

Article

Solving Least-Squares Problems via a Double-Optimal Algorithm and a Variant of the Karush–Kuhn–Tucker Equation for Over-Determined Systems

Chein-Shan Liu ¹, Chung-Lun Kuo ¹ and Chih-Wen Chang ^{2,*}

¹ Center of Excellence for Ocean Engineering, National Taiwan Ocean University, Keelung 20224, Taiwan; cslu@mail.ntou.edu.tw (C.-S.L.); eji1215@gmail.com (C.-L.K.)

² Department of Mechanical Engineering, National United University, Miaoli 36063, Taiwan

* Correspondence: cwchang@nuu.edu.tw

Abstract: A double optimal solution (DOS) of a least-squares problem $\mathbf{Ax} = \mathbf{b}$, $\mathbf{A} \in \mathbb{R}^{q \times n}$ with $q \neq n$ is derived in an $m + 1$ -dimensional varying affine Krylov subspace (VAKS); two minimization techniques exactly determine the $m + 1$ expansion coefficients of the solution \mathbf{x} in the VAKS. The minimal-norm solution can be obtained automatically regardless of whether the linear system is consistent or inconsistent. A new double optimal algorithm (DOA) is created; it is sufficiently time saving by inverting an $m \times m$ positive definite matrix at each iteration step, where $m \ll \min(n, q)$. The properties of the DOA are investigated and the estimation of residual error is provided. The residual norms are proven to be strictly decreasing in the iterations; hence, the DOA is absolutely convergent. Numerical tests reveal the efficiency of the DOA for solving least-squares problems. The DOA is applicable to least-squares problems regardless of whether $q < n$ or $q > n$. The Moore–Penrose inverse matrix is also addressed by adopting the DOA; the accuracy and efficiency of the proposed method are proven. The $m + 1$ -dimensional VAKS is different from the traditional m -dimensional affine Krylov subspace used in the conjugate gradient (CG)-type iterative algorithms CGNR (or CGLS) and CGRE (or Craig method) for solving least-squares problems with $q > n$. We propose a variant of the Karush–Kuhn–Tucker equation, and then we apply the partial pivoting Gaussian elimination method to solve the variant, which is better than the original Karush–Kuhn–Tucker equation, the CGNR and the CGNE for solving over-determined linear systems. Our main contribution is developing a double-optimization-based iterative algorithm in a varying affine Krylov subspace for effectively and accurately solving least-squares problems, even for a dense and ill-conditioned matrix \mathbf{A} with $q \ll n$ or $q \gg n$.

Keywords: linear least-squares problems; varying affine Krylov subspace; double optimal algorithm; minimal-norm solution; residual orthogonality; absolute convergence; Moore–Penrose inverse; Karush–Kuhn–Tucker equation

MSC: 15A06; 65F10



Citation: Liu, C.-S.; Kuo, C.-L.; Chang, C.-W. Solving Least-Squares Problems via a Double-Optimal Algorithm and a Variant of the Karush–Kuhn–Tucker Equation for Over-Determined

Systems. *Algorithms* **2024**, *17*, 211.
<https://doi.org/10.3390/a17050211>

Academic Editor: Alicia Cordero

Received: 1 April 2024

Revised: 2 May 2024

Accepted: 7 May 2024

Published: 14 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

One of the most important numerical methods to solve linear systems is the Krylov subspace method [1–7]; since the appearance of pioneering works in [8,9], it has been studied extensively. A lot of Krylov subspace methods were reported in [10–16].

Many algorithms have been used to solve the least-squares problem with a minimum norm [17,18]:

$$\min \|\mathbf{x}\| \quad \text{s.t.} \quad \mathbf{x} \in \arg \min_{\mathbf{x}} \|\mathbf{b} - \mathbf{Ax}\|, \quad (1)$$

where $\|\cdot\|$ denotes the Euclidean norm, $\mathbf{x} \in \mathbb{R}^n$ is unknown, and $\mathbf{A} \in \mathbb{R}^{q \times n}$ and $\mathbf{b} \in \mathbb{R}^q$ are given.

We are concerned with a varying affine Krylov subspace (VAKS) solution and its corresponding iterative algorithm for solving

$$\mathbf{Ax} = \mathbf{b}, \tag{2}$$

which is an over-determined system when $q > n$ and an under-determined system when $q < n$.

Most results for Equation (2) are directly extendable to Equation (1), whose solution is called the minimum-norm or pseudo-inverse solution and is given by $\mathbf{x} = \mathbf{A}^\dagger \mathbf{b}$, where \mathbf{A}^\dagger denotes the pseudo-inverse of \mathbf{A} satisfying the following Penrose equations:

$$\mathbf{AA}^\dagger \mathbf{A} = \mathbf{A}, \tag{3}$$

$$\mathbf{A}^\dagger \mathbf{AA}^\dagger = \mathbf{A}^\dagger, \tag{4}$$

$$(\mathbf{AA}^\dagger)^\top = \mathbf{AA}^\dagger, \tag{5}$$

$$(\mathbf{A}^\dagger \mathbf{A})^\top = \mathbf{A}^\dagger \mathbf{A}. \tag{6}$$

The pseudo-inverse has been investigated and applied to solve linear least-squares problems, and some computational methods have been developed to compute the Moore–Penrose inverse matrix [19–24]. In terms of the Moore–Penrose inverse matrix \mathbf{A}^\dagger , the general solution of the least-squares problem is given by

$$\mathbf{x} = \mathbf{A}^\dagger \mathbf{b} + (\mathbf{I} - \mathbf{R}_A) \mathbf{z}, \tag{7}$$

$$\mathbf{R}_A = \mathbf{A}^\dagger \mathbf{A}, \tag{8}$$

where \mathbf{z} is arbitrary. Because $\mathbf{A}^\dagger \mathbf{b}$ and $(\mathbf{I} - \mathbf{R}_A) \mathbf{z}$ are orthogonal, it follows that $\mathbf{x} = \mathbf{A}^\dagger \mathbf{b}$ is the unique solution of Equation (1).

QR factorization is a good method to solve Equation (2). The normalized Gram–Schmidt process is a modification of the original Gram–Schmidt orthogonalization by normalizing each vector after orthogonalization. Let \mathbf{Q} denote a $q \times n$ matrix obtained by applying the normalized Gram–Schmidt process on the column vectors of \mathbf{A} :

$$\mathbf{Q} := [\mathbf{q}_1, \dots, \mathbf{q}_n], \tag{9}$$

which possesses the following orthonormal property:

$$\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_n. \tag{10}$$

Hence, by $\mathbf{QRx} = \mathbf{b}$, it is easy to solve \mathbf{x} by applying the backward substitution method to

$$\mathbf{Rx} = \mathbf{Q}^\top \mathbf{b}. \tag{11}$$

However, the QR method is only applicable to an over-determined system with $q > n$. For an under-determined system with $q < n$, the QR method is not applicable because \mathbf{R} is no longer an upper triangular matrix.

Recently, many scholars have proposed many algorithms to tackle least-squares problems, like relaxed greedy randomized coordinate descent methods [25], the parallel approximate pseudo-inverse matrix technique in conjunction with the explicit preconditioned conjugate gradient least squares scheme [26], the new randomized Gauss–Seidel method [27], the QR–Cholesky method, the implicit QR method, the bidiagonal implicit QR method [28], the randomized block coordinate descent algorithm [29], an iterative pre-conditioning technique [30], splitting-based randomized iterative methods [31], the projection method [32], the index-search-method-based inner–outer iterative algorithm [33], the greedy double subspace coordinate descent method [34], a distributed algorithm [35], and multivalued projections [36].

As just mentioned, there are many algorithms for solving linear least-squares problems. For small- and medium-sized problems, one can use QR-based methods, and, if the problem is rank-deficient or ill conditioned, singular value decomposition (SVD) can be adopted [37,38]. For large-sized least-squares problems, Krylov subspace methods are more efficient than direct methods if the matrix is sparse or otherwise structured. In the present paper, we plan to develop an efficient and accurate Krylov subspace method for large-sized least-squares problems with dense and maybe ill-conditioned matrices, not limited to sparse matrices. It is also interesting that the proposed iterative algorithm can solve least-squares problems with extremal cases of $q \ll n$ or $q \gg n$. In fact, for under-determined systems, many methods are not applicable, for instance, QR and conjugate gradient (CG)-type methods, for which the basis vectors of the Krylov subspace are orthogonalized implicitly.

In addition to Equation (1), Liu [39] considered the following minimization

$$\min_x \left\{ f = \frac{\mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x}}{(\mathbf{b}^T \mathbf{A} \mathbf{x})^2} \right\} \tag{12}$$

for solving ill-posed linear systems of Equation (2) with $q = n$. Liu [40,41] employed the double optimization technique to iteratively solve inverse problems. Liu [42] replaced the Krylov subspace by the column space of \mathbf{A} and derived an effective iterative algorithm based on the maximal projection in Equation (12). These results are quite promising. However, for a non-square system (2), there exists no similar method of double optimization in the literature.

It is well known that the least-squares solution of the minimization in Equation (1) is obtained by

$$\mathbf{A}^T \mathbf{A} \mathbf{x} = \mathbf{A}^T \mathbf{b}. \tag{13}$$

When the derivative of f in Equation (12) vs. \mathbf{x} is equal to zero, we can obtain

$$\mathbf{A}^T \mathbf{A} \mathbf{x} = \frac{\mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x}}{\mathbf{b}^T \mathbf{A} \mathbf{x}} \mathbf{A}^T \mathbf{b}. \tag{14}$$

Taking the transpose of Equation (2) to $\mathbf{x}^T \mathbf{A}^T = \mathbf{b}^T$ and inserting it into Equation (14), we can immediately obtain

$$\mathbf{A}^T \mathbf{A} \mathbf{x} = \frac{\mathbf{b}^T \mathbf{A} \mathbf{x}}{\mathbf{b}^T \mathbf{A} \mathbf{x}} \mathbf{A}^T \mathbf{b}, \tag{15}$$

which implies Equation (13) by $\mathbf{b}^T \mathbf{A} \mathbf{x} / \mathbf{b}^T \mathbf{A} \mathbf{x} = 1$. Many numerical methods for least-squares problems are based on Equation (1); no numerical methods for least-squares problems are based on Equation (12).

To solve the least-squares problem (2), the usual approach is the Karush–Kuhn–Tucker equation method, which sets Equation (2) to a square linear system but with a larger dimension $q + n$. Let

$$\mathbf{r} = \mathbf{b} - \mathbf{A} \mathbf{x} \tag{16}$$

be the residual vector. By considering the enforcing constraint:

$$\mathbf{A}^T \mathbf{r} = \mathbf{0}, \tag{17}$$

which is equivalent to the normal form of Equation (2), Equations (16) and (17) constitute a $q + n$ -dimensional linear system:

$$\begin{bmatrix} \mathbf{I}_q & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}. \tag{18}$$

The advantage of the so-called Karush–Kuhn–Tucker equation [15] is that one can employ a linear solver to solve Equation (18) to obtain the least-squares solution of Equation (2); however, a disadvantage is that the dimension is enlarged to $q + n$.

A major novelty of the present paper is that we employ both Equations (1) and (12) to develop a novel numerical method in a varying affine Krylov subspace. The innovative contributions are as follows: this is the first time a new concept of varying affine Krylov subspaces has been introduced, and a double-optimal iterative algorithm has been derived based on the double optimization method to solve least-squares problems. A new variant of the Karush–Kuhn–Tucker equation can significantly improve the accuracy of solving least-squares problems by adopting the partial pivoting Gaussian elimination method.

In Section 2, the solution of Equation (2) is expanded in an $m + 1$ -dimensional varying affine Krylov subspace (VAKS), whose $m + 1$ unknown coefficients are optimized in Section 3 by two merit functions. The double optimal algorithm (DOA) is developed in Section 4; we prove a key property of absolute convergence, and the decreasing quantity of residual squares is derived explicitly. Examples of linear least-squares problems are given and compared in Section 5; in addition, Moore–Penrose inverses of non-square matrices are discussed. In Section 6, we discuss a variant of the Karush–Kuhn–Tucker equation for solving over-determined linear systems, which is compared to conjugate-gradient-type iterative methods. Finally, Section 7 describes the conclusions.

2. A Varying Affine Krylov Subspace Method

We consider an $m + 1$ -dimensional varying affine Krylov subspace (VAKS) by

$$\begin{aligned} \mathcal{K}_m &:= \text{span}\{\mathbf{A}^T \mathbf{A} \mathbf{u}_0, \dots, (\mathbf{A}^T \mathbf{A})^m \mathbf{u}_0\}, \\ \mathcal{K}'_m &:= \text{span}\{\mathbf{u}_0, \mathcal{K}_m\}, \end{aligned} \tag{19}$$

where

$$\mathbf{u}_0 := \mathbf{A}^T \mathbf{b} \in \mathbb{R}^n, \tag{20}$$

and \mathcal{K}'_m is equivalent to $\mathcal{K}_{m+1}(\mathbf{A}^T \mathbf{A}, \mathbf{u}_0)$.

Then, we expand the solution \mathbf{x} of Equation (2) in \mathcal{K}'_m by

$$\mathbf{x} = \alpha_0 \mathbf{u}_0 + \sum_{k=1}^m \alpha_k \mathbf{u}_k \in \mathcal{K}'_m, \tag{21}$$

where $\mathbf{u}_k := (\mathbf{A}^T \mathbf{A})^k \mathbf{u}_0$, $k = 1, \dots, m$, and α_0 and α_k are constants to be determined. We take $m \ll \min(n, q)$.

Let

$$\mathbf{U} := [\mathbf{u}_1, \dots, \mathbf{u}_m] = [\mathbf{A}^T \mathbf{A} \mathbf{u}_0, \dots, (\mathbf{A}^T \mathbf{A})^m \mathbf{u}_0] \tag{22}$$

be an $n \times m$ matrix. The Arnoldi process [15] as a normalized Gram–Schmidt process is used to normalize and orthogonalize the Krylov vectors $(\mathbf{A}^T \mathbf{A})^j \mathbf{u}_0$, $j = 1, \dots, m$, such that the resultant vectors \mathbf{u}_i , $i = 1, \dots, m$ satisfy $\mathbf{u}_i^T \mathbf{u}_j = \delta_{ij}$, $i, j = 1, \dots, m$, where δ_{ij} is the Kronecker delta symbol. Select m , give an initial $\mathbf{v}_0 = \mathbf{A}^T \mathbf{A} \mathbf{u}_0$, and compute $\mathbf{u}_1 = \frac{\mathbf{v}_0}{\|\mathbf{v}_0\|}$:

$$\begin{aligned} &\text{Do } j = 1 : m - 1, \\ &\quad \mathbf{w}_j = \mathbf{A}^T \mathbf{A} \mathbf{u}_j, \\ &\quad \text{Do } i = 1 : j, \\ &\quad \quad h_{ij} = \mathbf{u}_i^T \mathbf{w}_j, \\ &\quad \quad \mathbf{w}_j = \mathbf{w}_j - h_{ij} \mathbf{u}_i, \\ &\quad \text{End do of } i, \\ &\quad h_{j+1,j} = \|\mathbf{w}_j\|; \text{ if } h_{j+1,j} = 0 \text{ stop,} \\ &\quad \mathbf{u}_{j+1} = \frac{\mathbf{w}_j}{h_{j+1,j}}, \\ &\quad \text{End do of } j. \end{aligned} \tag{23}$$

Hence, Equation (21) can be written as

$$\mathbf{x} = \alpha_0 \mathbf{u}_0 + \mathbf{U}\boldsymbol{\alpha}, \tag{24}$$

where $\boldsymbol{\alpha} := (\alpha_1, \dots, \alpha_m)^\top$ and \mathbf{U} satisfies the orthonormal property:

$$\mathbf{U}^\top \mathbf{U} = \mathbf{I}_m. \tag{25}$$

For a consistent system, the exact solution \mathbf{x}^* satisfies $\mathbf{A}\mathbf{x}^* = \mathbf{b}$ exactly. For an inconsistent system, the exact solution \mathbf{x}^* exactly satisfies Equation (1) given by $\mathbf{x}^* = \mathbf{A}^\dagger \mathbf{b}$, because the equality in $\mathbf{A}\mathbf{x}^* = \mathbf{b}$ no longer holds.

On this occasion, we briefly recall the m -dimensional conjugate gradient (CG)-type iterative method with an approximation to Equation (2) with $q > n$. The CGNR (or CGLS) minimizes the energy function of the error [15,43]:

$$f(\mathbf{x}) = (\mathbf{A}^\top \mathbf{A}(\mathbf{x}^* - \mathbf{x}), (\mathbf{x}^* - \mathbf{x})), \tag{26}$$

where (\mathbf{a}, \mathbf{b}) denotes the inner product of vectors \mathbf{a} and \mathbf{b} , and \mathbf{x}^* is the exact solution of Equation (2). Overall, \mathbf{x} in the affine Krylov subspace (AKS) is:

$$\mathbf{x}_0 + \mathcal{K}_m(\mathbf{A}^\top \mathbf{A}, \mathbf{A}^\top \mathbf{r}_0) := \mathbf{x}_0 + \text{span}\{\mathbf{A}^\top \mathbf{r}_0, \mathbf{A}^\top \mathbf{A} \mathbf{A}^\top \mathbf{r}_0, \dots, (\mathbf{A}^\top \mathbf{A})^{m-1} \mathbf{A}^\top \mathbf{r}_0\}, \tag{27}$$

where $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$. There is some extension of the CGNR (or CGLS) to bounded perturbation resilient iterative methods [44].

Craig’s idea was to seek the solution of Equation (2) with $q > n$ by

$$\mathbf{x} = \arg \min_{\mathbf{z} \in \mathbb{R}^n} \|\mathbf{z}\| \quad \text{subject to} \quad \mathbf{A}\mathbf{z} = \mathbf{b}. \tag{28}$$

By introducing $\mathbf{x} = \mathbf{A}^\top \mathbf{y}$, we seek \mathbf{y} in the AKS by

$$\mathbf{y}_0 + \mathcal{K}_m(\mathbf{A}\mathbf{A}^\top, \mathbf{r}_0), \tag{29}$$

to minimize

$$f(\mathbf{y}) = (\mathbf{A}\mathbf{A}^\top(\mathbf{y}^* - \mathbf{y}), (\mathbf{y}^* - \mathbf{y})), \tag{30}$$

where \mathbf{y}^* is the exact solution of $\mathbf{A}\mathbf{A}^\top \mathbf{y} = \mathbf{b}$. The above statement for \mathbf{x}^* is also valid for \mathbf{y}^* with $\mathbf{A}\mathbf{x}^* = \mathbf{b}$ replaced by $\mathbf{A}\mathbf{A}^\top \mathbf{y}^* = \mathbf{b}$. The resulting CG iterative algorithm is known as the CGNE (or the Craig method), which minimizes the error function [15,43].

Notice that the affine Krylov subspaces in Equations (27) and (29) are fixed AKSs upon setting \mathbf{x}_0 and \mathbf{y}_0 , while in Equation (21), the AKS is varying, denoted as a VAKS, owing to the translation vector $\alpha_0 \mathbf{u}_0$ not being fixed to \mathbf{u}_0 :

$$\mathcal{K}'_m = \mathcal{K}_{m+1}(\mathbf{A}^\top \mathbf{A}, \mathbf{A}^\top \mathbf{b}) = \text{span}\{\mathbf{A}^\top \mathbf{b}, \mathbf{A}^\top \mathbf{A} \mathbf{A}^\top \mathbf{b}, \dots, (\mathbf{A}^\top \mathbf{A})^m \mathbf{A}^\top \mathbf{b}\}. \tag{31}$$

Upon comparing Equation (21), whose coefficient α_0 before \mathbf{u}_0 is an unknown constant, to Equations (27) and (29), whose coefficients before \mathbf{x}_0 and \mathbf{y}_0 are fixed to one, the affine Krylov subspace in Equation (21) is a varying affine Krylov subspace (VAKS). The dimension of the VAKS is one more than that of the AKS.

3. A Double Optimal Solution

In this section, we adopt the new concept of the VKAS to derive a double optimal solution of Equation (2). The starting point is Equation (21), rather than $\mathbf{x} = \mathbf{u}_0 + \sum_{k=1}^m \alpha_k \mathbf{u}_k \in \mathbf{u}_0 + \mathcal{K}_m$.

3.1. Two Minimized Functions

Let

$$\mathbf{y} := \mathbf{A}\mathbf{x}, \tag{32}$$

such that the approximation to \mathbf{b} in Equation (2) is accounted by the error vector:

$$\mathbf{e} := \mathbf{b} - \mathbf{b}^T \left(\frac{\mathbf{y}}{\|\mathbf{y}\|} \right) \frac{\mathbf{y}}{\|\mathbf{y}\|}; \tag{33}$$

the squared-norm of \mathbf{e} reads as

$$\|\mathbf{e}\|^2 = \|\mathbf{b}\|^2 - \frac{(\mathbf{b}^T \mathbf{y})^2}{\|\mathbf{y}\|^2} \geq 0. \tag{34}$$

We maximize

$$\max_{\mathbf{y}} \left\{ \frac{(\mathbf{b}^T \mathbf{y})^2}{\|\mathbf{y}\|^2} \right\} \tag{35}$$

as the maximal projection of \mathbf{b} on the direction $\mathbf{y}/\|\mathbf{y}\|$, such that $\|\mathbf{e}\|^2$ is minimized. Equation (35) is equivalent to the minimization of

$$\min_{\mathbf{y}} \left\{ f := \frac{\|\mathbf{y}\|^2}{(\mathbf{b}^T \mathbf{y})^2} \right\} \geq \frac{1}{\|\mathbf{b}\|^2}. \tag{36}$$

Let \mathbf{J} be

$$\mathbf{J} := \mathbf{A}\mathbf{U} \in \mathbb{R}^{q \times m}. \tag{37}$$

Then, with the aid of Equation (24), Equation (32) can be written as

$$\mathbf{y} = \mathbf{y}_0 + \mathbf{J}\boldsymbol{\alpha}, \tag{38}$$

where

$$\mathbf{y}_0 := \alpha_0 \mathbf{A}\mathbf{u}_0 \in \mathbb{R}^q. \tag{39}$$

Inserting Equation (38) for \mathbf{y} into Equation (36) yields

$$\min_{\boldsymbol{\alpha}} \left\{ f = \frac{\|\mathbf{y}\|^2}{(\mathbf{b}^T \mathbf{y})^2} = \frac{\|\mathbf{y}_0 + \mathbf{J}\boldsymbol{\alpha}\|^2}{(\mathbf{b}^T \mathbf{y}_0 + \mathbf{b}^T \mathbf{J}\boldsymbol{\alpha})^2} \right\}, \tag{40}$$

which is used to find $\boldsymbol{\alpha}$. However, α_0 in $\mathbf{y}_0 = \alpha_0 \mathbf{A}\mathbf{u}_0$ is still an unknown value to be determined by imposing the minimization of squared-norm of residuals, shortened to the residual square:

$$\min_{\alpha_0} \{ \|\mathbf{r}\|^2 = \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2 \}. \tag{41}$$

3.2. Mathematical Preliminaries

To determine the unknowns α_j , $j = 0, 1, \dots, m$, the following main theorem is an extension of [39] to a non-square system (2).

Theorem 1. Assume that $m_0 = \text{rank}(\mathbf{A}) \leq \min(n, q)$. \mathbf{U} satisfies Equation (25) with $m < m_0$. $\mathbf{x} \in \mathcal{K}'_m$, approximately satisfying Equation (2) derived from two minimizations (40) and (41), is a double optimal solution (DOS), given by

$$\mathbf{x} = \alpha_0(\mathbf{u}_0 - \mathbf{V}\mathbf{A}\mathbf{u}_0 + \lambda_0 \mathbf{V}\mathbf{b}), \tag{42}$$

where

$$\begin{aligned} \mathbf{u}_0 &= \mathbf{A}^T \mathbf{b}, \mathbf{J} = \mathbf{A}\mathbf{U}, \mathbf{C} = \mathbf{J}^T \mathbf{J}, \mathbf{D} = \mathbf{C}^{-1}, \mathbf{V} = \mathbf{U}\mathbf{D}\mathbf{J}^T, \mathbf{P} = \mathbf{A}\mathbf{V} = \mathbf{J}\mathbf{D}\mathbf{J}^T, \\ \lambda_0 &= \frac{\|\mathbf{A}\mathbf{u}_0\|^2 - \mathbf{u}_0^T \mathbf{A}^T \mathbf{P}\mathbf{A}\mathbf{u}_0}{\mathbf{b}^T \mathbf{A}\mathbf{u}_0 - \mathbf{b}^T \mathbf{P}\mathbf{A}\mathbf{u}_0}, \end{aligned} \tag{43}$$

$$\mathbf{w} = \lambda_0 \mathbf{P}\mathbf{b} + \mathbf{A}\mathbf{u}_0 - \mathbf{P}\mathbf{A}\mathbf{u}_0, \tag{44}$$

$$\alpha_0 = \frac{\mathbf{w}^T \mathbf{b}}{\|\mathbf{w}\|^2}. \tag{45}$$

To guarantee the existence of $\mathbf{D} = \mathbf{C}^{-1}$, the dimension m of \mathcal{K}_m follows $m < m_0$.

Proof. In view of Equation (38), we can write

$$\mathbf{b}^T \mathbf{y} = \mathbf{b}^T \mathbf{y}_0 + \mathbf{b}^T \mathbf{J} \boldsymbol{\alpha}, \tag{46}$$

$$\|\mathbf{y}\|^2 = \|\mathbf{y}_0\|^2 + 2\mathbf{y}_0^T \mathbf{J} \boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{C} \boldsymbol{\alpha} \tag{47}$$

for f in Equation (40), where

$$\mathbf{C} := \mathbf{J}^T \mathbf{J} \tag{48}$$

is an $m \times m$ positive definite matrix; hence, \mathbf{C}^{-1} exists. For the rank deficiency matrix \mathbf{A} , we suppose that its rank is $m_0 = \text{rank}(\mathbf{A})$, where $m_0 < \min(n, q)$; to guarantee that \mathbf{C} is an $m \times m$ positive definite matrix, m must be smaller than m_0 .

The minimality condition of Equation (40) is

$$(\mathbf{b}^T \mathbf{y})^2 \nabla_{\boldsymbol{\alpha}} \|\mathbf{y}\|^2 - 2\mathbf{b}^T \mathbf{y} \|\mathbf{y}\|^2 \nabla_{\boldsymbol{\alpha}} (\mathbf{b}^T \mathbf{y}) = \mathbf{0}; \tag{49}$$

$\nabla_{\boldsymbol{\alpha}}$ is the gradient with respect to $\boldsymbol{\alpha}$. Cancelling $\mathbf{b}^T \mathbf{y}$ in Equation (49) yields

$$\mathbf{b}^T \mathbf{y} \mathbf{y}_2 - 2\|\mathbf{y}\|^2 \mathbf{y}_1 = \mathbf{0}, \tag{50}$$

where

$$\mathbf{y}_1 := \nabla_{\boldsymbol{\alpha}} (\mathbf{b}^T \mathbf{y}) = \mathbf{J}^T \mathbf{b}, \tag{51}$$

$$\mathbf{y}_2 := \nabla_{\boldsymbol{\alpha}} \|\mathbf{y}\|^2 = 2\mathbf{J}^T \mathbf{y}_0 + 2\mathbf{C} \boldsymbol{\alpha}. \tag{52}$$

From Equation (50), \mathbf{y}_2 is observed to be proportional to \mathbf{y}_1 in the m -dimensional space \mathbb{R}^m , which is supposed to be

$$\mathbf{y}_2 = \frac{2\|\mathbf{y}\|^2}{\mathbf{b}^T \mathbf{y}} \mathbf{y}_1 = 2\lambda \mathbf{y}_1, \tag{53}$$

where 2λ is a proportional factor determined below. The second equality leads to

$$\|\mathbf{y}\|^2 = \lambda \mathbf{b}^T \mathbf{y}. \tag{54}$$

Then, by Equations (51)–(53), we have

$$\boldsymbol{\alpha} = \lambda \mathbf{D} \mathbf{J}^T \mathbf{b} - \mathbf{D} \mathbf{J}^T \mathbf{y}_0, \tag{55}$$

where

$$\mathbf{D} := \mathbf{C}^{-1} = (\mathbf{J}^T \mathbf{J})^{-1} > \mathbf{0} \tag{56}$$

is an $m \times m$ matrix. It follows from Equations (46), (47) and (55) that

$$\mathbf{b}^T \mathbf{y} = \mathbf{b}^T \mathbf{y}_0 + \lambda \mathbf{b}^T \mathbf{P} \mathbf{b} - \mathbf{b}^T \mathbf{P} \mathbf{y}_0, \tag{57}$$

$$\|\mathbf{y}\|^2 = \lambda^2 \mathbf{b}^T \mathbf{P} \mathbf{b} + \|\mathbf{y}_0\|^2 - \mathbf{y}_0^T \mathbf{P} \mathbf{y}_0, \tag{58}$$

where

$$\mathbf{P} := \mathbf{J} \mathbf{D} \mathbf{J}^T \tag{59}$$

is a $q \times q$ positive semi-definite matrix. \mathbf{P} is indeed a projection operator.

From Equations (54), (57) and (58), λ is solved from

$$\|\mathbf{y}_0\|^2 - \mathbf{y}_0^T \mathbf{P} \mathbf{y}_0 = \lambda (\mathbf{b}^T \mathbf{y}_0 - \mathbf{b}^T \mathbf{P} \mathbf{y}_0) \tag{60}$$

as follows:

$$\lambda = \frac{\|\mathbf{y}_0\|^2 - \mathbf{y}_0^T \mathbf{P} \mathbf{y}_0}{\mathbf{b}^T \mathbf{y}_0 - \mathbf{b}^T \mathbf{P} \mathbf{y}_0}. \tag{61}$$

Inserting this into Equation (55) generates

$$\alpha = \frac{\|y_0\|^2 - y_0^T P y_0}{b^T b y_0 - b^T P y_0} D J^T b - D J^T y_0. \tag{62}$$

By inserting the above α into Equation (24) and using Equation (39) for y_0 , we can obtain

$$x = \alpha_0(u_0 - V A u_0 + \lambda_0 V b), \tag{63}$$

where

$$V := U D J^T, \tag{64}$$

$$\lambda_0 := \frac{\|A u_0\|^2 - u_0^T A^T P A u_0}{b^T A u_0 - b^T P A u_0}. \tag{65}$$

Upon letting

$$v := u_0 - V A u_0 + \lambda_0 V b, \tag{66}$$

x in Equation (63) can be expressed as

$$x = \alpha_0 v, \tag{67}$$

where α_0 is determined by inserting Equation (67) into Equation (41):

$$\|b - A x\|^2 = \|\alpha_0 A v - b\|^2 = \alpha_0^2 \|w\|^2 - 2\alpha_0 w^T b + \|b\|^2, \tag{68}$$

where

$$w := A v = A u_0 - A V A u_0 + \lambda_0 A V b. \tag{69}$$

In view of Equations (37) and (64), P in Equation (59) can be written as

$$P = A V, \tag{70}$$

such that w can be written as

$$w = \lambda_0 P b + A u_0 - P A u_0.$$

This is just w in Equation (44).

Set the differential of Equation (68) vs. α_0 to zero, obtaining

$$\alpha_0 = \frac{w^T b}{\|w\|^2}; \tag{71}$$

hence, the proof of Equation (42) in Theorem 1 is complete. \square

In the proof of Theorem 1, we have used the minimality condition of Equation (40) to determine α in Equation (62), and then the minimality condition of Equation (41) was used to determine α_0 in Equation (71). Two optimizations were used to derive x in Equation (42), which is thus named a double optimal solution (DOS).

Theorem 2. *The two factors λ_0 in Equation (43) and α_0 in Equation (45) satisfy*

$$\alpha_0 \lambda_0 = 1. \tag{72}$$

Proof. It follows from Equations (59) and (56) that

$$P^2 = P; \tag{73}$$

hence, P is a projection operator.

In Equation (44), \mathbf{w} can be written as

$$\mathbf{w} = \lambda_0 \mathbf{Pb} + (\mathbf{I}_q - \mathbf{P})\mathbf{Au}_0. \tag{74}$$

Utilizing Equation (73), $\mathbf{P}^T = \mathbf{P}$, and $(\mathbf{I}_q - \mathbf{P})^2 = \mathbf{I}_q - \mathbf{P}$, it follows that

$$\|\mathbf{w}\|^2 = \lambda_0^2 \mathbf{b}^T \mathbf{Pb} + \mathbf{u}_0^T \mathbf{A}^T \mathbf{Au}_0 - \mathbf{u}_0^T \mathbf{A}^T \mathbf{PAu}_0, \tag{75}$$

$$\mathbf{b}^T \mathbf{w} = \lambda_0 \mathbf{b}^T \mathbf{Pb} + \mathbf{b}^T \mathbf{Au}_0 - \mathbf{b}^T \mathbf{PAu}_0. \tag{76}$$

With the aid of Equation (65), Equation (76) is further reduced to

$$\mathbf{b}^T \mathbf{w} = \lambda_0 \mathbf{b}^T \mathbf{Pb} + \frac{1}{\lambda_0} (\mathbf{u}_0^T \mathbf{A}^T \mathbf{Au}_0 - \mathbf{u}_0^T \mathbf{A}^T \mathbf{PAu}_0). \tag{77}$$

Now, after inserting Equation (77) into Equation (71), we can obtain

$$\alpha_0 \lambda_0 \|\mathbf{w}\|^2 = \lambda_0^2 \mathbf{b}^T \mathbf{Pb} + \mathbf{u}_0^T \mathbf{A}^T \mathbf{Au}_0 - \mathbf{u}_0^T \mathbf{A}^T \mathbf{PAu}_0, \tag{78}$$

which is just equal to $\|\mathbf{w}\|^2$ by Equation (75). Hence, we obtain Equation (72) readily. The proof of $\alpha_0 \lambda_0 = 1$ is complete. \square

Remark 1. The DOS is derived in the VAKS \mathcal{K}'_m in a way to stick onto Equation (21). If one takes $\alpha_0 = 1$ in Equation (21), \mathcal{K}'_m reduces to the usual affine Krylov subspace $\mathbf{u}_0 + \mathcal{K}_m$. However, in this situation, Theorem 1 no longer holds. If one takes $\mathbf{u}_0 = \mathbf{0}$ and works directly in the $(m + 1)$ -dimensional Krylov \mathcal{K}_{m+1} by expanding $\boldsymbol{\alpha}$ in an $(m + 1)$ -dimensional subspace, it would result in failure, because after inserting $\mathbf{y}_0 = \mathbf{Au}_0 = \mathbf{0}$ into Equation (62), $\boldsymbol{\alpha}$ cannot be defined. We can conclude that the DOS directly benefits from the VAKS.

The following corollaries are proven to help us understand the properties of DOS whose bases appeared in Theorems 1 and 2 and equations appeared in the proof of Theorem 1.

Corollary 1. The multiplier λ defined by Equation (61) satisfies

$$\lambda = 1, \tag{79}$$

and the double optimal solution (42) can be written as

$$\mathbf{x} = \alpha_0 \mathbf{u}_0 - \alpha_0 \mathbf{VAu}_0 + \mathbf{Vb}. \tag{80}$$

Proof. Inserting Equation (39) for \mathbf{y}_0 into Equation (61), we have

$$\lambda = \alpha_0 \frac{\|\mathbf{Au}_0\|^2 - \mathbf{u}_0^T \mathbf{A}^T \mathbf{PAu}_0}{\mathbf{b}^T \mathbf{Au}_0 - \mathbf{b}^T \mathbf{PAu}_0}, \tag{81}$$

in view of Equation (65), which can be written as

$$\lambda = \alpha_0 \lambda_0. \tag{82}$$

Hence, $\lambda = 1$ follows immediately from Equation (72). Then, using Equations (72) and (42) proves Equation (80). \square

Remark 2. Inserting Equation (20) for \mathbf{u}_0 into Equation (80) yields

$$\mathbf{x} = [\mathbf{V} + \alpha_0 \mathbf{A}^T - \alpha_0 \mathbf{VAA}^T] \mathbf{b},$$

which is a pseudo-inverse solution obtained from the DOS.

Corollary 2. *If $m = n < q$ and $\mathbf{A}^T \mathbf{A}$ is non-singular, then the double optimal solution \mathbf{x} of Equation (2) is the Penrose solution, given by*

$$\mathbf{x} = \mathbf{A}^\dagger \mathbf{b}, \tag{83}$$

where \mathbf{A}^\dagger is the Moore–Penrose pseudo-inverse of \mathbf{A} .

Proof. If $m = n$, then $\mathbf{J} = \mathbf{A}\mathbf{U}$ defined by Equation (37) is a $q \times n$ matrix. Meanwhile, \mathbf{U} defined by Equation (22) is a rank n orthogonal matrix due to Equation (25):

$$\mathbf{U}^T \mathbf{U} = \mathbf{U} \mathbf{U}^T = \mathbf{I}_n. \tag{84}$$

From Equations (56) and (84), it follows that

$$\mathbf{D} = \mathbf{U}^T (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{U}. \tag{85}$$

Inserting $\mathbf{J} = \mathbf{A}\mathbf{U}$, $\mathbf{V} = \mathbf{U}\mathbf{D}\mathbf{J}^T$ and Equation (85) into Equation (80), we have

$$\mathbf{x} = \alpha_0 \mathbf{u}_0 + \mathbf{U} \mathbf{U}^T (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{U} \mathbf{U}^T \mathbf{A}^T \mathbf{b} - \alpha_0 \mathbf{U} \mathbf{U}^T (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{U} \mathbf{U}^T \mathbf{A}^T \mathbf{A} \mathbf{u}_0. \tag{86}$$

By using Equation (84), we can derive

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}, \tag{87}$$

which is a Penrose solution of Equation (2). \square

Corollary 2 reveals that the DOS can give the pseudo-inverse solution. SVD can offer a pseudo-inverse solution as follows [37]. In SVD, the matrix \mathbf{A} is decomposed by

$$\mathbf{A} = \mathbf{U} \mathbf{D} \mathbf{V}^T, \tag{88}$$

where \mathbf{U} is a $q \times q$ orthonormal matrix, \mathbf{V} is an $n \times n$ orthonormal matrix, and \mathbf{D} is a $q \times n$ pseudo-diagonal matrix with non-zero singular values on the diagonal and all other elements are zero. Suppose that $\mathbf{A}^T \mathbf{A}$ is non-singular. Then, Equation (87) is a pseudo-inverse solution, which in terms of SVD can be derived as follows:

$$\mathbf{x} = \mathbf{V} (\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{U}^T \mathbf{b} = \mathbf{V} \mathbf{D}^\dagger \mathbf{U}^T \mathbf{b}, \tag{89}$$

letting

$$\mathbf{A}^\dagger = \mathbf{V} \mathbf{D}^\dagger \mathbf{U}^T, \tag{90}$$

we can derive Equation (83) in Corollary 2. Mathematically, the DOS is equivalent to the SVD solution in Equation (89), even though they are different approaches to the least-squares problem.

3.3. Estimation of Residual Error

Equation (80) is written as

$$\mathbf{x} = \alpha_0 (\mathbf{u}_0 - \mathbf{V} \mathbf{A} \mathbf{u}_0) + \mathbf{V} \mathbf{b}, \tag{91}$$

where

$$\alpha_0 = \frac{1}{\lambda_0} = \frac{\mathbf{b}^T \mathbf{A} \mathbf{u}_0 - \mathbf{b}^T \mathbf{P} \mathbf{A} \mathbf{u}_0}{\|\mathbf{A} \mathbf{u}_0\|^2 - \mathbf{u}_0^T \mathbf{A}^T \mathbf{P} \mathbf{A} \mathbf{u}_0}, \tag{92}$$

by using Equations (65) and (72).

Let us investigate the residual square:

$$\|\mathbf{b} - \mathbf{A} \mathbf{x}\|^2 = \|\mathbf{y} - \mathbf{b}\|^2 = \|\mathbf{y}\|^2 - 2\mathbf{b}^T \mathbf{y} + \|\mathbf{b}\|^2, \tag{93}$$

where

$$\mathbf{y} = \mathbf{Ax} = \alpha_0 \mathbf{Au}_0 - \alpha_0 \mathbf{PAu}_0 + \mathbf{Pb}. \tag{94}$$

is obtained from Equation (91) by using Equation (70).

From Equation (94), it follows that

$$\|\mathbf{y}\|^2 = \alpha_0^2 (\mathbf{u}_0^T \mathbf{A}^T \mathbf{Au}_0 - \mathbf{u}_0^T \mathbf{A}^T \mathbf{PAu}_0) + \mathbf{b}^T \mathbf{Pb}, \tag{95}$$

$$\mathbf{b}^T \mathbf{y} = \alpha_0 \mathbf{b}^T \mathbf{Au}_0 - \alpha_0 \mathbf{b}^T \mathbf{PAu}_0 + \mathbf{b}^T \mathbf{Pb}. \tag{96}$$

In the first equation, we take Equation (73) into account. Inserting them into Equation (93) leads to

$$\|\mathbf{b} - \mathbf{Ax}\|^2 = \alpha_0^2 (\mathbf{u}_0^T \mathbf{A}^T \mathbf{Au}_0 - \mathbf{u}_0^T \mathbf{A}^T \mathbf{PAu}_0) + 2\alpha_0 (\mathbf{b}^T \mathbf{PAu}_0 - \mathbf{b}^T \mathbf{Au}_0) + \|\mathbf{b}\|^2 - \mathbf{b}^T \mathbf{Pb}. \tag{97}$$

Finally, inserting Equation (92) for α_0 into the above equation yields

$$\|\mathbf{b} - \mathbf{Ax}\|^2 = \|\mathbf{b}\|^2 - \mathbf{b}^T \mathbf{Pb} - \frac{(\mathbf{b}^T \mathbf{Au}_0 - \mathbf{b}^T \mathbf{PAu}_0)^2}{\mathbf{u}_0^T \mathbf{A}^T \mathbf{Au}_0 - \mathbf{u}_0^T \mathbf{A}^T \mathbf{PAu}_0}. \tag{98}$$

If $\mathbf{P} \approx \mathbf{I}_q$, the value of $\|\mathbf{b} - \mathbf{Ax}\|^2$ in Equation (98) is made as small as possible:

$$\|\mathbf{b} - \mathbf{Ax}\|^2 \approx 0. \tag{99}$$

Theorem 3. For the double optimal solution $\mathbf{x} \in \mathcal{K}'_m$ derived in Equation (42) of the least-squares problem (2), the squared-norm of the error vector \mathbf{e} :

$$\|\mathbf{e}\|^2 = \|\mathbf{b}\|^2 - \frac{(\mathbf{b}^T \mathbf{Ax})^2}{\|\mathbf{Ax}\|^2}, \tag{100}$$

and the squared-norm of the residual vector $\mathbf{r} = \mathbf{b} - \mathbf{Ax}$ are equal, i.e.,

$$\|\mathbf{e}\|^2 = \|\mathbf{b} - \mathbf{Ax}\|^2. \tag{101}$$

Moreover,

$$\|\mathbf{r}\| < \|\mathbf{b}\|. \tag{102}$$

Proof. Inserting Equation (92) for α_0 into Equations (95) and (96), we can obtain

$$\|\mathbf{y}\|^2 = \frac{(\mathbf{b}^T \mathbf{Au}_0 - \mathbf{b}^T \mathbf{PAu}_0)^2}{\mathbf{u}_0^T \mathbf{A}^T \mathbf{Au}_0 - \mathbf{u}_0^T \mathbf{A}^T \mathbf{PAu}_0} + \mathbf{b}^T \mathbf{Pb}, \tag{103}$$

$$\mathbf{b}^T \mathbf{y} = \frac{(\mathbf{b}^T \mathbf{Au}_0 - \mathbf{b}^T \mathbf{PAu}_0)^2}{\mathbf{u}_0^T \mathbf{A}^T \mathbf{Au}_0 - \mathbf{u}_0^T \mathbf{A}^T \mathbf{PAu}_0} + \mathbf{b}^T \mathbf{Pb}. \tag{104}$$

Consequently, we have

$$\|\mathbf{y}\|^2 = \mathbf{b}^T \mathbf{y}. \tag{105}$$

It follows from Equations (98) and (103) that

$$\|\mathbf{b} - \mathbf{Ax}\|^2 = \|\mathbf{b}\|^2 - \|\mathbf{y}\|^2. \tag{106}$$

By Equations (105) and (34),

$$\|\mathbf{e}\|^2 = \|\mathbf{b}\|^2 - \|\mathbf{y}\|^2. \tag{107}$$

Equation (101) was proven upon comparing Equations (106) and (107).

Using $\mathbf{r} = \mathbf{b} - \mathbf{Ax}$ and Equations (101) and (107) leads to

$$\|\mathbf{r}\|^2 = \|\mathbf{e}\|^2 = \|\mathbf{b}\|^2 - \|\mathbf{y}\|^2. \tag{108}$$

Due to $\|\mathbf{y}\|^2 > 0$, Equation (102) follows readily. \square

4. A Numerical Algorithm

In this section, we develop an iterative algorithm to solve the least-squares problem, starting from an initial guess \mathbf{x}_0 . We assume that the value of \mathbf{x}_k at the k th step is already known, and \mathbf{x}_{k+1} is computed at the next step via the iterative algorithm. According to the value \mathbf{x}_k , the k th step residual $\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$ can be computed.

When the initial guess \mathbf{x}_0 is given, an initial residual is written as follows:

$$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0. \tag{109}$$

Upon letting

$$\mathbf{z} = \mathbf{x} - \mathbf{x}_0, \tag{110}$$

Equation (2) is equivalent to solving \mathbf{z} from

$$\mathbf{A}\mathbf{z} = \mathbf{r}_0, \tag{111}$$

which is deduced by

$$\mathbf{A}\mathbf{z} = \mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{x}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0 = \mathbf{r}_0.$$

For system (111), we seek \mathbf{z} in the VAKS by

$$\mathbf{z} = \alpha_0 \mathbf{u}_0 + \sum_{k=1}^m \alpha_k \mathbf{u}_k = \alpha_0 \mathbf{u}_0 + \mathbf{U}\boldsymbol{\alpha} \in \mathcal{K}'_m, \tag{112}$$

where

$$\mathbf{u}_0 = \mathbf{A}^T \mathbf{r}_0, \tag{113}$$

and $\mathbf{u}_k := (\mathbf{A}^T \mathbf{A})^k \mathbf{u}_0$, $k = 1, \dots, m$. The constants α_k , $k = 1, \dots, m$ and α_0 are determined by the following two minimizations:

$$\min_{\mathbf{z}} \left\{ f = \frac{\|\mathbf{A}\mathbf{z}\|^2}{(\mathbf{r}_0^T \mathbf{A}\mathbf{z})^2} \right\}, \tag{114}$$

$$\min_{\mathbf{z}} r = \|\mathbf{r}_0 - \mathbf{A}\mathbf{z}\|^2. \tag{115}$$

After inserting Equation (112) for \mathbf{z} , the first minimization can derive $\boldsymbol{\alpha}$, while the second minimization can derive α_0 .

Since two minimizations in Equations (114) and (115) are adopted to determine the descent vector \mathbf{z} in Equation (112), the resulting iterative algorithm to solve the least-square problem might be labeled as a double optimal algorithm (DOA).

To treat the rank deficient least-squares problem, the dimension m of the VAKS must be $m < m_0 = \text{rank}(\mathbf{A})$, such that $\text{rank}(\mathbf{J}_k) = m$ and $\mathbf{C}_k = \mathbf{J}_k^T \mathbf{J}_k > \mathbf{0}$ is an $m \times m$ invertible matrix. Consequently, the DOA is depicted by (i), giving \mathbf{A} , \mathbf{b} , $m < m_0$, an initial value \mathbf{x}_0 and the convergence criterion ε and (ii) doing $k = 0, 1, \dots$,

$$\begin{aligned}
 \mathbf{r}_k &= \mathbf{b} - \mathbf{A}\mathbf{x}_k, \\
 \mathbf{u}_0^k &= \mathbf{A}^T \mathbf{r}_k, \\
 \mathbf{u}_j^k &= (\mathbf{A}^T \mathbf{A})^j \mathbf{u}_0^k, \quad j = 1, \dots, m, \\
 \mathbf{U}_k &= [\mathbf{u}_1^k, \dots, \mathbf{u}_m^k] \quad (\text{By Arnoldi procedure}), \\
 \mathbf{J}_k &= \mathbf{A}\mathbf{U}_k, \\
 \mathbf{C}_k &= \mathbf{J}_k^T \mathbf{J}_k, \\
 \mathbf{D}_k &= \mathbf{C}_k^{-1}, \\
 \mathbf{V}_k &= \mathbf{U}_k \mathbf{D}_k \mathbf{J}_k^T, \\
 \mathbf{P}_k &= \mathbf{A}\mathbf{V}_k, \\
 \lambda_k &= \frac{\|\mathbf{A}\mathbf{u}_0^k\|^2 - (\mathbf{u}_0^k)^T \mathbf{A}^T \mathbf{P}_k \mathbf{A} \mathbf{u}_0^k}{\mathbf{r}_k^T \mathbf{A} \mathbf{u}_0^k - \mathbf{r}_k^T \mathbf{P}_k \mathbf{A} \mathbf{u}_0^k}, \\
 \mathbf{z}_k &= \mathbf{V}_k \mathbf{r}_k + \frac{1}{\lambda_k} (\mathbf{u}_0^k - \mathbf{V}_k \mathbf{A} \mathbf{u}_0^k), \\
 \mathbf{x}_{k+1} &= \mathbf{x}_k + \mathbf{z}_k,
 \end{aligned} \tag{116}$$

until the convergence with $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \varepsilon$ or $\|\mathbf{r}_{k+1}\| < \varepsilon$. We call $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|$ the relative residual.

In the above, \mathbf{U}_k is $n \times m$ matrix, \mathbf{J}_k is a $q \times m$ matrix, \mathbf{C}_k and \mathbf{D}_k are $m \times m$ matrices, \mathbf{V}_k is an $n \times q$ matrix and \mathbf{P}_k is a $q \times q$ matrix. The computational cost is expanded to compute \mathbf{D}_k , which is however quite time-saving because m is a small number. $\mathbf{A}^T \mathbf{A}$ is an $n \times n$ fixed matrix computed once and used in the construction of \mathbf{U}_k , which requires mn operations. \mathbf{J}_k requires mnq operations; \mathbf{C}_k requires m^2q operations; \mathbf{D}_k requires m^2q operations; \mathbf{V}_k requires nm^2q operations; and \mathbf{P}_k requires nq^2 operations. In each iteration, there are $mn + mnq + 2m^2q + nm^2q + nq^2$ operations. Denote the number of iterations by NI. The computational complexity is in total $\text{NI} \times [mn + mnq + 2m^2q + nm^2q + nq^2]$.

It is known that m is a key parameter in the Krylov subspace. A proper choice of m can significantly enhance the convergence speed and accuracy. For ill-posed linear least-squares problems, there exists the best value of m , but for well-posed linear least-squares problems, small values of m may slow down convergence. When m is increased, both the convergence speed and accuracy are increased. However, when m is increased, more computational power is required in the construction of the projection operator \mathbf{P} and the inversion matrix \mathbf{C}^{-1} .

We make some comments about the initial value of \mathbf{x}_0 . For under-determined systems, there are many solutions; hence, different choices of \mathbf{x}_0 would generate different solutions. In general, as required in Equation (1), the minimal-norm solution can be obtained by setting $\mathbf{x}_0 = \mathbf{0}$, whose norm is zero $\|\mathbf{x}_0\| = 0$. For over-determined systems, we take $\mathbf{x}_0 = \mathbf{0}$ such that the initial residual is a non zero vector $\mathbf{r}_0 = \mathbf{b}$ and the DOA based on the residual \mathbf{r}_k is solvable. Indeed, the DOA is not sensitive to the initial value of \mathbf{x}_0 ; we will take $\mathbf{x}_0 = \mathbf{0}$ for most problems unless otherwise specified.

The following corollaries help the clarification of the DOA and are based on the DOS used in the residual Equation (111).

Corollary 3. \mathbf{r}_{k+1} is orthogonal to $\mathbf{A}\mathbf{z}_k$ in Equation (111) and $\mathbf{r}_{k+1} - \mathbf{r}_k$, i.e.,

$$\mathbf{r}_{k+1}^T (\mathbf{A}\mathbf{z}_k) = 0, \tag{117}$$

$$\mathbf{r}_{k+1}^T (\mathbf{r}_{k+1} - \mathbf{r}_k) = 0. \tag{118}$$

Proof. By Equation (111), the next \mathbf{r}_{k+1} is given by

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \mathbf{y}_k, \tag{119}$$

where $\mathbf{y}_k = \mathbf{A}\mathbf{z}_k$. By Equation (116), we can also derive Equation (119) as follows:

$$\mathbf{b} - \mathbf{A}\mathbf{x}_{k+1} = \mathbf{b} - \mathbf{A}\mathbf{x}_k - \mathbf{A}\mathbf{z}_k.$$

For Equation (111), Equation (105) is written as

$$\mathbf{r}_k^T \mathbf{y}_k - \|\mathbf{y}_k\|^2 = 0; \tag{120}$$

and taking the \mathbf{y}_k inner product to Equation (119), we have

$$\mathbf{r}_{k+1}^T \mathbf{y}_k = \mathbf{r}_{k+1}^T (\mathbf{A}\mathbf{z}_k) = \mathbf{r}_k^T \mathbf{y}_k - \|\mathbf{y}_k\|^2 = 0. \tag{121}$$

Inserting $\mathbf{A}\mathbf{z}_k = \mathbf{r}_k - \mathbf{r}_{k+1}$ into Equation (117), we can prove Equation (118). The DOA is a good approximation of Equation (2) with a better descent direction \mathbf{z}_k in the VAKS. \square

Corollary 4. *For the DOA, the convergence rate is*

$$\frac{\|\mathbf{r}_k\|}{\|\mathbf{r}_{k+1}\|} = \frac{1}{\sin \theta} > 1, \quad \theta \neq \frac{\pi}{2}. \tag{122}$$

Proof. It follows from Equations (119) and (120) that

$$\|\mathbf{r}_{k+1}\|^2 = \|\mathbf{r}_k\|^2 - \|\mathbf{r}_k\| \|\mathbf{y}_k\| \cos \theta, \tag{123}$$

where θ is the intersection angle between \mathbf{r}_k and $\mathbf{y}_k = \mathbf{A}\mathbf{z}_k$. With help from Equation (120), we have

$$\cos \theta = \frac{\|\mathbf{y}_k\|}{\|\mathbf{r}_k\|}. \tag{124}$$

Obviously, $\theta \neq \pi/2$ because $\|\mathbf{y}_k\| > 0$. Then, Equation (123) can be further reduced to

$$\|\mathbf{r}_{k+1}\|^2 = \|\mathbf{r}_k\|^2 [1 - \cos^2 \theta] = \|\mathbf{r}_k\|^2 \sin^2 \theta. \tag{125}$$

Dividing both sides by $\|\mathbf{r}_{k+1}\|^2$ and taking the square-roots of both sides, we can obtain Equation (122). \square

Corollary 5. *The residual is decreased step by step,*

$$\|\mathbf{r}_{k+1}\|^2 = \|\mathbf{r}_k\|^2 - \|\mathbf{A}\mathbf{z}_k\|^2 \Rightarrow \|\mathbf{r}_{k+1}\| < \|\mathbf{r}_k\|, \tag{126}$$

Proof. Taking the squared norms of Equation (119) and using Equation (120) generates

$$\|\mathbf{r}_{k+1}\|^2 = \|\mathbf{r}_k\|^2 - \|\mathbf{y}_k\|^2; \tag{127}$$

and after inserting $\mathbf{y}_k = \mathbf{A}\mathbf{z}_k$, Equation (126) is proven. Corollary 5 is easily deduced by noting $\|\mathbf{A}\mathbf{z}_k\|^2 > 0$. \square

The property in Equation (126) is very important, which guarantees that the DOA is absolutely convergent at each iteration.

Remark 3. *At the very beginning, the DOS and DOA are developed in the VAKS \mathcal{K}'_m similar to those in Equation (21). If one takes $\alpha_0 = 1$ in Equation (21), \mathcal{K}'_m reduces to the usual affine Krylov subspace $\mathbf{u}_0 + \mathcal{K}_m$. However, in this situation, several good properties similar to those in Theorem 3 and Corollaries 3–5 will be lost; convergence cannot be guaranteed in the resulting iterative algorithm. On the other hand, if we take $\mathbf{u}_0 = \mathbf{0}$ and directly work in the $(m + 1)$ -dimensional Krylov subspace \mathcal{K}_{m+1} , by extending $\boldsymbol{\alpha}$ to an $(m + 1)$ -dimensional vector, this would result in failure because after inserting $\mathbf{y}_0 = \mathbf{A}\mathbf{u}_0 = \mathbf{0}$ into Equation (62), $\boldsymbol{\alpha}$ cannot be defined.*

Corollary 6. In the iterative algorithm (116), we may encounter a slowly varying point:

$$\mathbf{x}_{k+1} \approx \mathbf{x}_k, \tag{128}$$

when \mathbf{x}_0 is given such that $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$ satisfies $\mathbf{A}^T \mathbf{A}\mathbf{r}_0 = c_0 \mathbf{r}_0$ for some $c_0 \gg 1$.

Proof. Inserting $\mathbf{b} = \mathbf{r}_0$ and

$$\mathbf{u}_0 = \mathbf{A}^T \mathbf{r}_0$$

into Equation (92) yields

$$\alpha_0 = \frac{1}{c_0}.$$

when $c_0 \gg 1$, $\alpha_0 \approx 0$, and $\mathbf{x}_{k+1} \approx \mathbf{x}_k$ by Equation (116). \square

5. Numerical Examples

To demonstrate the efficiency and accuracy of the present iterative DOA, several examples are presented. All the numerical computations were carried out in Microsoft Developer Studio with an Intel Core I7-3770, CPU 2.80 GHz and 8 GB memory. The precision is 10^{-16} .

5.1. Example 1

In this example, we find a minimal-norm and least-squares solution of the following consistent equation:

$$\begin{bmatrix} 1 & 2 & 3 & -1 \\ 3 & 2 & 1 & -1 \\ 2 & 3 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. \tag{129}$$

Although Example 1 is simple, we use it to test the DOS and DOA for finding the minimal-norm solution for a consistent system.

First we apply the DOS with $m = 2$ to solve this problem, where we can find a quite accurate solution of $x_1 = 1.48176 \times 10^{-1}$, $x_2 = 1.92567 \times 10^{-1}$, $x_3 = 1.48141 \times 10^{-1}$, and $x_4 = -2.22689 \times 10^{-2}$, for which the three residual errors are $r_1 = 2.012 \times 10^{-6}$, $r_2 = 7.125 \times 10^{-5}$ and $r_3 = -7.51756 \times 10^{-5}$, respectively.

Although the DOS is an accurate solution with an error on the order of 10^{-5} , the accuracy can be improved by the DOA. In the application of the iterative DOA, initial guesses are given by $x_1^0 = x_2^0 = x_3^0 = x_4^0 = c_0$ for some constant. Then, the DOA with $m = 1$ solves this problem with $c_0 = 0$ and $\varepsilon = 10^{-12}$. As shown in Figure 1a, the DOA converges very fast in only seven steps. Furthermore, we can find very accurate solutions of $x_1 = 1.48148 \times 10^{-1}$, $x_2 = 1.925926 \times 10^{-1}$, $x_3 = 1.48148 \times 10^{-1}$, and $x_4 = -2.22222 \times 10^{-2}$; the norm of \mathbf{x} is $\|\mathbf{x}\| = 0.28545$. The three residual errors are $r_1 = -5.63993 \times 10^{-14}$, $r_2 = -5.61773 \times 10^{-14}$ and $r_3 = -5.63993 \times 10^{-14}$, respectively.

On the other hand, if we take $c_0 = 1$, $r_1 = 5.6 \times 10^{-13}$, $r_2 = 5.59 \times 10^{-13}$ and $r_3 = 8.47 \times 10^{-13}$ are obtained, which show that the solution $(x_1, x_2, x_3, x_4) = (0.4815, -0.2741, 0.4815, 0.3778)$ is accurate. However, the norm $\|\mathbf{x}\| = 0.82552$ is not the minimal one. Therefore, the correct one for the minimal-norm solution is $c_0 = 0$. Unless otherwise specified, we will take the initial guess by $\mathbf{x}_0 = \mathbf{0}$ for most problems computed below.

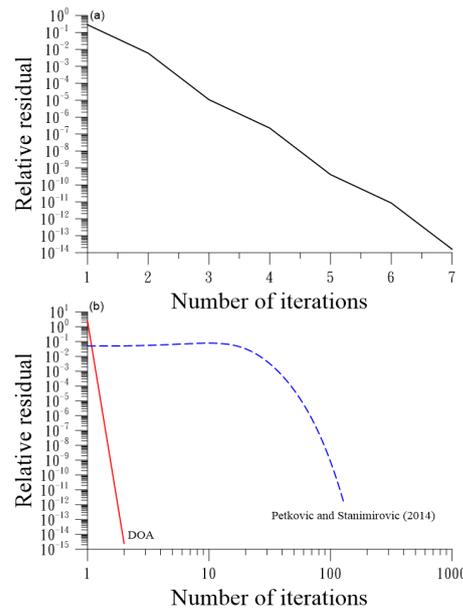


Figure 1. Examples 1 and 2, showing the relative residuals with respect to the number of iterations; the latter is compared with [20].

5.2. Example 2

A minimal norm and least-squares solution is sought for of the following inconsistent equation:

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ -1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 2 \end{bmatrix}. \tag{130}$$

Although Example 2 is simple, we use it to test the DOS and DOA for finding the minimal-norm solution for an inconsistent system.

We first apply the DOS with $m = 1$, where we obtain a very accurate solution with $x_1 = -1.25, x_2 = 1.5$, and $x_3 = 1.5$. The residuals are $(r_1, r_2, r_3, r_4) = (0.25, 0.25, 0.25, -0.25)$ with the residual norm being 0.5.

Then, the DOA with $m = 1$ and $\epsilon = 10^{-12}$ solves this problem. Starting from the initial guesses $x_1^0 = x_2^0 = x_3^0 = 0$, as shown in Figure 1b, the DOA converges very fast in only two steps, where we obtain a very accurate solution with $x_1 = -1.25, x_2 = 1.5$, and $x_3 = 1.5$. The residuals are $(r_1, r_2, r_3, r_4) = (0.25, 0.25, 0.25, -0.25)$, with the residual norm being 0.5. For the inconsistent system, the equality in Equation (130) no longer holds, such that the residual is not a zero vector.

The Moore–Penrose inverse of the coefficient matrix \mathbf{A} in Equation (130) has an exact solution:

$$\mathbf{A}^\dagger = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & -\frac{3}{4} & -\frac{1}{4} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}. \tag{131}$$

Then, we find that the exact solution of Equation (130) is given by $\mathbf{x} = \mathbf{A}^\dagger \mathbf{b}$, as just $(x_1, x_2, x_3) = (-1.25, 1.5, 1.5)$ mentioned above.

We apply the iterative method developed by Petkovic and Stanimirovic [20] to find the Moore–Penrose inverse, which has a simpler form:

$$\mathbf{X}_{k+1} = (1 + \beta)\mathbf{X}_k - \beta\mathbf{X}_k\mathbf{A}\mathbf{X}_k, \quad \mathbf{X}_0 = \beta\mathbf{A}^\dagger, \tag{132}$$

based on the Penrose Equations (4) and (6). Under the same convergence criterion, this iteration process is convergent in 130 steps, as shown in Figure 1b by a dashed line.

Note that the DOA can be also used to find the Moore–Penrose inverse \mathbf{A}^\dagger by taking the right-hand side \mathbf{b} to be a zero vector and the k -th element to be one, of which the corresponding solution of \mathbf{x} denoted by \mathbf{x}^k constitutes the k th column of \mathbf{A}^\dagger . Then, we can run the DOA q times from $k = 1, \dots, q$ to obtain

$$\mathbf{A}^\dagger = \begin{bmatrix} x_1^1 & \dots & x_1^q \\ x_2^1 & \dots & x_2^q \\ \vdots & \dots & \vdots \\ x_n^1 & \dots & x_n^q \end{bmatrix}. \tag{133}$$

The dimension of \mathbf{A}^\dagger is $n \times q$. By applying the DOA with $m = 1$ under the same convergence criterion $\varepsilon = 10^{-12}$, we can find a very accurate solution of \mathbf{A}^\dagger , as shown in Equation (131) with the maximum error of all elements being 10^{-14} , which is achieved in a total of 26 steps and converges faster than the method developed by Petkovic and Stanimirovic [20].

5.3. Example 3

We consider an over-determined Hilbert linear problem:

$$\begin{aligned} A_{ij} &= \frac{1}{i+j-1}, \quad 1 \leq i \leq q, \quad 1 \leq j \leq n, \\ b_i &= \sum_{j=1}^n \frac{1}{i+j-1} x_j, \quad 1 \leq i \leq q, \end{aligned} \tag{134}$$

where we fix $q = 6$ and $n = 5$, and the exact solutions $x_j = 1/j, 1 \leq j \leq n$ are given.

The DOA with $m = 4$ and $\varepsilon = 10^{-13}$ can solve this problem very fast, starting from the initial guesses $x_1^0 = x_2^0 = x_3^0 = x_4^0 = x_5^0 = 0$, in only four steps, as shown in Figure 2a. Furthermore, we can find a very accurate solution, as shown in Figure 2b with the maximum error (ME) being 8.91×10^{-12} . If we consider an under-determined system with $q = 5$ and $n = 6$, the DOA with $m = 3$ is still workable with $ME = 2.22 \times 10^{-8}$.

For least-squares problems, QR is a famous method to directly solve Equation (2). This process is shown in Equations (9)–(11). However, we find that $ME = 1.05 \times 10^{-6}$ obtained by QR is less accurate than that obtained via the DOA in the Hilbert least-squares problem with $q = 6$ and $n = 5$. We have checked the accuracy of the orthogonality in Equation (10) by

$$\|\mathbf{Q}^T \mathbf{Q} - \mathbf{I}_n\|,$$

which is on the order of 8.65×10^{-12} .

The Hilbert matrix is known as a notorious example of a highly ill-conditioned matrix. For this problem, the condition number is 2.5×10^5 . Then, in the floating point, \mathbf{b} has a rounding error on the the order of 10^{-16} . The algorithm introduces further floating point errors which are of the same order of magnitude. Perturbation theory for the least-squares problem [38] shows that one can only expect to find a solution that is approximately within $2 \times 10^{(5-16)} = 2 \times 10^{-11}$ of the exact solution. The error $\|\mathbf{Q}^T \mathbf{Q} - \mathbf{I}_n\| = 8.65 \times 10^{-12}$ is within this range; however, the $ME = 1.05 \times 10^{-6}$ obtained by QR is not within this range. The reason for this is that when we use Equation (11) to find the solution in the QR method, very small singular values appearing in the denominator will enlarge the rounding error and the error from \mathbf{Q} ; for example, the last two singular values 4.803×10^{-4} and 1.734×10^{-5} , upon dividing by those two small values sequentially, lead to $ME = 1.05 \times 10^{-6}$, which is not within the range of 2×10^{-11} . In contrast, the proposed DOA method with an error of 8.91×10^{-12} is within this range. For highly ill-conditioned

least-squares problems, the QR method loses some accuracy, and even \mathbf{Q} well satisfies Equation (10) within the range of 2×10^{-11} .

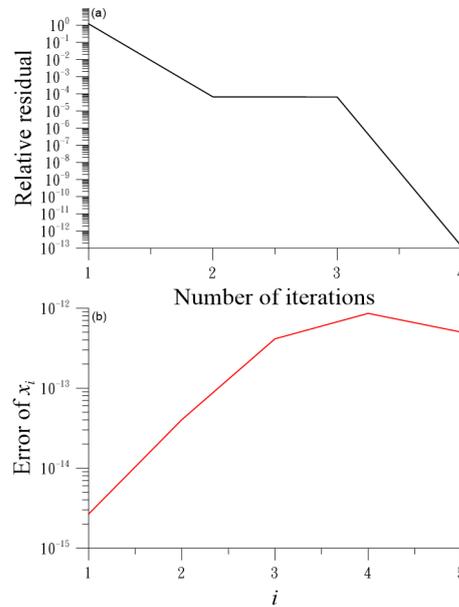


Figure 2. An over-determined Hilbert linear problem solved by the DOA: (a) relative residual, and (b) numerical error.

We raise q to 10, and Table 1 lists the computed ME for different values of n . We also list the error $\|\mathbf{Q}^T\mathbf{Q} - \mathbf{I}_n\|$ for the QR method, which is worse when n is increased. Based on Equations (88)–(90), this problem is solved by SVD. The DOA is competitive with SVD, and for all cases, it is slightly more accurate than SVD.

Table 1. Example 3: ME obtained by the DOA and QR for different n and a fixed $q = 10$.

n	2	3	4	5
$\ \mathbf{Q}^T\mathbf{Q} - \mathbf{I}_n\ $	1.63×10^{-15}	2.48×10^{-14}	3.11×10^{-13}	5.24×10^{-12}
QR	8.33×10^{-15}	3.55×10^{-12}	6.77×10^{-10}	2.27×10^{-7}
SVD	5.55×10^{-16}	1.60×10^{-14}	1.89×10^{-13}	9.09×10^{-13}
DOA	1.11×10^{-16}	3.55×10^{-15}	9.09×10^{-14}	7.81×10^{-13}

To investigate the influence of m on the DOA, Table 2 compares the ME and the iteration number (IN). For ill-posed Hilbert problems, there exists the best value $m = 3$.

Table 2. Example 3: the ME and NI obtained by the DOA for different m , where $(q, n) = (20, 8)$ and $\varepsilon = 10^{-8}$.

m	2	3	4	5	6
ME	4.94×10^{-6}	5.30×10^{-8}	1.34×10^{-6}	2.90×10^{-6}	6.28×10^{-6}
IN	14	4	4	4	5

5.4. Example 4

We find the Moore–Penrose inverse of the following rank deficient matrix [45]:

$$\mathbf{A} = \begin{bmatrix} -1 & 0 & 1 & 2 \\ -1 & 1 & 0 & -1 \\ 0 & -1 & 1 & 3 \\ 0 & 1 & -1 & -3 \\ 1 & -1 & 0 & 1 \\ 1 & 0 & -1 & -2 \end{bmatrix}. \tag{135}$$

We apply the iterative method of the DOA as demonstrated in Section 5.2 to find the Moore–Penrose inverse, which has an exact solution:

$$\mathbf{A}^\dagger = \frac{1}{102} \begin{bmatrix} -15 & -18 & 3 & -3 & 18 & 15 \\ 8 & 13 & -5 & 5 & -13 & -8 \\ 7 & 5 & 2 & -2 & -5 & -7 \\ 6 & -3 & 9 & -9 & 3 & -6 \end{bmatrix}. \tag{136}$$

Under the same convergence criterion $\varepsilon = 10^{-9}$ as used by Xia et al. [45], the iteration process of the DOA with $m = 1$ converges very fast in a total of 12 steps, as shown in Figure 3. In Table 3, we compare the DOA with other numerical methods specified by Xia et al. [45] and Petkovic and Stanimirovic [20] to assess the performance of the DOA measured by four numerical errors of the Penrose Equations (3)–(6) and the iteration number (IN).

Table 3. Computed results of a rank-deficient matrix in Example 4.

Alg.	IP	$\ \mathbf{A}\mathbf{A}^\dagger\mathbf{A} - \mathbf{A}\ ^2$	$\ \mathbf{A}^\dagger\mathbf{A}\mathbf{A}^\dagger - \mathbf{A}^\dagger\ ^2$	$\ (\mathbf{A}\mathbf{A}^\dagger)^\mathbf{T} - \mathbf{A}\mathbf{A}^\dagger\ ^2$	$\ (\mathbf{A}^\dagger\mathbf{A})^\mathbf{T} - \mathbf{A}^\dagger\mathbf{A}\ ^2$	IN
[45]	$10^{-5}\mathbf{A}^\mathbf{T}$	8.92×10^{-16}	3.36×10^{-16}	7.85×10^{-17}	1.97×10^{-15}	50
[45]	$0.1\mathbf{A}^\mathbf{T}$	9.17×10^{-20}	6.99×10^{-16}	7.85×10^{-17}	2.86×10^{-15}	60
[20]	$0.1\mathbf{A}^\mathbf{T}$	1.53×10^{-14}	9.21×10^{-17}	4.13×10^{-31}	1.17×10^{-30}	165
DOA	$\mathbf{x}_0 = 10^{-16}\mathbf{1}$	5.21×10^{-27}	2.57×10^{-29}	3.82×10^{-27}	1.61×10^{-27}	12

In the above, Alg. is the shorthand of algorithm, IP is the shorthand of initial point and IN is the shorthand of “iteration number”. $\mathbf{1} = (1, \dots, 1)^\mathbf{T}$. The first two algorithms were reported by Xia et al. [45]. It can be seen that the DOA converges much faster and is more accurate than the other algorithms.

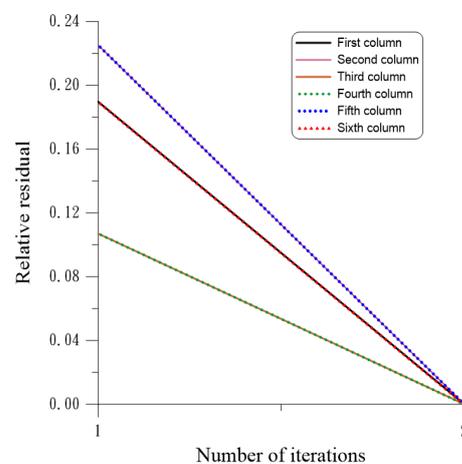


Figure 3. A computation of the Moore–Penrose inverse by the DOA, showing the relative residuals in the computation of each column of the Moore–Penrose inverse matrix.

In the computation of the pseudo-inverse, the last two singular values close to zero can be avoided to appear in the matrix $\mathbf{C} \in \mathbb{R}^{m \times m}$ if we take m small enough. For example, we take $m = 1$ in Table 3, such that $\mathbf{D} = \mathbf{C}^{-1}$ can be computed accurately without inducing a zero singular value to appear in the denominator to increase the rounding error.

5.5. Example 5

In this example, we find the Moore–Penrose inverse of the Hilbert matrix in Equation (134), which is more difficult than the previous example. Here, we fix $q = 3$, $n = 50$, and $q = 50$, $n = 3$. The numerical errors of the Penrose Equations (3)–(6) are compared in Tables 4 and 5, respectively.

Table 4. Computed results of the Hilbert matrix $q = 3, n = 50$ in Example 5.

Alg.	IP	$\ \mathbf{A}\mathbf{A}^\dagger\mathbf{A} - \mathbf{A}\ ^2$	$\ \mathbf{A}^\dagger\mathbf{A}\mathbf{A}^\dagger - \mathbf{A}^\dagger\ ^2$	$\ (\mathbf{A}\mathbf{A}^\dagger)^\top - \mathbf{A}\mathbf{A}^\dagger\ ^2$	$\ (\mathbf{A}^\dagger\mathbf{A})^\top - \mathbf{A}^\dagger\mathbf{A}\ ^2$	IN
[20]	$0.8\mathbf{A}^\top$	1.4×10^{-28}	3.7×10^{-21}	4.1×10^{-30}	3.5×10^{-27}	34
DOA	$\mathbf{x}_0 = \mathbf{0}$	9.8×10^{-28}	2×10^{-20}	4.1×10^{-24}	9.3×10^{-28}	3

Table 5. Computed results of the Hilbert matrix $q = 50, n = 3$ in Example 5.

Alg.	IP	$\ \mathbf{A}\mathbf{A}^\dagger\mathbf{A} - \mathbf{A}\ ^2$	$\ \mathbf{A}^\dagger\mathbf{A}\mathbf{A}^\dagger - \mathbf{A}^\dagger\ ^2$	$\ (\mathbf{A}\mathbf{A}^\dagger)^\top - \mathbf{A}\mathbf{A}^\dagger\ ^2$	$\ (\mathbf{A}^\dagger\mathbf{A})^\top - \mathbf{A}^\dagger\mathbf{A}\ ^2$	IN
[20]	$0.8\mathbf{A}^\top$	1.2×10^{-28}	3.7×10^{-21}	3.3×10^{-27}	6.3×10^{-28}	34
DOA	$\mathbf{x}_0 = \mathbf{0}$	9×10^{-29}	2.6×10^{-22}	3.2×10^{-23}	5.5×10^{-29}	3

5.6. Example 6

In this example, we consider a non-harmonic boundary value problem on an amoeba-like boundary:

$$\Delta u = u_{xx} + u_{yy} = 0, \quad (x, y) \in \Omega, \tag{137}$$

$$u(x, y) = f(x, y) = x^2y^3, \quad (x, y) \in \Gamma, \tag{138}$$

$$\Gamma := \{\rho(\theta) \cos \theta, \rho(\theta) \sin \theta, \quad 0 \leq \theta \leq 2\pi\},$$

$$\rho = \exp(\sin \theta) \sin^2(2\theta) + \exp(\cos \theta) \cos^2(2\theta), \tag{139}$$

where $\Delta f(x, y) = 2y^3 + 6x^2y \neq 0$, and the contour is displayed in Figure 4a by a solid black line. Feng et al. [46] have pointed out that “non-harmonic boundary data” mean that the solution does not have a harmonic extension to the whole plane and a solution is very difficult to achieve.

Let

$$v(x, y) = u(x, y) - f(x, y) \tag{140}$$

be a new variable, and then we come to a Poisson equation under a homogeneous boundary condition:

$$\text{Poisson equation : } \begin{cases} \Delta v = F(x, y) = -\Delta f(x, y), \\ v(x, y)|_{(x,y) \in \Gamma} = 0, \end{cases} \tag{141}$$

where $F(x, y) \neq 0$, because $f(x, y)$ is a non-harmonic boundary function. When $v(x, y)$ is solved, we can find $u(x, y) = v(x, y) + f(x, y)$.

In order to obtain an accurate solution of $v(x, y)$, we use the multiple-scale Pascal triangle polynomial expansion method developed by Liu and Kuo [47]:

$$v(x, y) = \sum_{i=1}^{m_0} \sum_{j=1}^i c_{ij} s_{ij} \left(\frac{x}{R_0}\right)^{i-j} \left(\frac{y}{R_0}\right)^{j-1}. \tag{142}$$

After collocating q points to satisfy the governing equation and boundary condition (141), we have a non-square linear system (2), where the scales s_{ij} are determined such that each column of the coefficient matrix \mathbf{A} has the same norm.

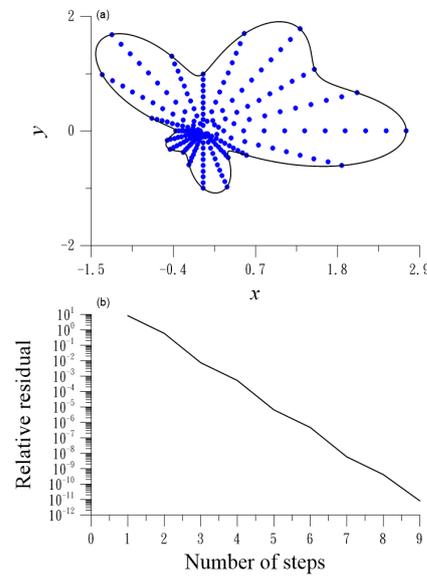


Figure 4. An amoeba-like boundary under a non-harmonic boundary function solved by the DOA: (a) the contour and distribution of collocation points, and (b) the convergence rate.

By using the DOA, we take $m = 4$, $q = 200$, $m_0 = 5$, $R_0 = 1000$ and $c_{ij}^0 = 0$. Hence, the dimension of \mathbf{A} is 200×15 , for which system (2) is an over-determined system. The distribution of collocation points is shown in Figure 4a, while the convergence rate is shown in Figure 4b. It is amazing that the DOA converges with nine steps even under a stringent convergence criterion $\varepsilon = 10^{-10}$. In Figure 5a, we compare the recovered boundary function with the exact function at 1000 points along the boundary. They are almost coincident, and thus we plot the absolute error in Figure 5b, whose ME = 2.95×10^{-12} and the root mean square error (RMSE) = 1.2×10^{-12} . This result is much better than that computed by Feng et al. [46].

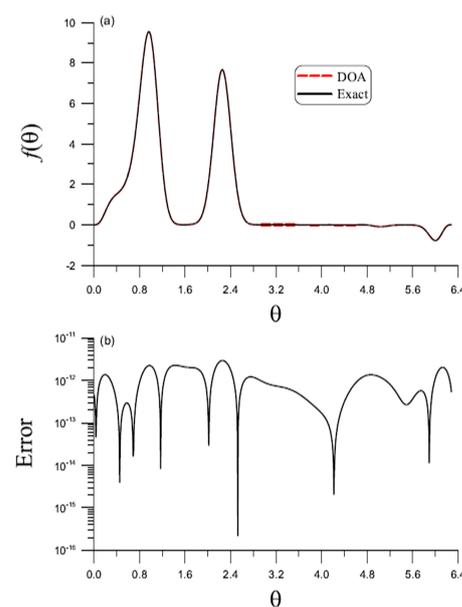


Figure 5. An amoeba-like boundary under a non-harmonic boundary function solved by the DOA: (a) comparison of numerical and exact boundary conditions, and (b) the numerical error.

5.7. Example 7

The method of fundamental solutions (MFS) is adopted to solve Equation (137) by using

$$u(\mathbf{z}) = \sum_{j=1}^n c_j \ln \|\mathbf{z} - \mathbf{s}_j\|, \quad \mathbf{z} \in \Omega, \quad \mathbf{s}_j \in \bar{\Omega}^c, \tag{143}$$

where $\bar{\Omega}^c$ is the complementary set of $\bar{\Omega}$, and

$$\mathbf{s}_j = (R \cos \theta_j, R \sin \theta_j)^T, \quad \theta_j = \frac{2j\pi}{n} \tag{144}$$

are source points, in which $R > 0$ is a parameter of the circle on which are the source points placed on.

We consider an exact solution $u(x, y) = x^2 - y^2$ and specify the Dirichlet boundary condition on the boundary given by Equation (139). In Equation (144), we fix $R = 5$. By using the DOA, we take $m = 11$ and $c_j^0 = 0$.

Table 6 lists the computed ME for different values of (n, q) obtained by the DOA. It can be seen that the DOA is applicable to least-squares problems regardless of whether $q < n$ or $q > n$.

Table 6. For the MFS on the 2D Laplace equation, ME with different (n, q) obtained by the DOA.

(n, q)	(100,80)	(100,90)	(100,110)	(100,120)	(100,150)
ME	1.88×10^{-8}	7.25×10^{-9}	7.67×10^{-8}	2.13×10^{-8}	4.45×10^{-8}

When we apply the QR to solve this problem with $n = 100$ and $q = 150$, $ME = 42.81$ is obtained. When we take $n = 150$ and $q = 400$, $ME = 2.11$ is obtained. Obviously, the QR is not applicable to this problem.

5.8. Example 8

In this example, we consider Equation (2) with a cyclic matrix, whose first row is given by $(1, \dots, n)$. The coefficient matrix can be generated by the following procedure:

$$\begin{aligned} &\text{Do } i = 1 : n, \\ &\text{Do } j = 1 : n, \\ &\text{If } i = 1, \text{ then } B_{i,j} = j; \text{ otherwise,} \\ &B_{i,j} = B_{i-1,j} + 1, \\ &\text{If } B_{i,j} > n, \text{ then } B_{i,j} = B_{i,j} - n, \\ &\text{Enddo.} \end{aligned} \tag{145}$$

The exact solution is $x_i = 1, i = 1, \dots, n$. The non-square matrix \mathbf{A} is obtained by taking the first q rows from \mathbf{B} .

Since the right-hand side vector \mathbf{b} is obtained from Equation (2) after inserting \mathbf{A} and $\mathbf{x} = \mathbf{1} = (1, \dots, 1)^T$, the non-square system is consistent. Therefore, we can compute the maximal error (ME) by

$$ME := \max_{i=1, \dots, n} |x_i^e - x_i^n|,$$

where $x_i^e = 1, i = 1, \dots, n$ are the exact solutions and x_i^n are numerical solutions.

For a large matrix of \mathbf{A} with $n = 2000$ and $q = 100$, Table 7 lists the computed ME for different values of m and the INs under $\epsilon = 10^{-5}$ with the initial guesses $x_i^0 = 1 + 0.1i, i = 1, \dots, n$. If we take the initial guesses to be $x_i^0 = 0, i = 1, \dots, n$, the accuracy is poor with $ME = 0.961$ and the convergence is very slow. As shown in Corollary 6, for the zero initial value, we have $\mathbf{r}_0 = \mathbf{b}$ and for the cyclic matrix, we have $\mathbf{r}_0 = \mathbf{b} = c\mathbf{1}$, where $c_1 = \sum_{j=1}^n j$. Thus, $\mathbf{A}^T \mathbf{A} \mathbf{r}_0 = c_0^2 c_2 \mathbf{r}_0$, where $c_2 = \sum_{j=1}^q j$. Hence, $c_0 = 2001000^2 \times 5050$ is a huge value,

which causes the failure of the DOA with a zero initial value for this problem. The same situation happens for the constant initial guesses $x_i^0 = c, i = 1, \dots, n$. It is apparent that the DOA is quite accurate and converges faster. It is interesting that a small m is enough to achieve accurate solutions; when m is smaller, the accuracy is better, but it converges slower. In contrast, when m is larger, the accuracy is worse but it converges faster.

Table 7. Example 8 with $n = 2000$ and $q = 100$: ME obtained by the DOA and INs for different m .

m	5	8	10	12	15
ME	7.24×10^{-5}	1.73×10^{-4}	2.95×10^{-4}	1.02×10^{-3}	4.99×10^{-3}
IN	174	42	23	16	10

Next, we construct a large-sized matrix \mathbf{B} from Equation (145) with n replaced by n_q . The non-square matrix \mathbf{A} is obtained by taking the first n columns from \mathbf{B} . With $n = 500$ and $q = 2000$, Table 8 lists the computed ME for different values of m , and IN under $\varepsilon = 10^{-5}$ with the initial guesses $x_i^0 = 1 + 0.1i, i = 1, \dots, n$. It is apparent that the DOA is quite accurate and converges faster. When m is increased, the accuracy is better and convergence is faster.

Table 8. Example 8 with $n = 500$ and $q = 2000$: ME obtained by the DOA and INs for different m .

m	10	12	15	18	20
ME	1.39×10^{-4}	6.82×10^{-5}	3.20×10^{-5}	1.70×10^{-5}	1.26×10^{-5}
IN	123	72	41	25	19

As mentioned, the QR is not applicable to the least-squares problem with $q < n$; however, for $q > n$, the QR is applicable. Table 9 lists the computed ME for different values of (q, n) . In the DOA, we take $m = 30, \varepsilon = 10^{-12}$, and $x_i^0 = 1 + 0.1i, i = 1, \dots, n$. The DOA converges within 25 iterations for the first four cases, and 79 iterations for the last case. The DOA can improve the accuracy by about three and four orders compared to the QR. We also list the error $\|\mathbf{Q}^T\mathbf{Q} - \mathbf{I}_n\|$ for the QR method, which is very accurate.

Table 9. Example 8 with different $(q, n), q > n$: ME obtained by QR and the DOA.

(q, n)	(1000,500)	(1500,500)	(1500,1000)	(2000,500)	(2500,1000)
$\ \mathbf{Q}^T\mathbf{Q} - \mathbf{I}_n\ $	1.81×10^{-12}	2.55×10^{-12}	8.99×10^{-12}	2.20×10^{-12}	1.14×10^{-11}
QR	3.18×10^{-10}	6.91×10^{-10}	1.73×10^{-9}	1.61×10^{-9}	4.56×10^{-9}
DOA	2.49×10^{-13}	2.66×10^{-13}	2.46×10^{-13}	1.77×10^{-13}	1.24×10^{-13}

If we take $\alpha_0 = 1$ in Equation (21), \mathcal{K}'_m reduces to the usual affine Krylov subspace $\mathbf{u}_0 + \mathcal{K}_m$. Equation (80) becomes

$$\mathbf{x} = \mathbf{u}_0 + \lambda_0 \mathbf{Vb} - \mathbf{VAu}_0.$$

As shown in Table 10, the DOA is more accurate than the DOA with $\alpha_0 = 1$.

Table 10. Example 8 with $n = 500, n_q = 2000, \varepsilon = 10^{-12}$, ME1 and IN1 obtained by taking $\alpha_0 = 1$ in the DOA and ME2 and IN2 obtained by the DOA for different m .

m	20	22	25	28	30	35
ME1	1.13×10^{-12}	8.50×10^{-13}	3.33×10^{-13}	2.0×10^{-3}	1.88×10^{-13}	2.15×10^{-13}
IN1	62	48	34	25	21	15
ME2	9.15×10^{-13}	8.59×10^{-13}	3.94×10^{-13}	2.71×10^{-13}	1.77×10^{-13}	1.41×10^{-13}
IN2	64	48	33	25	21	16

6. A Variant of the Karush–Kuhn–Tucker Equation

In many meshless collocation methods to solve linear partial differential equations, we may collocate more q points than the number of coefficients n in the expansion of the solution, which results in an over-determined system in Equation (2) with $q > n$. The residual vector $\mathbf{r} = \mathbf{b} - \mathbf{Ax}$ together with Equation (2) can be written as

$$\begin{bmatrix} \mathbf{I}_q & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}. \tag{146}$$

This is called the Karush–Kuhn–Tucker equation [15].

Instead of Equation (146), we may consider a permutation of the Karush–Kuhn–Tucker equation:

$$\begin{bmatrix} \mathbf{A} & \mathbf{I}_q \\ \mathbf{0} & \mathbf{A}^T \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{r} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}, \tag{147}$$

which is a $(q + n) \times (q + n)$ system. Equation (147) is a variant of the Karush–Kuhn–Tucker Equation (146) by re-ordering the unknown vector from (\mathbf{r}, \mathbf{x}) to (\mathbf{x}, \mathbf{r}) . When the dimension $q + n$ is a moderate value, we can apply the partial pivoting Gaussian elimination method (GEM) to solve Equations (146) and (147). We name the first partial pivoting Gaussian elimination method on Equation (146) as GEM1, and on Equation (147) as GEM2. It would be convincing if medium-size least-squares problems shown below the partial pivoting Gaussian elimination method can be an effective and accurate method. Moreover, the variant of the Karush–Kuhn–Tucker equation is more accurate than the original Karush–Kuhn–Tucker equation, even they have the same condition numbers.

6.1. Example 9

We consider Example 7 again, which is now solved by the above four methods. Table 11 lists the computed ME for different values of (n, q) obtained by GEM1 and GEM2. Here, CGNR is the conjugate gradient method applied to the normal equation with minimal residuals, while CGNE is the conjugate gradient method applied to the normal equation with minimal errors [15]. CGNR is also known as the CGLS (least-squares), and CGNE is Craig’s method. In this test, we find that when the partial pivoting Gaussian elimination method is applied to Equation (147) as GEM2, the accuracy is the best.

Table 11. The MFS on a 2D Laplace equation: ME with different (n, q) obtained by different methods.

(n, q)	(100,120)	(100,150)	(100,170)	(100,200)	(150,180)
GEM1	7.76×10^{-6}	1.81×10^{-6}	6.12×10^{-7}	2.99×10^{-6}	5.73×10^{-6}
GEM2	2.17×10^{-11}	3.82×10^{-14}	2.24×10^{-14}	1.99×10^{-14}	3.52×10^{-14}
CGNR	8.33×10^{-11}	8.23×10^{-11}	7.18×10^{-10}	6.62×10^{-9}	7.98×10^{-10}
CGNE	1.84×10^{-11}	4.78×10^{-7}	5.44×10^{-6}	6.46×10^{-10}	2.58×10^{-6}

6.2. Example 10

We consider Example 3 again, which is now solved by the above four methods, the QR method and the DOA with $\epsilon = 10^{-14}$. Based on Equations (88)–(90), this problem is solved via SVD. As shown in Table 12, the GEM2, DOA and SVD are competitive and are more accurate than other methods.

We also list the errors $\|Q^T Q - I_n\|$ for the QR method in Table 12, which is very accurate. However, since quite small singular values are present for the highly ill-conditioned Hilbert matrix, the QR method gradually deteriorates when the condition number is increased from 7.04×10^4 to 2.1×10^7 .

Table 12. Example 10: ME with different (n, q) obtained by GEM1, GEM2, CGNR, CGNE, the DOA and QR.

(n, q)	(5,6)	(5,8)	(5,10)	(7,25)	(8,25)
GEM1	1.80×10^{-6}	5.18×10^{-7}	4.57×10^{-7}	2.20×10^{-3}	1.55×10^{-3}
GEM2	3.15×10^{-12}	2.27×10^{-13}	1.02×10^{-12}	1.29×10^{-10}	4.70×10^{-9}
CGNR	4.98×10^{-8}	1.62×10^{-7}	3.05×10^{-7}	4.83×10^{-8}	7.64×10^{-8}
CGNE	3.52×10^{-12}	1.64×10^{-12}	1.88×10^{-12}	4.83×10^{-8}	7.64×10^{-8}
DOA	1.88×10^{-13}	2.55×10^{-12}	7.81×10^{-13}	2.72×10^{-9}	1.48×10^{-9}
SVD	4.85×10^{-12}	1.52×10^{-12}	9.09×10^{-13}	3.08×10^{-10}	1.52×10^{-8}
$\ Q^T Q - I_n\ $	8.65×10^{-12}	4.57×10^{-12}	5.24×10^{-12}	5.66×10^{-10}	1.89×10^{-8}
QR	1.05×10^{-6}	3.50×10^{-7}	2.27×10^{-7}	2.46×10^{-3}	5.68×10^{-1}

7. Conclusions

Least-squares problems arise in a lot of applications, and many iterative algorithms are already available to seek their solutions. In this paper, a new concept of a varying affine Krylov subspace was introduced, which is different to the fixed-type affine Krylov subspace used in the CG-type numerical solution of over-determined least-squares problems. In the $m + 1$ -dimensional varying affine Krylov subspace, a closed-form double optimal solution in a simple form in Equation (42) was derived and further reduced to Equation (80), which was obtained by two minimizations in Equations (40) and (41). We analyzed a key equation (72) to link these two optimizations together. The double optimal solution is an excellent approximation, as verified for least-squares problems. The iterative DOA was developed, which leads to step-by-step absolute convergence. In each iterative step, by merely inverting an $m \times m$ matrix with a small value of m , the computational cost of the DOA is very low, lower than other methods. The Moore–Penrose inverses of non-square matrices were also derived by using the varying affine Krylov subspace method. For over-determined least-squares problems, we proposed a variant of the Karush–Kuhn–Tucker equation, which uses the partial pivoting Gaussian elimination method and is better than the original Karush–Kuhn–Tucker equation and CG-type numerical methods of CGNR (CGLS) and CGNE (Craig’s method). It is very important that the DOA can be applied to both rectangular systems with $q > n$ or $q < n$.

The novelties involved in this paper are as follows:

- A double-optimization-based iterative algorithm for least-squares problems was developed in a varying affine Krylov subspace.
- For dense large-size least-squares problems, the proposed iterative DOA is efficient and accurate.
- The DOA can be applied to both rectangular systems with extremal cases of $q \ll n$ or $q \gg n$.
- A variant of the Karush–Kuhn–Tucker equation was presented, which is an improved version of the original Karush–Kuhn–Tucker equation.

The idea of varying the affine Krylov subspace is useful to find a better solution of linear algebraic equations based on the mechanism of double optimization. In the future, we may extend the new methods of DOSs and the DOA to other linear matrix equations. The limitations are that there are several matrix multiplications needed to construct the projection operator \mathbf{P} in the Krylov subspace and the high computational cost for inversion of the matrix \mathbf{C}^{-1} with dimension m .

Author Contributions: Conceptualization, C.-S.L. and C.-W.C.; Methodology, C.-S.L. and C.-W.C.; Software, C.-S.L., C.-W.C. and C.-L.K.; Validation, C.-S.L., C.-W.C. and C.-L.K.; Formal analysis, C.-S.L. and C.-W.C.; Investigation, C.-S.L., C.-W.C. and C.-L.K.; Resources, C.-S.L. and C.-W.C.; Data curation, C.-S.L., C.-W.C. and C.-L.K.; Writing—original draft, C.-S.L.; Writing—review & editing, C.-W.C.; Visualization, C.-S.L., C.-W.C. and C.-L.K.; Supervision, C.-S.L. and C.-W.C.; Project administration, C.-W.C.; Funding acquisition, C.-W.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding authors.

Acknowledgments: The authors would like to express their thanks to the reviewers, who supplied valuable opinions to improve the quality of this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Liu, C.S. An optimal multi-vector iterative algorithm in a Krylov subspace for solving the ill-posed linear inverse problems. *Comput. Mater. Contin.* **2013**, *33*, 175–198.
- Dongarra, J.; Sullivan, F. Guest editors' introduction to the top 10 algorithms. *Comput. Sci. Eng.* **2000**, *2*, 22–23. [[CrossRef](#)]
- Simoncini, V.; Szyld, D.B. Recent computational developments in Krylov subspace methods for linear systems. *Numer. Linear Alg. Appl.* **2007**, *14*, 1–59. [[CrossRef](#)]
- Saad, Y.; Schultz, M.H. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.* **1986**, *7*, 856–869. [[CrossRef](#)]
- Saad, Y. Krylov subspace methods for solving large unsymmetric linear systems. *Math. Comput.* **1981**, *37*, 105–126. [[CrossRef](#)]
- Freund, R.W.; Nachtigal, N.M. QMR: A quasi-minimal residual method for non-Hermitian linear systems. *Numer. Math.* **1991**, *60*, 315–339. [[CrossRef](#)]
- van Den Eshof, J.; Sleijpen, G.L.G. Inexact Krylov subspace methods for linear systems. *SIAM J. Matrix Ana. Appl.* **2004**, *26*, 125–153. [[CrossRef](#)]
- Hestenes, M.R.; Stiefel, E.L. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.* **1952**, *49*, 409–436. [[CrossRef](#)]
- Lanczos, C. Solution of systems of linear equations by minimized iterations. *J. Res. Nat. Bur. Stand.* **1952**, *49*, 33–53. [[CrossRef](#)]
- Paige, C.C.; Saunders, M.A. Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.* **1975**, *12*, 617–629. [[CrossRef](#)]
- Fletcher, R. *Conjugate Gradient Methods for Indefinite Systems*; Lecture Notes in Math; Springer: Berlin/Heidelberg, Germany, 1976; Volume 506, pp. 73–89.
- Sonneveld, P. CGS: A fast Lanczos-type solver for nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.* **1989**, *10*, 36–52. [[CrossRef](#)]
- van der Vorst, H.A. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.* **1992**, *13*, 631–644. [[CrossRef](#)]
- Saad, Y.; van der Vorst, H.A. Iterative solution of linear systems in the 20th century. *J. Comput. Appl. Math.* **2000**, *123*, 1–33. [[CrossRef](#)]
- Saad, Y. *Iterative Methods for Sparse Linear Systems*, 2nd ed.; SIAM: Philadelphia, PA, USA, 2003.
- van der Vorst, H.A. *Iterative Krylov Methods for Large Linear Systems*; Cambridge University Press: New York, NY, USA, 2003.
- Golub, G. Numerical methods for solving linear least squares problems. *Numer. Math.* **1965**, *7*, 206–216. [[CrossRef](#)]
- Choi, S.C.; Paige, C.C.; Saunders, M.A. MINRES-QLP: A Krylov subspace method for indefinite or singular symmetric systems. *SIAM J. Sci. Comput.* **2011**, *33*, 1810–1836. [[CrossRef](#)]
- Petkovic, M.D.; Stanimirovic, P.S. Iterative method for computing Moore-Penrose inverse based on Penrose equations. *J. Comput. Appl. Math.* **2011**, *235*, 1604–1613. [[CrossRef](#)]
- Petkovic, M.D.; Stanimirovic, P.S. Two improvements of the iterative method for computing Moore-Penrose inverse based on Penrose equations. *J. Comput. Appl. Math.* **2014**, *267*, 61–71. [[CrossRef](#)]

21. Katsikis, V.N.; Pappas, D.; Petralias, A. An improved method for the computation of the Moore-Penrose inverse matrix. *Appl. Math. Comput.* **2011**, *217*, 9828–9834. [[CrossRef](#)]
22. Stanimirovic, I.; Tasic, M. Computation of generalized inverse by using the LDL* decomposition. *Appl. Math. Lett.* **2012**, *25*, 526–531. [[CrossRef](#)]
23. Sheng, X.; Wang, T. An iterative method to compute Moore-Penrose inverse based on gradient maximal convergence rate. *Filomat* **2013**, *27*, 1269–1276. [[CrossRef](#)]
24. Toutounian, F.; Ataei, A. A new method for computing Moore-Penrose inverse matrices. *J. Comput. Appl. Math.* **2009**, *228*, 412–417. [[CrossRef](#)]
25. Zhang, J.; Guo, J. On relaxed greedy randomized coordinate descent methods for solving large linear least-squares problems. *Appl. Numer. Math.* **2020**, *157*, 372–384. [[CrossRef](#)]
26. Lipitakis, A.-D.; Filelis-Papadopoulos, C.K.; Gravvanis, G.A.; Anagnostopoulos, D. A note on parallel approximate pseudoinverse matrix techniques for solving linear least squares problems. *J. Comput. Sci.* **2020**, *41*, 101092. [[CrossRef](#)]
27. Niu, Y.-Q.; Zhang, B. A new randomized Gauss–Seidel method for solving linear least-squares problems. *Appl. Math. Lett.* **2021**, *116*, 107057. [[CrossRef](#)]
28. Bojanczyk, A.W. Algorithms for indefinite linear least squares problems. *Linear Algebra Appl.* **2021**, *623*, 104–127. [[CrossRef](#)]
29. Du, K.; Ruan, C.-C.; Sun, X.-H. On the convergence of a randomized block coordinate descent algorithm for a matrix least squares problem. *Appl. Math. Lett.* **2022**, *124*, 107689. [[CrossRef](#)]
30. Chakrabarti, K.; Gupta, N.; Chopra, N. Iterative pre-conditioning for expediting the distributed gradient-descent method: The case of linear least-squares problem. *Automatica* **2022**, *137*, 110095. [[CrossRef](#)]
31. Zhang, Y.; Li, H. Splitting-based randomized iterative methods for solving indefinite least squares problem. *Appl. Math. Comput.* **2023**, *446*, 127892. [[CrossRef](#)]
32. Pes, F.; Rodriguez, G. A projection method for general form linear least-squares problems. *Appl. Math. Lett.* **2023**, *145*, 108780. [[CrossRef](#)]
33. Kuo, Y.-C.; Liu, C.-S. An index search method based inner-outer iterative algorithm for solving nonnegative least squares problems. *J. Comput. Appl. Math.* **2023**, *424*, 114954. [[CrossRef](#)]
34. Jin, L.-L.; Li, H.B. Greedy double subspaces coordinate descent method for solving linear least-squares problems. *J. Comput. Sci.* **2023**, *70*, 102029. [[CrossRef](#)]
35. Jahvani, M.; Guay, M. Solving least-squares problems in directed networks: A distributed approach. *Comput. Chem. Eng.* **2024**, *185*, 108654. [[CrossRef](#)]
36. Laura Arias, M.; Contino, M.; Maestripieri, A.; Marcantognini, S. Matrix representations of multivalued projections and least squares problems. *J. Math. Anal. Appl.* **2024**, *530*, 127631. [[CrossRef](#)]
37. Golub, G.H.; Reinsch, C. Singular value decomposition and least squares solutions. *Numer. Math.* **1970**, *14*, 403–420. [[CrossRef](#)]
38. Björck, A. *Numerical Methods for Least Squares Problems*; SIAM Publisher: Philadelphia, PA, USA, 1996.
39. Liu, C.S. A doubly optimized solution of linear equations system expressed in an affine Krylov subspace. *J. Comput. Appl. Math.* **2014**, *260*, 375–394. [[CrossRef](#)]
40. Liu, C.S. Optimal algorithms in a Krylov subspace for solving linear inverse problems by MFS. *Eng. Anal. Bound. Elem.* **2014**, *44*, 64–75. [[CrossRef](#)]
41. Liu, C.S. A double optimal descent algorithm for iteratively solving ill-posed linear inverse problems. *Inv. Prob. Sci. Eng.* **2015**, *23*, 38–66. [[CrossRef](#)]
42. Liu, C.S. A maximal projection solution of ill-posed linear system in a column subspace, better than the least squares solution. *Comput. Math. Appl.* **2014**, *67*, 1998–2014. [[CrossRef](#)]
43. Papez, J.; Tichy, P. Estimating error norms in CG-like algorithms for least-squares and least-norm problems. *Numer. Algorithms* **2024**, *1*–28. [[CrossRef](#)]
44. Abbasi, M.; Nikazad, T. Bounded perturbations resilient iterative methods for linear systems and least squares problems: Operator-based approaches, analysis, and performance evaluation. *BIT Numer. Math.* **2024**, *64*, 15. [[CrossRef](#)]
45. Xia, Y.; Chen, T.; Shan, J. A novel iterative method for computing generalized inverse. *Neural Comput.* **2014**, *26*, 449–465. [[CrossRef](#)] [[PubMed](#)]
46. Feng, G.; Li, M.; Chen, C.S. On the ill-conditioning of the MFS for irregular boundary data with sufficient regularity. *Eng. Anal. Bound. Elem.* **2014**, *41*, 98–102. [[CrossRef](#)]
47. Liu, C.S.; Kuo, C.L. A multiple-scale Pascal polynomial triangle solving elliptic equations and inverse Cauchy problems. *Eng. Anal. Bound. Elem.* **2016**, *62*, 35–43. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.