Journal of Mathematical Modeling
Vol. 13, No. 3, 2025, pp. 573-592. Research Article

JMM

An iterative method for solving the generalized total least squares problem

Saeed Karimi[†,‡,*], Bentelhoda Zali[‡]

[†]*Department of Mathematics, College of Sciences, Shiraz University, Shiraz 7187919556, Iran*
[‡]*Department of Mathematics, Faculty of Intelligent Systems Engineering and Data Science, Persian Gulf University, Bushehr, Iran*
Email(s): karimi@pgu.ac.ir, hoda.zali@mehr.pgu.ac.ir

Abstract. This paper introduces a novel method for solving the generalized total least squares problem, an extension of the total least squares problem. The generalized total least squares problem emerges when solving overdetermined linear systems with the multiple right-hand sides $\mathbf{AX} \approx \mathbf{B}$, where both the observation matrix $\mathbf{B}$ and the data matrix $\mathbf{A}$ contain errors. Our approach involves extending the Taylor series expansion to reformulate the generalized total least squares problem into a linear problem, allowing us to employ the tensor form of the generalized least squares algorithm for efficient computation. This technique streamlines the computational process and enhances solution accuracy. For a more detailed survey, we compare the proposed method for solving the generalized total least squares problem with one of the matrix format methods for the associated total least squares problem. Empirical results show that our method significantly improves computational efficiency and solution precision. Additionally, we demonstrate its practical application in the context of image blurring.

Keywords: Least squares problem, total least squares problem, tensor form of the generalized least squares problem, Taylor series expansion, image deblurring.
AMS Subject Classification 2010: 34A34, 65L05.

1 Introduction

The least squares (LS) method is a widely used technique in solving mathematical and statistical problems, particularly for fitting models to data and solving equations, considering the vector equation

$$\mathbf{Ax} = \mathbf{b} + \mathbf{f}.$$

*Corresponding author

Received: 2 November 2024 / Revised: 30 January 2025 / Accepted: 25 February 2025
DOI: 10.22124/jmm.2025.28866.2568

In the LS approach, the data matrix is assumed to be free from errors, with all errors being limited to the observation vector. The LS is aimed for solving the optimization problem

$$\begin{cases} \min\limits_{\mathbf{f}} \ \|\mathbf{f}\|_2, \\ s.t. \ \ \mathbf{A}\mathbf{x} = \mathbf{b} + \mathbf{f}, \end{cases}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{b}, \mathbf{f} \in \mathbb{R}^m$ $(m \geq n)$ [3, 4].

Many problems in applied statistics and error regression arise where this assumption is often unrealistic because the coefficient matrix is not constant. In such cases, a better and more general fitting technique, known as the total least squares, minimizes the errors. The total least squares problem is a well-studied topic in mathematics and statistics. It deals with finding the best-fitting line (or hyperplane in higher dimensions) that minimizes the sum of the squares of the perpendicular distances from the data points to the line. This problem arises in various applications such as data fitting, regression analysis, and signal processing. The total least squares problem has garnered significant attention in the academic community. Researchers persistently investigate and expand their knowledge about total least squares and its uses. The total least squares problem literature encompasses theory, computational methods, various field applications, and comparisons to alternative regression techniques. Researchers have analyzed the robustness of various total least squares algorithms and their properties. Now, the total least squares problem is defined as follows:

Definition 1. *[7] Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$ with $m \geq n$. Then the total least squares (TLS) problem is*

$$\begin{cases} \min\limits_{\mathbf{E},\mathbf{f}} \ \|[\mathbf{E}, \mathbf{f}]\|_F, \\ s.t. \ \ (\mathbf{A} + \mathbf{E})\mathbf{x} = \mathbf{b} + \mathbf{f}, \end{cases} \tag{1}$$

where $\mathbf{E} \in \mathbb{R}^{m\times n}$, $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{f} \in \mathbb{R}^m$.

The TLS method, introduced by Golub and Van Loan, is a powerful tool for estimating parameters in linear models with errors in both the observation vector and the data matrix [9, 10]. Golub and Van Loan proposed a numerical algorithm that uses singular value decomposition (SVD) of the augmented matrix $[\mathbf{A}, \mathbf{b}]$ which is described in the following theorem [32].

Theorem 1. *Let $\mathbf{C} = \mathbf{U}\Sigma\mathbf{V}^{\mathbf{T}}$ be the singular value decomposition of the matrix $\mathbf{C} = [\mathbf{A}, \mathbf{b}] \in \mathbb{R}^{m\times(n+1)}$, where*

$$\Sigma = diag(\sigma_1, ..., \sigma_n, \sigma_{n+1}), \quad \sigma_1 \geq \cdots \geq \sigma_n \geq \sigma_{n+1} > 0.$$

Let $\mathbf{v_{n+1}}$ be the last column of $\mathbf{V}$ and $v_{i,n+1}$ be the i-th component of $\mathbf{v_{n+1}}$. If $v_{n+1,n+1} \neq 0$ and $\sigma_n > \sigma_{n+1}$, then the TLS problem (1) has a unique solution $\mathbf{x}_{TLS}$, which is given by

$$\mathbf{x}_{TLS} = -\frac{1}{v_{n+1,n+1}}[v_{1,n+1}, \cdots, v_{n,n+1}]^T,$$

with the corresponding correction matrix $[\mathbf{E}, \mathbf{f}] = -\sigma_{n+1}u_{n+1}v_{n+1}^T$ for which $\|[\mathbf{E}, \mathbf{f}]\|_F^2 = \sigma_{n+1}^2$.

The TLS problem remains a versatile tool for modeling residual equation errors. Mathematical advances have led to the improvement of more efficient and effective algorithms and tools. Because of its dynamic nature, TLS can adapt to developments in technology and research. The use of TLS and errors-in-variables (EIV) modeling has seen a rise in popularity in recent years, thanks to the development of new algorithms based on the singular value decomposition method [9]. Total least squares have been widely used in computer vision [21], image reconstruction [8, 22], voice and audio processing [14, 18], modal and spectrum analysis [30, 31], linear system theory [5, 6], and system identification [19, 25].

Markovsky et al. [20] presented a novel rank-revealing method for addressing the total least squares problem found in considerable applications. Accurately determining the rank of the data matrix enhances the effectiveness of total least squares problem solutions. They have presented a new method of rank recognition for more accurate solutions for the total least squares problem making it a valuable tool for researchers and practitioners in various fields. Van Huffel et al. [27, 29] explored the challenges and solutions to the total least squares problem from a computational standpoint. The authors proposed methods for efficiently solving the total least squares problem. The paper underscores the significance of computational considerations in addressing the total least squares problem and provides valuable insights into the optimization and analysis of related algorithms. Golub et al. analyzed the mathematical fundamentals and features of the total least squares problem in their article. They studied the connection between the TLS problem and the least squares problems. The study enhances the comprehension of the total least squares issue and its ramifications for diverse applications, paving the way for additional investigation in this domain [10]. The issue with the proposed methods for the total least squares problem is that computing a full SVD can be expensive when the matrix $\mathbf{A}$ is large. One way to improve this is by calculating a partial SVD using Householder transformations [28] or Lanczos bidiagonalization, as introduced by Golub and Kahan [11]. One possible improvement is the employment of the Rayleigh quotient iteration [2] or the Gauss-Newton iteration [7, 26].

The problem of finding the solution $\mathbf{x}$ for the TLS problem is equivalent to the solution of the following minimization problem [10]:

$$\min_{\mathbf{x}} \eta(\mathbf{x}) = \frac{\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2}{\sqrt{1+\mathbf{x}^{\mathbf{T}}\mathbf{x}}}.$$

The Gauss-Newton method is an influential algorithm for solving nonlinear optimization problems. It is one of the proposed methods for obtaining the solution to the total least squares problem [7]. In more recent years, Bagheri et al. [1] proposed alternative Arnoldi process for ill-conditioned tensor equations with applications to image restoration. They illustrated how the tensor alternative Arnoldi process can be exploited to solve ill-posed problem. Also, Han et al. [13] presented the tensor regularized TLS and structured tensor TLS methods for solving ill-conditioned and structured tensor equations, respectively, adopting a tensor-tensor-product.

A generalized total least squares problem mathematically characterized many recent data analysis models. The Generalized Total Least Squares (GTLS) is a method for solving overdetermined sets of linear systems with multiple right-hand sides $\mathbf{AX} \approx \mathbf{B}$. It is suitable when there are errors in both the observation matrix $\mathbf{B}$ and the data matrix $\mathbf{A}$.

Golub et al. [10] introduced numerical analysis to solve generalized total least squares problems and presented an algorithm based on singular value decomposition. Their presented paper extends TLS computations to solve these problems. They described the properties of those problems and showed that the proposed generalization of the TLS criteria remains optimal for any number of observation vectors in $\mathbf{B}$. In this paper, we consider the generalized total least squares problem

$$\begin{cases} \min\limits_{\mathbf{E},\mathbf{F}} \|[\mathbf{E},\mathbf{F}]\|_F, \\ s.t. \ \ (\mathbf{A}+\mathbf{E})\mathbf{X} = \mathbf{B}+\mathbf{F}, \end{cases}$$

where $\mathbf{A},\mathbf{E} \in \mathbb{R}^{m\times n}$, $\mathbf{X} \in \mathbb{R}^{n\times s}$ and $\mathbf{B},\mathbf{F} \in \mathbb{R}^{m\times s}$.

Inspired by the Gauss-Newton method for total least squares problem, we prepose the Gauss-Newton-type method for solving the nonlinear generalized total least squares problem, namely the GN-GTLS method. In the next step, we transform the generalized total least squares problem into a linear problem by extending the Taylor series expansion. Then, the solution of the general least squares problem is obtained by solving the tensor equation.

The rest of this paper is organized as follows. In Section 2, we present some basic definitions and lemmas. The third section reviews the Gauss-Newton method for solving the total least squares problem. In Section 4, we introduce the GN-GTLS method for solving the generalized total least squares problem. In Section 5, we provide some numerical examples of the feasibility and efficiency of our proposed method. Also, as an application example, we implement the GN-GTLS method to restore gray-scale and color images. Finally, we conclude some remarks in the last section.

2 Preliminaries

In this section, we briefly introduce some fundamental concepts and the notations used throughout the paper. We use the following notations: $\mathbb{R}$ denotes the real field, and $\mathbb{N}$ represents positive integer numbers. We employ lowercase, uppercase, Greek lowercase letters and bold lowercase letters to denote scalars in $\mathbb{R}$ and vectors, respectively. Matrices are denoted by bold uppercase letters, while tensors are represented in calligraphic font. We denote $\mathbf{I}_m$ the identity matrix of size m. Also, $\mathbf{A}^j$ denotes the matrix form of $\mathscr{A}(:,:,j)$. Let $\mathbf{A} \in \mathbb{R}^{n\times m}$ and $\mathbf{B} \in \mathbb{R}^{n\times k}$. The MATLAB notation $[\mathbf{A},\mathbf{B}]$ is an augmented matrix where the columns of $\mathbf{B}$ add to the ones of $\mathbf{A}$.

A tensor is a mathematical tool widely used to describe physical properties in multidimensional systems. Tensors can be defined as vectors (first order), matrices (second order), or even more complex multidimensional objects (higher orders). The definition of tensor as an array is given below.

Definition 2. *[17] For a positive integer N, an order N tensor $\mathscr{A} = (a_{i_1\cdots i_N}) \in \mathbb{R}^{I_1\times\cdots\times I_N}$ is a multidimensional array with $\prod_{i=1}^{N} I_i$ entries in the real field $\mathbb{R}$.*

Let M,N be the positive integers, $\mathscr{A} \in \mathbb{R}^{I_1\times\cdots\times I_N\times J_1\times\cdots\times J_N}$ and $\mathscr{B} \in \mathbb{R}^{J_1\times\cdots\times J_N\times K_1\times\cdots\times K_M}$, the Einstein product of $\mathscr{A}$ and $\mathscr{B}$ is defined by operation $*_N$ via

$$(\mathscr{A} *_N \mathscr{B})_{i_1\cdots i_N k_1\cdots k_M} = \sum_{j_1,\cdots,j_N} a_{i_1\cdots i_N j_1,\cdots j_N} b_{j_1\cdots j_N k_1\cdots k_M}, \quad 1 \le j_i \le J_i, \quad i = 1,\ldots,N,$$

thus $(\mathscr{A} *_N \mathscr{B}) \in \mathbb{R}^{I_1\times\cdots\times I_N\times K_1\times\cdots\times K_M}$ and the associative law of this tensor product holds [4].

Definition 3. *Let $\mathbf{X} \in \mathbb{R}^{m\times n}$, the column-vector operator on a matrix $\mathbf{X}$ is an $mn\times 1$ vector which is denoted by $vec(\mathbf{X}) = (\mathbf{X}_{.1}^T, \mathbf{X}_{.2}^T, \ldots, \mathbf{X}_{.n}^T)^T$, where $\mathbf{X}_{.\mathbf{k}} = (x_{1k}, \ldots, x_{mk})^T$.*

Definition 4. *Let* $\mathbf{A} \in \mathbb{R}^{n\times n}$. *The trace of* $\mathbf{A}$ *is defined as the sum of its diagonal components:* $tr(\mathbf{A}) = \sum_{i=1}^{n} a_{ii}$.

A few utile properties for trace:

$$tr(\mathbf{A}) = tr(\mathbf{A}^T), \quad tr(\mathbf{ABC}) = tr(\mathbf{BCA}) = tr(\mathbf{CAB}), \quad tr(\mathbf{A}+\mathbf{B}) = tr(\mathbf{A}) + tr(\mathbf{B}).$$

Definition 5. *Let* $\mathbf{A} = (a_{mn})$ *be an* $m \times n$ *matrix and* $\mathbf{B} = (b_{pq})$ *be a* $p \times q$ *which is denoted* $\mathbf{A} \otimes \mathbf{B}$ *is defined as*

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \dots & a_{1n}\mathbf{B} \\ a_{21}\mathbf{B} & \dots & a_{2n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \dots & a_{mn}\mathbf{B} \end{bmatrix}.$$

Considering the matrix equation $\mathbf{AXB} = \mathbf{C}$, according to the definition of the Kronecker product, the following equivalent equation holds

$$vec(\mathbf{AXB}) = (\mathbf{B}^T \otimes \mathbf{A})vec(\mathbf{X}). \tag{2}$$

Theorem 2. *Let* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *be a matrix of rank* r. *Then there exist unitary matrices* $\mathbf{U} \in \mathbb{R}^{m\times m}$ *and* $\mathbf{V} \in \mathbb{R}^{n\times n}$ *such that*

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T, \tag{3}$$

where $\Sigma \in \mathbb{R}^{m\times n}$, $\Sigma = diag(\sigma_1, \sigma_2, \cdots, \sigma_r)$, *and* $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$. *The* σ_i *are called the singular values of the matrix* $\mathbf{A}$. *The decomposition* (3) *is the singular value decomposition (SVD) of the matrix* $\mathbf{A}$.

Definition 6. *[16] Let* $f : \mathbb{R}^n \to \mathbb{R}^m$ *be a mapping with* $\mathbf{y} = \mathbf{f}(\mathbf{x})$. *The partial derivative of the vector* $\mathbf{y}$ *with respect to the vector* $\mathbf{x}$ *is defined by*

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \dots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \dots & \frac{\partial y_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \frac{\partial y_m}{\partial x_2} & \dots & \frac{\partial y_m}{\partial x_n} \end{bmatrix},$$

where $\frac{\partial \mathbf{y}}{\partial \mathbf{x}} \in \mathbb{R}^{m\times n}$ *is the first-order partial derivates of the transformation from* $\mathbf{x}$ *to* $\mathbf{y}$. *The matrix* $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$ *is called the Jacobian matrix of the transformation* f.

Definition 7. *[16] Let* $\mathbf{A} \in \mathbb{R}^{m\times p}$ *be a function of vector* $\mathbf{x} \in \mathbb{R}^n$. *The partial derivative of matrix* $\mathbf{A}(\mathbf{x})$ *with respect to the vector* $\mathbf{x}$ *is defined by*

$$\begin{aligned} \frac{\partial \mathbf{A}}{\partial \mathbf{x}} &= \begin{bmatrix} \frac{\partial a_{11}}{\partial \mathbf{x}} & \frac{\partial a_{12}}{\partial \mathbf{x}} & \dots & \frac{\partial a_{1p}}{\partial \mathbf{x}} \\ \frac{\partial a_{21}}{\partial \mathbf{x}} & \frac{\partial a_{22}}{\partial \mathbf{x}} & \dots & \frac{\partial a_{2p}}{\partial \mathbf{x}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial a_{m1}}{\partial \mathbf{x}} & \frac{\partial a_{m2}}{\partial \mathbf{x}} & \dots & \frac{\partial a_{mp}}{\partial \mathbf{x}} \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial a_{11}}{\partial x_1} & \dots & \frac{\partial a_{11}}{\partial x_n} & \frac{\partial a_{12}}{\partial x_1} & \dots & \frac{\partial a_{12}}{\partial x_n} \dots & \frac{\partial a_{1p}}{\partial x_1} & \dots & \frac{\partial a_{1p}}{\partial x_n} \\ \frac{\partial a_{21}}{\partial x_1} & \dots & \frac{\partial a_{21}}{\partial x_n} & \frac{\partial a_{21}}{\partial x_1} & \dots & \frac{\partial a_{21}}{\partial x_n} \dots & \frac{\partial a_{1p}}{\partial x_1} & \dots & \frac{\partial a_{1p}}{\partial x_n} \\ \frac{\partial a_{m1}}{\partial x_1} & \dots & \frac{\partial a_{m1}}{\partial x_n} & \frac{\partial a_{m2}}{\partial x_1} & \dots & \frac{\partial a_{m2}}{\partial x_n} \dots & \frac{\partial a_{mp}}{\partial x_1} & \dots & \frac{\partial a_{mp}}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{m\times pn}. \end{aligned}$$

Note that the partial derivative $\frac{\partial \mathbf{A}}{\partial \mathbf{x}}$ can be reduced to a third order tensor in $\mathbb{R}^{m\times n\times p}$.

Definition 8. *[12] If $\mathbf{Y}=(y_{ij})\in\mathbb{R}^{p\times q}$ and $\mathbf{X}=(x_{ij})\in\mathbb{R}^{m\times n}$, the derivative of the matrix $\mathbf{Y}$ with respect to the matrix $\mathbf{X}$ is defined as follows*

$$\frac{\partial \mathbf{Y}}{\partial \mathbf{X}}=\begin{bmatrix} \frac{\partial \mathbf{Y}}{\partial x_{11}} & \frac{\partial \mathbf{Y}}{\partial x_{12}} & \cdots & \frac{\partial \mathbf{Y}}{\partial x_{1n}} \\ \frac{\partial \mathbf{Y}}{\partial x_{21}} & \frac{\partial \mathbf{Y}}{\partial x_{22}} & \cdots & \frac{\partial \mathbf{Y}}{\partial x_{2n}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \mathbf{Y}}{\partial x_{m1}} & \frac{\partial \mathbf{Y}}{\partial x_{32}} & \cdots & \frac{\partial \mathbf{Y}}{\partial x_{mn}} \end{bmatrix}\mathbb{R}^{mp\times nq},$$

where

$$\frac{\partial \mathbf{Y}}{\partial x_{ij}}=\begin{bmatrix} \frac{\partial y_{11}}{\partial x_{ij}} & \frac{\partial y_{12}}{\partial x_{ij}} & \cdots & \frac{\partial y_{1q}}{\partial x_{ij}} \\ \frac{\partial y_{21}}{\partial x_{ij}} & \frac{\partial y_{22}}{\partial x_{ij}} & \cdots & \frac{\partial y_{2q}}{\partial x_{ij}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial y_{p1}}{\partial x_{ij}} & \frac{\partial y_{p2}}{\partial x_{ij}} & \cdots & \frac{\partial y_{pq}}{\partial x_{ij}} \end{bmatrix}.$$

Lemma 1. *[16] If $\mathbf{A}\in\mathbb{R}^{m\times n}$, $\mathbf{b}\in\mathbb{R}^{n}$ and $\mathbf{x}\in\mathbb{R}^{p}$, then the partial derivative of the product of matrix $\mathbf{A}$ and vector $\mathbf{b}$ with respect to the vector $\mathbf{x}$ is defined by the following rule*

$$\frac{\partial \mathbf{Ab}}{\partial \mathbf{x}}=(\mathbf{b}^T\otimes \mathbf{I}_m)\frac{\partial \mathbf{A}}{\partial \mathbf{x}}+\mathbf{A}\frac{\partial \mathbf{b}}{\partial \mathbf{x}}.$$

Considering Eq. (2) and the chain rule, the following Lemma can be easily proved.

Lemma 2. *Let $\mathbf{X}\in\mathbb{R}^{m\times n}$, $\mathbf{Y}\in\mathbb{R}^{n\times v}$ and $\mathbf{Z}\in\mathbb{R}^{p\times q}$. Then the following rule for the derivative of a matrix product with respect to another matrix applies*

$$\frac{\partial \mathbf{XY}}{\partial \mathbf{Z}}=(\mathbf{Y}^T\otimes \mathbf{I}_m)\frac{\partial \mathbf{X}}{\partial \mathbf{Z}}+(\mathbf{I}_v\otimes \mathbf{X})\frac{\partial \mathbf{Y}}{\partial \mathbf{Z}}.$$

Lemma 3. *Let $\mathbf{A}=\mathbf{A}(\mathbf{x})\in\mathbb{R}^{m\times n}$ be a differentiable function of each of the elements of $\mathbf{x}$, then*

$$\frac{\partial\, tr(\mathbf{A}(\mathbf{x}))}{\partial\, \mathbf{x}}=tr(\frac{\partial\, \mathbf{A}(\mathbf{x})}{\partial\, \mathbf{x}}).$$

Proof. The trace of $\mathbf{A}(\mathbf{x})$ is given by $tr(\mathbf{A}(\mathbf{x}))=\sum_{i=1}^{n}a_{ii}(x)$. Then

$$\frac{\partial\, tr(\mathbf{A}(\mathbf{x}))}{\partial\, \mathbf{x}}=\frac{\partial}{\partial\, \mathbf{x}}\sum_{i=1}^{n}a_{ii}(x)=\sum_{i=1}^{n}\frac{\partial a_{ii}(x)}{\partial \mathbf{x}}=tr(\frac{\partial \mathbf{A}(\mathbf{x})}{\partial \mathbf{x}}).$$

This shows that the differential of $tr(\mathbf{A}(\mathbf{x}))$ with respect to $\mathbf{x}$ is simply the sum of the differentials of the diagonal elements of $\mathbf{A}(\mathbf{x})$. □

Considering Lemma 3, the properties of the matrix trace and the chain rule, the following proposition can be easily shown.

Proposition 1. *Suppose* $\mathbf{A}, \mathbf{B}$ *and* $\mathbf{C}$ *are matrices in* $\mathbb{R}^{n\times n}$*, the following relations hold:*

$$\begin{cases} a.\ \frac{\partial}{\partial \mathbf{X}} tr(\mathbf{AXBX}) = \mathbf{A}^T\mathbf{X}^T\mathbf{B}^T + \mathbf{B}^T\mathbf{X}^T\mathbf{A}^T, \\ b.\ \frac{\partial}{\partial \mathbf{X}} tr(\mathbf{BX^TX}) = \mathbf{XB}^T + \mathbf{XB}, \\ c.\ \frac{\partial}{\partial \mathbf{X}} tr(\mathbf{AX^TB}) = \mathbf{BA}, \\ d.\ \frac{\partial}{\partial \mathbf{X}} tr(\mathbf{X}) = \mathbf{I_n}, \\ e.\ \frac{\partial}{\partial \mathbf{X}} tr(\mathbf{XA}) = \mathbf{A}^T, \\ f.\ \frac{\partial}{\partial \mathbf{X}} tr(\mathbf{AX^T}) = \mathbf{A}. \end{cases}$$

3 The Gauss-Newton method for solving TLS problems

In this section, we recall some properties of the Gauss-Newton method, which is an iterative method for solving TLS problems.

We consider the total least equares problem (1). Given a matrix $[\bar{\mathbf{E}}, \bar{\mathbf{f}}]$ that attains the minimum in (1), a solution of TLS is any $\mathbf{x} \in \mathbb{R}^n$ such that $(\mathbf{A} + \bar{\mathbf{E}})\mathbf{x} = (\mathbf{b} + \bar{\mathbf{f}})$. Consider the following minimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \eta(x) = \frac{\|\mathbf{Ax} - \mathbf{b}\|_2}{\sqrt{1+\mathbf{x}^T\mathbf{x}}}. \tag{4}$$

The $\mathbf{x}_{TLS}$ solution for the TLS problem is equivalent to solving the nonlinear least squares minimization problem (4). The following theorem confirms this.

Theorem 3. *[7] For each vector* $\mathbf{x}$ *there exists a rank one matrix* $[\bar{\mathbf{E}}, \bar{\mathbf{f}}]$ *such that* $(\mathbf{A} + \bar{\mathbf{E}})\mathbf{x} = (\mathbf{b} + \bar{\mathbf{f}})$ *and* $\|[\bar{\mathbf{E}}, \bar{\mathbf{f}}]\|_F = \|[\bar{\mathbf{E}}, \bar{\mathbf{f}}]\|_2 = \eta(x)$*. Moreover, for each matrix* $[\mathbf{E}, \mathbf{f}]$ *such that* $(\mathbf{A}+\mathbf{E})\mathbf{x} = (\mathbf{b}+\mathbf{f})$ *it holds*

$$\|[\mathbf{E}, \mathbf{f}]\|_F \geq \|[\mathbf{E}, \mathbf{f}]\|_2 \geq \eta(x).$$

Eq. (4) is a standard nonlinear optimization problem, so optimization techniques can be used to solve nonlinear problems. The Gauss-Newton method is a modification of the Newton method for solving nonlinear problems. In nonlinear problems, the Gauss-Newton method is used to minimize the sum of the values of square function without the need to calculate the second order derivatives.

Let $\mathbf{f} : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ be a nonlinear, continuously differentiable function. Consider the unconstrained optimization problem

$$\min_{\mathbf{x}} \Phi(\mathbf{x}) = \frac{1}{2}\|\mathbf{f}(\mathbf{x})\|_2^2 = \frac{1}{2}\sum_{i=1}^{m} \mathbf{f}_i^2(\mathbf{x}). \tag{5}$$

where $\mathbf{x}$ is an n-dimensional real vector and $\mathbf{f}$ is an m-dimensional real vector function of $\mathbf{x}$.

Finding the minimum value $\Phi(\mathbf{x}) = \|\mathbf{f}(\mathbf{x})\|_2^2$ is equivalent to solving the equation $\nabla\Phi(\mathbf{x}) = 0$. The Gauss-Newton algorithm is a popular method for solving (5). Differentiating (5) with respect to $\mathbf{x}_j$ gives

$$\frac{\partial \Phi(\mathbf{x})}{\partial \mathbf{x}_j} = \sum_{i=1}^{m} \frac{\partial \mathbf{f_i}(\mathbf{x})}{\partial \mathbf{x}_j}\mathbf{f_i}.$$

So the gradient of Φ is

$$\nabla\Phi(\mathbf{x}) = \mathbf{J}(\mathbf{x})^T\mathbf{f}(\mathbf{x}),$$

where $\mathbf{J} \in \mathbb{R}^{m\times n}$ represents the Jacobian matrix containing the first partial derivatives of the components of function $\mathbf{f}$. The Hessian matrix is given by

$$\nabla^2\Phi(\mathbf{x}) = \mathbf{J}(\mathbf{x})^T\mathbf{J}(\mathbf{x}) + \mathbf{Q}(\mathbf{x}),$$

where $\mathbf{Q}(\mathbf{x}) = \sum_{i=1}^{m}\mathbf{f_i}(\mathbf{x})\nabla^2(\mathbf{f_i}(\mathbf{x}))$. Now, write the objective function $\mathbf{f}(\mathbf{x})$ as the second order Taylor series expansion, which is the quadratic model, as follows:

$$\mathbf{q}(\mathbf{x}) = \frac{1}{2}\mathbf{f}(\mathbf{x_k})^T\mathbf{f}(\mathbf{x_k}) + \mathbf{J}(\mathbf{x_k})^T\mathbf{f}(\mathbf{x_k})(\mathbf{x}-\mathbf{x_k}) + \frac{1}{2}(\mathbf{x}-\mathbf{x_k})^T(\mathbf{Q}(\mathbf{x}) + \mathbf{J}(\mathbf{x_k})^T\mathbf{J}(\mathbf{x_k}))(\mathbf{x}-\mathbf{x_k}).$$

By taking the derivative of $\mathbf{q}$ and setting it equal to zero, we have

$$\mathbf{J}(\mathbf{x_k})^T\mathbf{f}(\mathbf{x_k}) + (\mathbf{Q}(\mathbf{x}) + \mathbf{J}(\mathbf{x_k})^T\mathbf{J}(\mathbf{x_k}))(\mathbf{x}-\mathbf{x_k}) = 0.$$

One motivation for this approach is that $\mathbf{Q}(\mathbf{x})$ is negligible. Now, neglecting the term $\mathbf{Q}(\mathbf{x})$, in the Gauss-Newton method, the next iteration is obtained with $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k$, where $\mathbf{h}_k$ is obtained by solving the following equation.

$$(\mathbf{J}(\mathbf{x}_k)^T\mathbf{J}(\mathbf{x}_k))\mathbf{h}_k = -\mathbf{J}(\mathbf{x}_k)^T\mathbf{f}(\mathbf{x}_k) \tag{6}$$

Eq. (6) is equivalent to the following linear least squares problem

$$\min_{\mathbf{h}}\|\mathbf{J}(\mathbf{x}_k)\mathbf{h} + \mathbf{f}(\mathbf{x}_k)\|_2. \tag{7}$$

The linear least squares problem (7) can be efficiently solved using orthogonal transformations or iterative methods.

The formulation (4) of TLS can be recast as a nonlinear least squares problem in the form (5). We set $\eta(\mathbf{x}) = \|\mathbf{f}(\mathbf{x})\|_2^2$, where $\mathbf{f}(\mathbf{x}) = \frac{\mathbf{Ax}-\mathbf{b}}{\sqrt{1+\mathbf{x}^T\mathbf{x}}}$. The Jacobian matrix of $\mathbf{f}$ is

$$\mathbf{J}(\mathbf{x}) = \frac{\mathbf{A}}{\sqrt{1+\mathbf{x}^T\mathbf{x}}} - \frac{(\mathbf{Ax}-\mathbf{b})\mathbf{x}^T}{\sqrt{1+\mathbf{x}^T\mathbf{x}}}.$$

The main steps of the Gauss-Newton algorithm for the TLS (GN-TLS) can be summarized as follows:

Algorithm 1 GN-TLS method

```
1: Input: $\mathbf{f}(\mathbf{x})$, $\mathbf{J}(\mathbf{x})$, $\mathbf{x}_0$, $\varepsilon$, $maxit$ (termination parameters)
2: Output: $\mathbf{x}^*$, approximate solution of (5)
3: $k = 0$, $\mathbf{f}_0 = \mathbf{h}(\mathbf{x})$, $\mathbf{J}_0 = \mathbf{J}(\mathbf{x}_0)$
4: while $\|\mathbf{J}(\mathbf{x}_k)^T\mathbf{f}_k\|_2 \geq \varepsilon$ and $k < maxit$ do
5:     Compute $\mathbf{h}_k = arg\min_{\mathbf{h}}\|\mathbf{f}(\mathbf{x}_k) + \mathbf{J}(\mathbf{x}_k)\mathbf{h}\|_2$
6:     $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k$,,
7:     $k = k+1$, $\mathbf{f}_k = \mathbf{f}(\mathbf{x}_k)$, $\mathbf{J}_k = \mathbf{J}(\mathbf{x}_k)$
8: end while
9: $\mathbf{x}^* = \mathbf{x}_k$
```

4 The proposed method

The generalized total least squares problem is an extension of the TLS problem, where the single right-hand side is reduced to multiple right-hand sides. Signal processing, system theory, autonomous control, engineering, physics, and data fitting are among the fields that employ the general total least squares [6–8, 14, 18, 22, 30, 31].

The generalized total least squares problem arises when solving overdetermined linear systems with multiple right hand sides $\mathbf{A}\mathbf{X} \approx \mathbf{B}$ where errors exist in the data matrix $\mathbf{A}$ and the observation matrix $\mathbf{B}$. In this section, we propose an iterative method inspired by the Gauss-Newton method for the generalized least-squares problem. To do so, we first define the generalized least-squares problem as follows:

Definition 9. *Given $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{B} \in \mathbb{R}^{m\times s}(m \geq (n+s))$, the generalized total least squares problem is defined as*

$$\begin{cases} \min\limits_{\mathbf{E},\mathbf{F}} \|[\mathbf{E},\mathbf{F}]\|_F, \\ s.t.\;\; (\mathbf{A}+\mathbf{E})\mathbf{X} = \mathbf{B}+\mathbf{F}, \end{cases} \tag{8}$$

where $\mathbf{E} \in \mathbb{R}^{m\times n}$, $\mathbf{X} \in \mathbb{R}^{n\times s}$ and $\mathbf{F} \in \mathbb{R}^{m\times s}$.

SVD is a powerful matrix decomposition technique that reveals the underlying structure of a matrix. Given that SVD decomposition is a useful tool for describing a GTLS solution. If the SVD decomposition of $[\mathbf{A},\mathbf{B}]$ is given by

$$[\mathbf{A},\mathbf{B}] = \mathbf{U}\Sigma\mathbf{V}^T, \quad \Sigma = (\sigma_1, \sigma_2, \cdots, \sigma_{n+s}) \in \mathbb{R}^{(n+s)\times(n+s)},$$

where $\sigma_i = \sigma_i([\mathbf{A},\mathbf{B}]), \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{n+s} \geq 0$, $\mathbf{U} \in \mathbb{R}^{m\times(n+s)}$ and $\mathbf{V} \in \mathbb{R}^{(n+s)\times(n+s)}$, then partition of the matrix $\mathbf{V}$ is as follows:

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} \\ \mathbf{V}_{21} & \mathbf{V}_{22} \end{bmatrix},$$

where $\mathbf{V}_{11} \in \mathbb{R}^{n\times n}$, $\mathbf{V}_{12} \in \mathbb{R}^{n\times s}$, $\mathbf{V}_{21} \in \mathbb{R}^{s\times n}$ and $\mathbf{V}_{22} \in \mathbb{R}^{s\times s}$. We have the following theorem.

Theorem 4. *[29] Let $[\mathbf{A},\mathbf{B}] = U\Sigma V^T$ be the SVD decomposition of $[\mathbf{A},\mathbf{B}]$. If $\sigma_n \geq \sigma_{n+1}$ and $\mathbf{V}_{22}$ is nonsingular, then $\mathbf{X}_{GTLS} = -\mathbf{V}_{12}\mathbf{V}_{22}^{-1}$, exists and solves the GTLS problem.*

The following theorem states sufficient conditions for the existence and uniqueness of the solution to the GTLS problem.

Theorem 5. *[29] Let $[\mathbf{A},\mathbf{B}] = \mathbf{U}\Sigma\mathbf{V}^T$ be the SVD of $[\mathbf{A},\mathbf{B}]$ and $\mathbf{A} = \mathbf{U}'\Sigma'\mathbf{V}^{T}{'}$ be the SVD of $\mathbf{A}$. If $\sigma{\prime}_n > \sigma_{n+1}$ where $\sigma{\prime}_n$ the singular value of $\mathbf{A}$, then $\sigma_n \geq \sigma_{n+1}$ and $\mathbf{V}_{22}$ is nonsingular.*

For more details of conditions for the existence and uniqueness of the solution to the GTLS problem and finding the solution via SVD decomposition, see [29] and references therein. One of the common approaches for obtaining the solution of the GTLS problem is to solve a nonlinear equation, which is discussed in this section. To begin with, we state the following theorem.

Theorem 6. *For any matrix* $\mathbf{X}$ *there exists a matrix* $[\bar{\mathbf{E}}, \bar{\mathbf{F}}]$ *such that* $(\mathbf{A}+\bar{\mathbf{E}})\mathbf{X} = \mathbf{B}+\bar{\mathbf{F}}$ *and* $\|[\bar{\mathbf{E}}, \bar{\mathbf{F}}]\|_F = \|\mathbf{G}(\mathbf{X})\|_F$*, where* $\mathbf{G}(\mathbf{X}) = (\mathbf{A}\mathbf{X}-\mathbf{B})(\mathbf{I}_s+\mathbf{X}^T\mathbf{X})^{-\frac{1}{2}}$*. Moreover, for every matrix* $[\mathbf{E},\mathbf{F}]$ *such that* $(\mathbf{A}+\mathbf{E})\mathbf{X} = \mathbf{B}+\mathbf{F}$*, the following inequalities hold*

(a) $\|[\mathbf{E},\mathbf{F}]\|_F \geq \dfrac{\|\mathbf{G}(\mathbf{X})\|_F}{\sqrt{s}}$,

(b) $\|[\mathbf{E},\mathbf{F}]\|_2 \geq \|\mathbf{G}(\mathbf{X})\|_2$.

Proof. Let $\mathbf{R} = \mathbf{A}\mathbf{X}-\mathbf{B}$ and $\mathbf{Y} = (\mathbf{X}^T, -\mathbf{I}_s)^T$, we define $\bar{\mathbf{E}} = -\mathbf{R}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{X}^T$, $\bar{\mathbf{F}} = \mathbf{R}(\mathbf{Y}^T\mathbf{Y})^{-1}$. So

$$\begin{aligned}(\mathbf{A}+\bar{\mathbf{E}})\mathbf{X} = \mathbf{A}\mathbf{X}+\bar{\mathbf{E}}\mathbf{X} &= \mathbf{R}+\mathbf{B}-\mathbf{R}(\mathbf{Y}^{\mathbf{T}}\mathbf{Y})^{-1}\mathbf{X}^T\mathbf{X}\\ &= \mathbf{B}+\mathbf{R}(\mathbf{Y}^T\mathbf{Y})^{-1}(\mathbf{Y}^T\mathbf{Y}-\mathbf{X}^T\mathbf{X})\\ &= \mathbf{B}+\mathbf{R}(\mathbf{Y}^T\mathbf{Y})^{-1} = \mathbf{B}+\bar{\mathbf{F}}.\end{aligned}$$

Note that

$$[\bar{\mathbf{E}}, \bar{\mathbf{F}}] = (\mathbf{R}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{X}^T, \mathbf{R}(\mathbf{Y}^T\mathbf{Y})^{-1}) = -\mathbf{R}(\mathbf{Y}^T\mathbf{Y})^{-1}(\mathbf{X}^T, -\mathbf{I}_s) = -\mathbf{R}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T,$$

accordingly

$$\begin{aligned}\|[\bar{\mathbf{E}}, \bar{\mathbf{F}}]\|_F^2 &= tr(\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{R}^T\mathbf{R}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T) = tr((\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{R}^T\mathbf{R}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{Y})\\ &= tr(\mathbf{R}^T\mathbf{R}(\mathbf{Y}^T\mathbf{Y})^{-1}) = \|\mathbf{R}(\mathbf{Y}^T\mathbf{Y})^{\frac{-1}{2}}\|_F^2 = \|\mathbf{G}(\mathbf{X})\|_F^2.\end{aligned}$$

This proves the first part of the theorem.

To prove the second part, if $(\mathbf{A}+\mathbf{E})\mathbf{X} = \mathbf{B}+\mathbf{F}$, we onsider $(\mathbf{A}+\mathbf{E})\mathbf{X} = \mathbf{B}+\mathbf{F}$ and $\mathbf{Y} = [\mathbf{X}^T, -\mathbf{I}_s]^T$. Since

$$\mathbf{E}\mathbf{X}-\mathbf{F} = [\mathbf{E},\mathbf{F}]\mathbf{Y} = \mathbf{B}-\mathbf{A}\mathbf{X} = -\mathbf{R},$$

it results in

$$\mathbf{G}(\mathbf{X}) = \mathbf{R}(\mathbf{Y}^T\mathbf{Y})^{\frac{-1}{2}} = -[\mathbf{E},\mathbf{F}]\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{\frac{-1}{2}}.$$

Therefore, by using the properties of the matrix norm, we have

$$\begin{aligned}\|\mathbf{G}(\mathbf{X})\|_F^2 = \|-[\mathbf{E},\mathbf{F}]\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{\frac{-1}{2}}\|_F^2 &\leq \|[\mathbf{E},\mathbf{F}]\|_F^2\|\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{\frac{-1}{2}}\|_F^2\\ &= \|[\mathbf{E},\mathbf{F}]\|_F^2 tr((\mathbf{Y}^T\mathbf{Y})^{\frac{-1}{2}}\mathbf{Y}^T\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{\frac{-1}{2}})\\ &= \|[\mathbf{E},\mathbf{F}]\|_F^2 tr((\mathbf{Y}^T\mathbf{Y})^{\frac{-1}{2}}(\mathbf{Y}^T\mathbf{Y})^{\frac{-1}{2}}\mathbf{Y}^T\mathbf{Y})\\ &= \|[\mathbf{E},\mathbf{F}]\|_F^2 tr(\mathbf{I}_s) = s\|[\mathbf{E},\mathbf{F}]\|_F^2,\end{aligned}$$

and this completes the proof of the case (a).

Since $\|\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{\frac{-1}{2}}\|_2 = 1$, the part (b) is proved similarly. □

According to Theorem 6, therefore, to solve the GTLS problem (8), it is necessary to solve the following nonlinear optimization problem:

$$\min_{\mathbf{X}\in\mathbb{R}^{n\times s}} \mathbf{F}(X) = \|\mathbf{G}(\mathbf{X})\|_F^2. \tag{9}$$

The Taylor series, a powerful mathematical tool, enables the approximation of nonlinear functions by using the derivative of various orders of the function at a particular point. For this purpose, since $\mathbf{G} : \mathbb{R}^{n\times s} \to \mathbb{R}^{m\times s}$ is a nonlinear function, we use the Taylor series expansion to approximate $\mathbf{G}(\mathbf{X})$. Regarding the matrix function $\mathbf{G}$, the Taylor expansion of $\mathbf{G}$ around $\mathbf{X}_k$ involves the Jacobian matrix $\mathbf{J}(\mathbf{X}_k) \in \mathbb{R}^{ms\times ns}$. For finding the step lenght of the Gauss-Newton iteration method, there are two ways. One is reshaping the Jacobian matrix $\mathbf{J}(\mathbf{X}_k) \in \mathbb{R}^{ms\times ns}$ into the fourth order tensor $\mathscr{J}_k \in \mathbb{R}^{m\times s\times n\times s}$ and we have the following tensor approximation

$$\mathscr{G}(\mathbf{X}) \approx \mathscr{G}_k + \mathscr{J}_k *_2 \mathscr{D}_k, \tag{10}$$

where $\mathscr{G}_k$ and $\mathscr{D}_k$ are the tensor form of the matrices $\mathbf{G}(\mathbf{X_k})$ and $\mathbf{D}_k = \mathbf{X} - \mathbf{X_k}$ and Eq. (10) results in the GN-GTLS algorithm. Another way is vectorizing the matrix $\mathbf{D}_k = (\mathbf{X} - \mathbf{X}_k) \in \mathbb{R}^{n\times s}$ into the vector $d_k \in \mathbb{R}^{ns}$ and we have the following approximation

$$\mathbf{g}(\mathbf{x}) \approx \mathbf{g}_k + \mathbf{J}_k d_k, \tag{11}$$

where $\mathbf{g}(\mathbf{x})$ and $\mathbf{g}_k$ are the vectorization of the matrices $\mathbf{G}(\mathbf{X})$ and $\mathbf{G}(\mathbf{X}_k)$, respectively, and Eq. (11) results in the GN-TLS algorithm [7]. Note that applying Eq. (10) to find the step length of the Gauss-Newton iteration method is much less expensive than other one. The numerical results show this claim. Based on Eqs. (10) and (11), it is necessary to first determine the derivative of the nonlinear function G. To do this, we present the following proposition.

Proposition 2. *Given* $\mathbf{G}(\mathbf{X}) = (\mathbf{A}\mathbf{X} - \mathbf{B})(\mathbf{I}_s + \mathbf{X}^T\mathbf{X})^{-\frac{1}{2}}$, *then the Jacobian tensor of* $\mathbf{G}(\mathbf{X})$ *is as follows:*

$$\frac{\partial \mathbf{G}(\mathbf{X})}{\partial \mathbf{X}} = (\mathbf{H}^{-1} \otimes \mathbf{A}) - (\mathbf{H}^{-1} \otimes \mathbf{G}(\mathbf{X}))[(\mathbf{I}_s \otimes \mathbf{H}) + (\mathbf{H} \otimes \mathbf{I}_s)]^{-1} \frac{\partial (\mathbf{I}_s + \mathbf{X}^T\mathbf{X})}{\partial \mathbf{X}},$$

where $\mathbf{H} = (\mathbf{I}_s + \mathbf{X}^T\mathbf{X})^{\frac{1}{2}}$.

Proof. By taking $\mathbf{R} = \mathbf{A}\mathbf{X} - \mathbf{B}$, we have

$$\mathbf{G}(\mathbf{X}) = (\mathbf{A}\mathbf{X} - \mathbf{B})(\mathbf{I}_s + \mathbf{X}^T\mathbf{X})^{\frac{-1}{2}} = \mathbf{R}\mathbf{H}^{-1}.$$

Based on Lemma 2, we deduce

$$\frac{\partial \mathbf{G}(\mathbf{X})}{\partial \mathbf{X}} = (\mathbf{I}_s \otimes \mathbf{R}) \frac{\partial \mathbf{H}^{-1}}{\partial \mathbf{X}} + (\mathbf{H}^{-1} \otimes \mathbf{I}_m) \frac{\partial \mathbf{R}}{\partial \mathbf{X}}. \tag{12}$$

Also, we can derive the expressions for the derivatives of $\frac{\partial \mathbf{R}}{\partial \mathbf{X}}$ and $\frac{\partial \mathbf{H}^{-1}}{\partial \mathbf{X}}$ as follows

$$\frac{\partial \mathbf{R}}{\partial \mathbf{X}} = \frac{\partial (\mathbf{A}\mathbf{X} - \mathbf{B})}{\partial \mathbf{X}} = (\mathbf{I}_s \otimes \mathbf{A}) \frac{\partial \mathbf{X}}{\partial \mathbf{X}} + (\mathbf{X}^T \otimes \mathbf{I}_m) \frac{\partial \mathbf{A}}{\partial \mathbf{X}}.$$

Since $\frac{\partial \mathbf{A}}{\partial \mathbf{X}} = 0$ and $\frac{\partial \mathbf{X}}{\partial \mathbf{X}} = \mathbf{I}_{ns}$, it results in

$$\frac{\partial \mathbf{R}}{\partial \mathbf{X}} = (\mathbf{I}_s \otimes \mathbf{A}). \tag{13}$$

By taking the derivative both sides of the equation $\mathbf{H}^{-1}\mathbf{H} = \mathbf{I}_s$ with respect to $\mathbf{X}$, we have

$$(\mathbf{I}_s \otimes \mathbf{H}^{-1})\frac{\partial \mathbf{H}}{\partial \mathbf{X}} + (\mathbf{H} \otimes \mathbf{I}_s)\frac{\partial \mathbf{H}^{-1}}{\partial \mathbf{X}} = \frac{\partial \mathbf{I}_s}{\partial \mathbf{X}},$$

since $\frac{\partial \mathbf{I}_s}{\partial \mathbf{X}} = 0$, it obtains

$$\frac{\partial \mathbf{H}^{-1}}{\partial \mathbf{X}} = -(\mathbf{H}^{-1} \otimes \mathbf{H}^{-1})\frac{\partial \mathbf{H}}{\partial \mathbf{X}}. \tag{14}$$

Since $\frac{\partial \mathbf{H}}{\partial \mathbf{X}} = [(\mathbf{I}_s \otimes \mathbf{H}) + (\mathbf{H} \otimes \mathbf{I}_s)]^{-1}\frac{\partial((\mathbf{I}_s + \mathbf{X}^T\mathbf{X})}{\partial \mathbf{X}}$, by leveraging the properties of Kronecker multiplication and referencing Eqs. (13) and (14), one can calculate the derivative of the matrix $\mathbf{G}(\mathbf{X})$ as follows

$$\begin{aligned}\frac{\partial \mathbf{G}(\mathbf{X})}{\partial \mathbf{X}} &= (\mathbf{H}^{-1} \otimes \mathbf{I}_m)(\mathbf{I}_s \otimes \mathbf{A}) - (\mathbf{H}^{-1} \otimes \mathbf{G}(\mathbf{X}))[(\mathbf{I}_s \otimes \mathbf{H}) + (\mathbf{H} \otimes \mathbf{I}_s)]^{-1}\frac{\partial(\mathbf{I}_s + \mathbf{X}^T\mathbf{X})}{\partial \mathbf{X}} \\ &= (\mathbf{H}^{-1} \otimes \mathbf{A}) - (\mathbf{H}^{-1} \otimes \mathbf{G}(\mathbf{X}))[(\mathbf{I}_s \otimes \mathbf{H}) + (\mathbf{H} \otimes \mathbf{I}_s)]^{-1}\frac{\partial(\mathbf{I}_s + \mathbf{X}^T\mathbf{X})}{\partial \mathbf{X}}.\end{aligned}$$

□

According to the approximation (10), the optimization problem (9) approximates to the following optimization problem

$$\min_{\mathbf{X}\in\mathbb{R}^{n\times s}} \mathbf{F}(\mathbf{X}) \approx \|\mathscr{G}_k + \mathscr{J}_k *_2 \mathscr{D}_k\|_F^2,$$

where $\mathscr{J}_k$ represents the tensor form of gradiant of $\mathbf{G}(\mathbf{X_k})$. Note that

$$\begin{aligned}\|\mathscr{G}_k + \mathscr{J}_k *_2 \mathscr{D}_k\|_F^2 &= tr((\mathscr{G}_k + \mathscr{J}_k *_2 \mathscr{D}_k)^T *_2 (\mathscr{G}_k + \mathscr{J}_k *_2 \mathscr{D}_k)) \\ &= tr((\mathscr{G}_k^T + \mathscr{D}_k^T *_2 \mathscr{J}_k^T) *_2 (\mathscr{G}_k + \mathscr{J}_k *_2 \mathscr{D}_k)) \\ &= tr(\mathscr{G}_k^T *_2 \mathscr{G}_k + \mathscr{G}_k^T *_2 \mathscr{J}_k *_2 \mathscr{D}_k \\ &\quad + \mathscr{D}_k^T *_2 \mathscr{J}_k^T *_2 \mathscr{G}_k + \mathscr{D}_k^T *_2 \mathscr{J}_k^T *_2 \mathscr{J}_k *_2 \mathscr{D}_k) \\ &= tr(\mathscr{G}_k^T *_2 \mathscr{G}_k) + tr(\mathscr{G}_k^T *_2 \mathscr{J}_k *_2 \mathscr{D}_k) \\ &\quad + tr(\mathscr{D}_k^T *_2 \mathscr{J}_k^T *_2 \mathscr{G}_k) + tr(\mathscr{D}_k^T *_2 \mathscr{J}_k^T *_2 \mathscr{J}_k *_2 \mathscr{D}_k)).\end{aligned}$$

Tensors are generalizations of matrices. A tensor differentiation, similar to a matrix differentiation, can be expressed using Einstein's product. In other words, the similar principles of the matrix derivative can be applied to the tensors to facilitate the calculations involving tensors.

Since $\frac{\partial \mathscr{D}_k}{\partial \mathbf{X}} = \mathbf{I}_{ns}$ and according to Proposition 1, we have

$$\frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{X}} \approx 2\mathscr{J}_k^T *_2 \mathscr{G}_k + 2\mathscr{J}_k^T *_2 \mathscr{J}_k *_2 \mathscr{D}_k.$$

Finding the minimum value of $\mathbf{F}(\mathbf{X})$ is equivalent to solving the equation $\nabla\mathbf{F}(X) = 0$. Therefore, we have the following tensor equation

$$(\mathscr{J}_k^T *_2 \mathscr{J}_k) *_2 \mathscr{D}_k = -\mathscr{J}_k^T *_2 \mathscr{G}_k. \tag{15}$$

The tensor equation (15) is similar to the Gauss-Newton method. For this propose, we have named the proposed method as the Gauss-Newton-type method. In the Gauss-Newton-type method, the next iteration is given by $\mathbf{X}_{k+1} = \mathbf{D}_k + \mathbf{X}_k$, where the direction matrix $\mathbf{D}_k$ is the matrix form of the solution of the tensor equation (15). To solve the tensor equation (15), we employed the GLS-BTF algorithm [15].

The main steps of the Gauss-Newton-type algorithm are summarized in Algorithm 2.

Algorithm 2 GN-GTLS algorithm for the GTLS problem

```
    Input: $\mathbf{X}_0, \varepsilon, maxit$ (termination parameters)
2:  Output: $X^*, \bar{E}(X^*), \bar{F}(X^*)$
    $k = 0,\ \mathscr{G}_0 = \mathscr{G}(X_0),\ \mathscr{J}_0 = \mathscr{J}(X_0)$
4:  while $\|\mathscr{J}_k^T *_2 \mathscr{G}_k\|_F \geq \varepsilon$ and $k < maxit$ do
        Compute $\mathscr{D}_k$ from Eq. (15)
6:      $\mathbf{X_{k+1}} = \mathbf{X_k} + \mathbf{D_k}$
        $k = k+1,\ \mathscr{G}_k = \mathscr{G}(X_k),\ \mathscr{J}_k = \mathscr{J}(X_k)$
8:  end while
    $X^* = X_k,\ \bar{E}(X^*),\ \bar{F}(X^*)$.
```

5 Numerical Examples

In this section, we give some numerical examples to show the performance and efficiency of the proposed method for solving the GTLS problem (8). Also, using Eq. (2), we reduce the GTLS problem to the associated TLS problem as follows:

$$(\mathbf{I} \otimes (\mathbf{A} + \mathbf{E}))vec(\mathbf{X}) = vec(\mathbf{B} + \mathbf{F}).$$

Then, we compare GN-GTLS with GN-TLS and show that GN-GTLS has the better results than the GN-TLS method.

All tests were run on the Intel (R), Core (TM) i7-8565U processor, with a 2.00GHz CPU and 16GB RAM. The programming language used was MATLAB R2021b. All used codes came from the MATLAB tensor toolbox developed by [17]. We know that it will enhance the practicality of our findings if we utilize the original image sizes in the numerical experiments. However, regarding the limitation of our personal computer, we reduced the dimension of images in the experimental examples.

In all the following examples, we choose the initial matrix to be the zero matrix and the stopping criterion is

$$\|\mathscr{J}_k^T *_2 \mathscr{G}_k\|_F \leq 10^{-7}.$$

Example 1. Consider the GTLS problem (8) to solve the deblurring image problem. In this example, $\mathbf{X}_{true} \in \mathbb{R}^{100\times 100}$ is gray-scale image (uncontaminated image) and $\mathbf{A}_{true}$ is blurring matrix. The matrix $\mathbf{A}_{true}$ is generated [24] by

$$\begin{cases} z = [exp(-([0:band-1]^2)/(2\sigma^2)), zeros(1, n-band)], \\ \mathbf{A}_{true} = \frac{1}{\sigma\sqrt{2\pi}} toeplitz(z(1)fliplr(z(2:end)), z), \end{cases}$$

where the MATLAB command $fliplr$ returns a vector with the order of elements flipped left and right along the second dimension and $n = 100, \sigma = 4, band = 7$. The data matrix $\mathbf{B}_{blur} \in \mathbb{R}^{100\times 100}$ is given by $\mathbf{A}_{true}\mathbf{X}_{true} = \mathbf{B}_{blur}$. However, the observation and data matrices are often prone to errors. To account for this, we construct error matrices for both $\mathbf{A}$ and $\mathbf{B}$. First, we created error matrices E and F, with each element following a standard normal distribution:

$$\mathbf{E} = \delta \frac{\mathbf{E}_0}{\|\mathbf{E}_0\|_F} \|\mathbf{A}_{trur}\|_F, \quad \mathbf{F} = \delta \frac{\mathbf{F}_0}{\|\mathbf{F}_0\|_F} \|\mathbf{B}_{blur}\|_F,$$

where $\mathbf{E}_0$ and $\mathbf{F}_0$ are white Gaussian noises. We generate the noise-contaminated matrix A and the noise-contaminated matrix B as

$$\mathbf{A} = \mathbf{A}_{true} + \mathbf{E}, \qquad \mathbf{B} = \mathbf{B}_{blur} + \mathbf{F}.$$

So, consider the GTLS problem with the data and observation matrices A and B. The relative error (RE), peak signal-to-noise ratio (PSNR) [23] and Mean Square Error (MSE) are commonly used as evaluation indicators in image processing as follows:

$$RE = \frac{\|\mathbf{X} - \mathbf{X}_{true}\|_F}{\|\mathbf{X}_{true}\|_F}, \qquad PSNR = 10log_{10}\frac{(\mathbf{X}_{true}^{max})^2 I_1 I_2}{\|\mathbf{X} - \mathbf{X}_{true}\|_F^2}, \quad MSE(\mathbf{X};\mathbf{X}_{true}) = \frac{\|\mathbf{X} - \mathbf{X}_{true}\|_F}{mn},$$

where $\mathbf{X}_{true}$ and $\mathbf{X}_{true}^{max}$ are the original image and the largest component of the matrix corresponding to the original image, respectively. We applied GN-GTLS and GN-TLS [7] to solve the GTLS problem and the corresponding TLS problem, respectively, for different values of δ. The numerical results are given in Table 1 and Figures 1, 2. In this Table, the notation $\mathbf{X}^*$ indicates the SVD solution for the corresponding GTLS and TLS problems. Furthermore, $Iter_{inner}$ represents the number of iterations required in the implementation of Algorithm GLS-BTF to derive $\mathscr{D}$ and $Iter_{outer}$ signifies the number of iterations needed in the implementation of Algorithm 2 to obtain $\mathbf{X}^*$. The numerical results in Table 1 show that the GN-GTLS method is superior to GN-TLS.

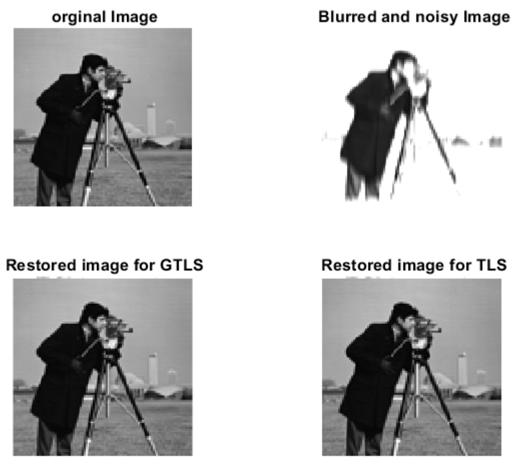

Figure 1: Blurred and deblurred results of cameraman image, with $\delta = 0.001$.

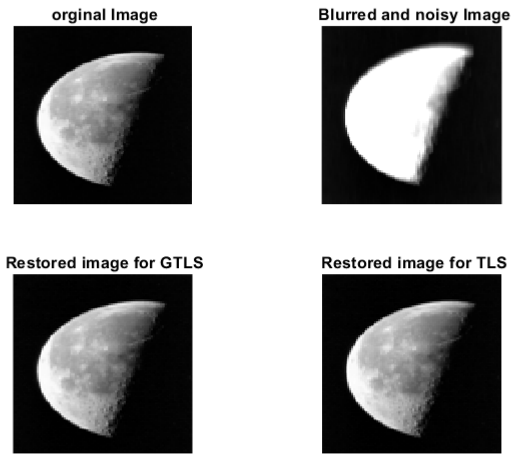

Figure 2: Blurred and deblurred results of moon image, with $\delta = 0.001$.

Table 1: The Comparison of the iteration numbers (Iter), the CPU times (CPU) for Example 1.

Algorithm	delta	Image	PSNR	MSE	RE	$\|X_k - X^*\|$	CPU	$Iter_{inner} + Iter_{outer}$
GN-GTLS	0.001	cameraman	26.66	4.4	9×10^{-2}	2.85×10^{-6}	**230.22**	182
		moon	58.52	0.11	3×10^{-3}	1.85×10^{-5}	**263.65**	172
GN-TLS	0.001	cameraman	26.66	4.5	9×10^{-2}	9.85×10^{-6}	352.35	181
		moon	58.52	0.11	3×10^{-3}	9.20×10^{-6}	339.76	173
GN-GTLS	0.01	cameraman	26.65	4.51	9×10^{-2}	2.85×10^{-6}	**239.75**	236
		moon	43.85	0.63	1×10^{-2}	9.2×10^{-7}	**276.08**	229
GN-TLS	0.01	cameraman	26.63	4.4	9×10^{-2}	9.85×10^{-6}	281.05	234
		moon	43.85	0.639	1×10^{-2}	9.5×10^{-7}	296.21	229

Example 2. [13] The GTLS model is employed to address the color image blurring problem. The process aims to improve image quality and clarity by eliminating blurring noise and restoring color images across all color components. When applying Algorithm 2 to color images, the first step involves dividing the image into separate color components. These components may include various color channels, such as red, green, and blue. The next step is implementing Algorithm 2 on each color channel, which accomplishes noise targeting and color blurring. Finally, the color components are reassembled to reconstruct the complete color image.

We set a tensor function of a color image based on the RGB color space as $\mathscr{X} = (\mathbf{X}^r \;\; \mathbf{X}^g \;\; \mathbf{X}^b) \in \mathbb{R}^{n\times n\times 3}$, where $\mathbf{X}^r, \mathbf{X}^g$ and $\mathbf{X}^b$ represent the red, green, and blue color channels, respectively. Each of these color channels is structured as an $n\times n$ matrix. For applying blurring and noise to the color images, we can blur and add noise to each color component individually. To do this, in this example, we denote $\bar{\mathscr{X}} \in \mathbb{R}^{n\times n\times 3}$ the orginal data. Let $\mathscr{A}, \mathscr{B} \in \mathbb{R}^{n\times n\times 3}$ be the blur and observation tensors, respectively, then $\mathscr{B} = \mathscr{A} \bullet \bar{\mathscr{X}}$, where $\bullet$ is defined by $\mathbf{B}^j = \mathbf{A}^j\bar{\mathbf{X}}^j$, $j = 1,2,3$, where $\bar{\mathbf{X}}^j$, $j = 1,2,3$ are color channels.

Similar to error matrices [24], we define the following errors

$$\varepsilon_{\mathbf{E}}{}^{j} = \eta \frac{\mathbf{E}_j}{\|\mathbf{E}_j\|_F} \|\mathbf{A}^j\|_F, \quad j = 1, 2, 3,$$

$$\varepsilon_{\mathbf{F}}{}^{j} = \eta \frac{\mathbf{F}_j}{\|\mathbf{F}_j\|_F} \|\mathbf{B}^j\|_F, \quad j = 1, 2, 3,$$

where $\mathbf{E}_j$ and $\mathbf{F}_j$ are white Gaussian noises and $\eta = 0.001$. Then a model of color image deblurring and denoising is proposed by

$$\begin{cases} \min\limits_{\mathscr{E},\mathscr{F}} \|(\mathscr{E}, \mathscr{F})\|_F, \\ s.t. \;\; (\mathscr{A} + \mathscr{E}) \bullet \mathscr{X} = \mathscr{B} + \mathscr{F}. \end{cases}$$

We generate the bluring tensor $\mathscr{A}$ by

$$\mathbf{A}^j = \frac{1}{j} \mathbf{A}_{true}, \quad j = 1, 2, 3,$$

where $\mathbf{A}_{true} \in \mathbb{R}^{n \times n}$ is the Toeplitz matrix generated by

$$\begin{cases} z = [exp(-([0 : band - 1]^2)/(2\sigma^2)), zeros(1, n - band)], \\ \mathbf{A}_{true} = \frac{1}{\sigma\sqrt{2\pi}} toeplitz(z(1) fliplr(z(2 : end)), z), \end{cases}$$

with $\sigma = 4$, $n = 80$ and $band = 2$. We applied GN-GTLS and GN-TLS [7] to solve the GTLS problem and the corresponding TLS problem on each color channel, respectively. The numerical results are displayed in Table 2 and Figure 3, 4.

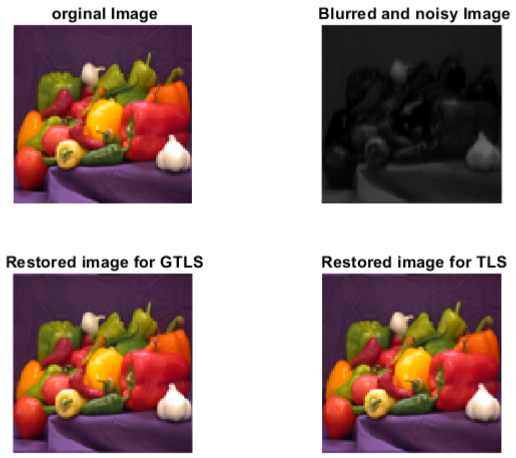

Figure 3: Blurred and deblurred results of peppers image.

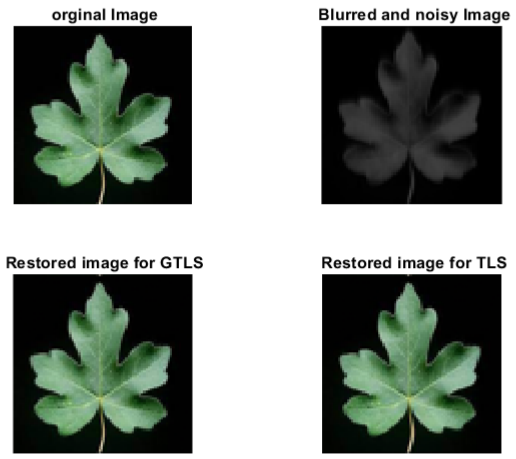

Figure 4: Blurred and deblurred results of leaf image.

Table 2: The Comparison of the iteration numbers (Iter), the CPU times (CPU) for Example 2.

Algorithm	Image	PSNR	MSE	RE	CPU	$Iter_{inner} + Iter_{outer}$
GN-GTLS	leaf	42.81	3.55	2.2×10^{-2}	**310.44**	70
	peppers	38.48	6.4	3.3×10^{-2}	**341.27**	77
GN-TLS	leaf	42.81	3.55	2.2×10^{-2}	448.32	70
	peppers	38.88	6.4	3.3×10^{-2}	447.03	77

Example 3. Consider the GTLS problem with the data and observation matrices A and B. In this example, inspired by test problem [2], we take $A = YDZ^T + \varepsilon E \in \mathbb{R}^{m\times n}$, where Y and Z are randomly orthogonal matrices and $D \in \mathbb{R}^{n\times n}$ is a diagonal matrix with $D_{ii} = 2^{1-i}$. Also, we have $B = YDZ^T X1 + \varepsilon F \in \mathbb{R}^{m\times s}$, where $X1 \in \mathbb{R}^{n\times s}$ and $X1(i,:) = \frac{1}{i}, \quad i = 1,\dots,n$. The terms $E \in \mathbb{R}^{m\times n}$ and $F \in \mathbb{R}^{m\times s}$ represent pseudo-random "noise" with entries drawn from a standard normal distribution $N(0,1)$, herein $\varepsilon = 10^{-6}$. We applied GN-GTLS and GN-TLS [7] to solve the GTLS problem and the corresponding TLS problem, respectively. The numerical results are displayed in Table 3.

Example 4. Consider the following GTLS problem $(\mathbf{A} + \varepsilon\mathbf{E})\mathbf{X} = \mathbf{B} + \varepsilon\mathbf{F}$, where

$$\mathbf{A} = \begin{bmatrix} hilb(10) & zeros(10,8) \\ eye(10) & ones(10,8) \end{bmatrix},$$

and $hilb(n)$ denote Hilbert matrix of order n. Let $\mathbf{B} = \mathbf{A}\bar{\mathbf{X}}$, where $\bar{\mathbf{X}}$ is ones$(18,4)$ and $\mathbf{F},\mathbf{E}$ represent pseudo-random noise with entries drawn from a standard normal distribution $N(0,1)$, herein $\varepsilon = 10^{-6}$. We applied GN-GTLS and GN-TLS [7] to solve the GTLS problem and the corresponding TLS problem, respectively. The result is displayed in Figure 5. In Figure 5, norm(JF) represents $\|\mathscr{J}_k^T *_2 \mathscr{G}_k\|_F$ for GTLS and $\|\mathbf{J}(\mathbf{x}_k)^T\mathbf{f}_k\|$ for TLS.

Table 3: The Comparison of the iteration numbers (Iter), the CPU times (CPU), and residual norm (RES) for Example 3.

Algorithm	(m, n, s)	$(30,20,10)$	$(40,30,15)$	$(50,30,20)$
GN-GTLS	$Iter_{inner}+Iter_{outer}$	9	37	67
	CPU	**11.22**	**30.21**	**84.62**
	RES	5.4×10^{-16}	5.29×10^{-14}	1.74×10^{-14}
GN-TLS	$Iter_{inner}+Iter_{outer}$	6	38	64
	CPU	11.43	31.24	85.22
	RES	1.52×10^{-15}	1.46×10^{-15}	1.63×10^{-14}

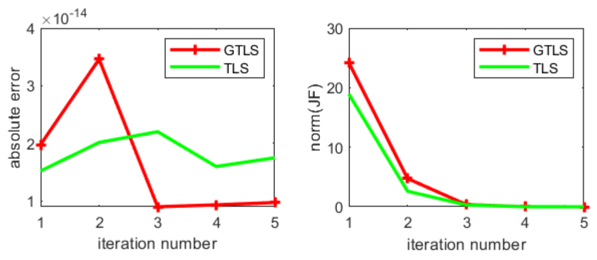

Figure 5: Numerical results for Example 4 with $\varepsilon = 10^{-6}$ and CPU time= 0.55 and 0.56 for GTLS and TLS, respectively.

6 Conclusion

This paper presents an iterative method to obtain an approximate solution for the GTLS problem (8). To this end, the first step is to propose a solution to the GTLS problem (8). To do so, we reduced the GTLS problem to an equivalent nonlinear optimization problem (9). Then, similar to the Gauss-Newton method, an approximate solution is obtained. Next, to compare the GN-GTLS method with GN-TLS, we converted the GTLS problem to the corresponding TLS problem by applying Kronecker product. The numerical results show that the GN-GTLS method is superior to GN-TLS. In addition, we applied the GN-GTLS method to restore gray-scale and color images, and the results were satisfactory.

7 Acknowledgements

The authors are grateful to anonymous referees for their valuable recommendations and comments which have improved the quality of the paper.

References

[1] M. Bagheri, A. Tajaddini, F. Kyanfar, A. Salemi, *Alternative Arnoldi process for ill-conditioned tensor equations with application to image restoration*, Comput. Appl. Math. **43** (2024) 375.

[2] A. Bjorck, P. Heggerness, P. Mathstoms, *Methods for large scale total least squares problems*, SIAM J. Matrix Anal. Appl. **22** (2000) 413–429.

[3] A. Bjorck, *Numerical Methods for Least Squares Problems*, SIAM, 1996.

[4] E.Kh. Dehdezi, S. Karimi, *A rapid and powerful iterative method for computing inverses of a parse tensors with applications*, Appl. Math. Comput. **415** (2022) 126720.

[5] B. De Moor, J. David, *Total linear least squares and the algebraic Riccati equation*, Systems Control Lett. **18** (1992) 329–337.

[6] B. De Moor, *Total least squares for affinely structured matrices and the noisy realization problem*, IEEE Trans. Signal Process. **42** (1994) 3104–3113.

[7] D. Fasino, A. Fazzi, *A Gauss-Newton iteration for total least squares problems*, BIT. **58** (2018) 281–299.

[8] H. Fu, J. Barlow, *regularized structured total least squares algorithm for high-resolution image reconstruction*, Linear Algebra Appl. **391** (2004) 75–98.

[9] G.H. Golub,*Some modified matrix eigenvalue problems*, SIAM Rev. **15** (1973) 318–344 .

[10] G.H. Golub, C.F. Van Loan. *An analysis of the total least squares problem*, SIAM J. Numer. Anal. **17** (1980) 883–893.

[11] G.H. Golub, C.F. Van Loan, *Matrix Computations*, Fourth Eidition, Johns Hopkins University Press, Baltimore, 2013.

[12] A. Graham, *Kronecker Products and Matrix Calculus with Applications*, Dover Publications, 2018.

[13] F. Han, Y. Wei, P. Xie, *Regularized and structured tensor total least squares methods with applications*, J. Optim. Theory Appl. **202** (2024) 1–36.

[14] K. Hermus, W. Verhelst, P. Lemmerling, P. Wambacq, S. Van Huffel, *Perceptual audio modeling with exponentially damped sinusoids*, Signal. Process. **85** (2005) 163–176.

[15] B. Huang, C. Ma, Global least squares methods based on tensor form to solve a class of generalized Sylvester tensor equations, Appl. Math. Comput. 369 (2020) 124892.

[16] V. Khang Nguyen, *Partial derivative of matrix function with respect to a vector variable*, Vietnam J. Math. **30** (2012) 269–279.

[17] TG. Kolda, B. W. Bader, *Tensor decompositions and application*, SIAM Rev. **51** (2009) 455–500.

[18] P. Lemmerling, N. Mastronardi, S. Van Huffel, *Efficient implementation of a structured total least squares based speech compression method*, Linear Algebra Appl. **366** (2003) 295–315.

[19] I. Markovsky, J.C. Willems, S. Van Huffel, B. De Moor, R. Pintelon, *Application of structured total least squares for system identification and model reduction*, IEEE Trans. Automat. Control. **50** (2005) 1490–1500.

[20] I. Markovsky,S. Van Huffel, B. De Moor, *A new rank revealing method for the total least squares problem*, SIAM J. Matrix Anal. Appl. **28** (2007) 810–825.

[21] M. Muhlich, R. Mester, *The role of total least squares in motion analysis*, Proceding European Conference on Computer Vision, (1998) 305-321.

[22] A. Pruessner, D. OLeary, *Blind deconvolution using a regularized structured total least norm algorithm*, SIAM J. Matrix Anal. Appl. **24** (2003) 1018–1037.

[23] A. F. Qasim, F. Meziane, R. Aspin, *Digital watermarking: Applicability for developing trust in medical imaging workflows state of the art review*, Comput. Sci. Rev. **27** (2018) 45–60.

[24] L. Reichel, U.O. Ugwu, *Tensor Arnoldi-Tikhonov and GMRES-Type methods for ill-posed problems with a t-product structure*, J. Sci. Comput. **59** (2022) 59.

[25] B. Roorda, C. Heij, *Global total least squares modeling of multivariate time series*, IEEE Trans. Automat. Control. **40** (1995) 50–63.

[26] Y. Shen, B. Li, Y. Chen, *An iterative solution of weighted total least-squares adjustment*, J. Geod. **85** (2011) 229–238.

[27] S. Van Huffel, J. Vandewalle, *Analysis and solution of the nongeneric total least squares problem*, SIAM J. Matrix Anal. Appl. **9** (1988) 360–372.

[28] S. Van Huffel, *Partial singular value decomposition algorithm*, J. Comput. Appl. Math. **33** (1990) 105–112.

[29] S. Van Huffel, J. Vandewalle *The Total Least Squares Problem: Computational Aspects and Analysis*, SIAM, 1991.

[30] P. Verboven, P. Guillaume, B. Cauberghe, E. Parloo, S. Vanlanduit, *Frequency-domain generalized total least squares identification for modal analysis*, J. Sound Vibration. **278** (2004) 21–38.

[31] A. Yeredor, *Multiple delays estimation for chirp signals using structured total least squares*, Linear Algebra Appl. **391** (2004) 261–286.

[32] H. Zare, M. Hajarian, *An efficient Gauss-Newton algorithm for solving regularized total least squares problems*, Numer. Algorithms. **89** (2022) 1049–1073.