm.

**Example 3.** In the following we consider a two-dimensional affine transformation with a full covariance matrix to show the performance of the presented method in the case of fully populated covariance matrices. The model for the 2D affine transformation (six-parameter transformation) is expressed as follows

$$X^j = c_1 x_i + c_2 y_i + c_3 x_i y_i + c_4 x_i^2 + c_5 y_i^2$$

and

$$Q_{E_{6 \times 6}} = \begin{bmatrix} \sigma_{\delta_1}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{\delta_2}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{\delta_3}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{\delta_4}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{\delta_5}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{\delta_6}^2 \end{bmatrix}$$  \hspace{1cm} (42)
where the parameters $c_1$ and $c_2$ are the shifts along the $u$- and $v$- axes, respectively. The other parameters $a_1, a_2, b_1$ and $b_2$ are related to the four physical parameters of a 2-D linear transformation, which include two scales along the $u$- and $v$- axes, one rotation, and one non-perpendicularity (or affinity) parameter. For each point $i=1, 2, ..., n$, we add two rows to the design matrix $A$ and the observation vector $y$ as follows

$$y^\sim = \begin{bmatrix} u_1 \\
 v_1 \\
 \vdots \\
 u_n \\
 v_n \end{bmatrix}, \quad A^\sim = \begin{bmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 \\
 0 & 0 & u_1 & v_1 & 1 \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 u_n & v_n & 1 & 0 & 0 \\
 0 & 0 & u_n & v_n & 1 \end{bmatrix}. \quad (44)$$

Therefore, for $n$ points, we have $2n$ observations and 6 unknown parameters. In this case, the VC matrix of the random error vector $e_A = \text{vec}E_A$ to the coefficient matrix $A$ reads

$$Q_E = \begin{bmatrix} Q_1 \\
 Q_2 \\
 Q_3 \\
 Q_4 \\
 Q_5 \\
 Q_6 \end{bmatrix} \begin{bmatrix} 0 & 0 \\
 0 & 0 \end{bmatrix}.$$

where

$$Q_1 = I_6 \otimes \begin{bmatrix} 1 & 0 \\
 0 & 0 \end{bmatrix}, \quad Q_2 = I_6 \otimes \begin{bmatrix} 0 & 1 \\
 0 & 0 \end{bmatrix}, \quad Q_3 = I_6 \otimes \begin{bmatrix} 0 & 0 \\
 0 & 0 \end{bmatrix}, \quad Q_4 = I_6 \otimes \begin{bmatrix} 0 & 0 \\
 1 & 0 \end{bmatrix}, \quad Q_5 = I_6 \otimes \begin{bmatrix} 0 & 0 \\
 0 & 1 \end{bmatrix}.$$

Table 4 Coordinate estimates in transformed and in start system

<table>
<thead>
<tr>
<th>Point no.</th>
<th>Transformed system</th>
<th>Start system</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$X_i[m]$</td>
<td>$Y_i[m]$</td>
</tr>
<tr>
<td>1</td>
<td>339.2178</td>
<td>971.2505</td>
</tr>
<tr>
<td>2</td>
<td>-778.759</td>
<td>607.9502</td>
</tr>
<tr>
<td>3</td>
<td>-778.742</td>
<td>-567.548</td>
</tr>
<tr>
<td>4</td>
<td>339.223</td>
<td>-930.773</td>
</tr>
<tr>
<td>5</td>
<td>1030.171</td>
<td>20.26966</td>
</tr>
</tbody>
</table>

Table 5 Solution from proposed WTLS algorithm

<table>
<thead>
<tr>
<th>Obtained parameters</th>
<th>Proposed algorithm solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter $\hat{\lambda}_1$</td>
<td>0.9998847501</td>
</tr>
<tr>
<td>Parameter $\hat{\lambda}_2$</td>
<td>-0.0000669326</td>
</tr>
<tr>
<td>Shifting $\hat{\beta}_1$</td>
<td>29.7700655730 m</td>
</tr>
<tr>
<td>Shifting $\hat{\beta}_2$</td>
<td>19.7171284297 m</td>
</tr>
<tr>
<td>Scale factor $\hat{\beta}_3$</td>
<td>0.9998847524</td>
</tr>
<tr>
<td>Rotation angle $\hat{\beta}_4$</td>
<td>-0.0000669403 rad</td>
</tr>
<tr>
<td>Estimated variance component $\hat{s}_2^2$</td>
<td>0.1515786581</td>
</tr>
<tr>
<td>Total sum of (weighted) squared residuals (TSSR)</td>
<td>0.9094719485</td>
</tr>
</tbody>
</table>

Table 6 Residuals obtained from proposed WTLS algorithm

<table>
<thead>
<tr>
<th>Point no.</th>
<th>Transformed system</th>
<th>Start system</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\hat{\delta}_x[m]$</td>
<td>$\hat{\delta}_y[m]$</td>
</tr>
<tr>
<td>1</td>
<td>-0.0184385870</td>
<td>-0.0190800413</td>
</tr>
<tr>
<td>2</td>
<td>-0.0506393539</td>
<td>-0.0435334581</td>
</tr>
<tr>
<td>3</td>
<td>-0.0734760840</td>
<td>-0.0458777577</td>
</tr>
<tr>
<td>4</td>
<td>-0.0530742021</td>
<td>-0.1464197113</td>
</tr>
<tr>
<td>5</td>
<td>-0.0891004598</td>
<td>-0.0093319041</td>
</tr>
</tbody>
</table>

Table 7 Covariance matrix of estimated parameters with proposed WTLS algorithm

Covariance matrix of estimated parameters

<table>
<thead>
<tr>
<th></th>
<th>0-000000000070</th>
<th>-0-00000000001</th>
<th>0-0000010143</th>
<th>-0-00000001387</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0-0000000001</td>
<td>0-00000000055</td>
<td>-0-0000000212</td>
<td>0-000000016194</td>
<td>-0-0000001387</td>
</tr>
<tr>
<td>0-0000010143</td>
<td>-0-000000212</td>
<td>0-0054406407</td>
<td>-0-00000238011</td>
<td>0-069427019</td>
</tr>
<tr>
<td>-0-0000001387</td>
<td>0-0000016194</td>
<td>-0-00000238011</td>
<td>-0-0000069427019</td>
<td></td>
</tr>
</tbody>
</table>
Twenty points are considered in the start system, ideally transformed by the parameters \(a_1 = 2, b_1 = -1, c_1 = 0, a_2 = -1, b_2 = 2\) and \(c_2 = 0\) into the coordinates in the transformed system. The errorless coordinates of the start and transformed systems are given in Table 8 and shown in Fig. 1.

The full covariance matrices of the start and transformed systems are constructed using MATLAB built-in function `randn` as \([\text{randn}(100,2)*n]^{T}\). We also have coordinates in both start and transformed system contaminated by random errors using the MATLAB built-in function in the statistics toolbox (mvnrnd.m), which allows to generate the coordinates for a given covariance matrix. The threshold \(e = 10^{-10}\) was used for termination, and the process has been repeated over 100 000 independent runs.

The algorithm presented in this contribution converges after 19.8 iterations on average. The Shen’s formulation converges after 37.87 iterations on average that indicates its difference with presented algorithm.

The histogram of the estimated parameters is given in Fig. 2.

The numerical results show a high convergence rate of the proposed algorithm in the case of fully populated variance–covariance matrices.

Conclusions

In this contribution, we have introduced a new algorithm to solve the WTLS problem. It is based on the complete description of the covariance matrices of the observations in the vectorised coefficient matrix and, of the observation vector without any restriction and without Lagrange multipliers. Three numerical examples, namely a linear regression model, a planar similarity transformation, and a two-dimensional affine transformation, were used to illustrate the WTLS algorithm. It was shown that the proposed algorithm is simple in the concept and easy in the implementation. The results indicate a high convergence rate of the algorithm. The presented method can be used as an alternative to computing the WTLS solution. Exactly, in approximation to the covariance matrix of the parameter estimates can be computed from the linear error propagation law in order to assess the precision of the WTLS estimates.

Acknowledgement

We would like to acknowledge useful comments of two anonymous reviewers that improved the presentation of this paper.
References