52nd IEEE Conference on Decision and Control
December 10-13, 2013. Florence, Italy

A Complex Singular Value Decomposition Algorithm Based on the Riemannian Newton Method

Hiroyuki Sato and Toshihiro Iwai

***Abstract*— In this paper, the problem of finding the singular value decomposition (SVD) of a complex matrix is formulated as an optimization problem on the product of two complex Stiefel manifolds. A new algorithm for the complex SVD is proposed on the basis of the Riemannian Newton method. This algorithm can provide the singular vectors associated with an arbitrary number of the singular values from the largest one down to a smaller one. Furthermore, once a sufficiently accurate approximate complex SVD is given, the Riemannian Newton method can improve it to be as accurate as the computer accuracy permits.**

I. INTRODUCTION

The singular value decomposition (SVD) is a very important matrix factorization in frequent use in various fields such as signal and image processing, control theory, and statistics [5], [8], [9]. On the other hand, optimization on Riemannian manifolds, which is called Riemannian optimization, has been developed and applied to several numerical computations [1], [2], [4]. In [7], an algorithm for the SVD of a real matrix has been proposed on the basis of Riemannian optimization methods on the product of two real Stiefel manifolds. In comparison with this, the SVD of a complex matrix, which is also used frequently [3], [6], has not been formulated as a Riemannian optimization problem.

In this paper, a complex SVD algorithm based on the Riemannian Newton method is proposed as a generalization of the real SVD algorithm discussed in [7]. As is expected, the complex SVD problem is described as a Riemannian optimization problem on the product of two complex Stiefel manifolds. For feasibility purpose, the problem is equivalently rewritten as a problem on the product of two real manifolds, each of which is an intersection of the real Stiefel manifold and the "quasi-symplectic set" to be defined in Section II. The Riemannian geometry of the real product manifold in question is investigated after [7] in Section III. In particular, the gradient and the Hessian of the objective function are given together with a retraction map. In this setting, Newton's method on the real product manifold is developed in Section IV, which is in turn converted to that on the complex product manifold, and followed by a new complex SVD algorithm. A numerical experiment is also performed to show that the present algorithm may improve the accuracy of a complex SVD obtained by an existing method. Moreover, like the algorithm given in [7], the proposed algorithm divides the problem into easier subproblems, which can be solved in parallel. This paper concludes with some remarks in Section V.

Hiroyuki Sato and Toshihiro Iwai are with Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Japan.

II. COMPLEX SVD AND THE CORRESPONDING RIEMANNIAN OPTIMIZATION PROBLEM

A. Setting Up

Let m and n be positive integers with $m \geq n$. The SVD of an $m\times n$ real matrix A is wholly or partly realized by solving an optimization (minimization) problem on $\mathrm{St}(p,m,\mathbb{R}) \times \mathrm{St}(p,n,\mathbb{R})$ [7], where p is an arbitrary positive integer not greater than n, $\mathrm{St}(p,n,\mathbb{R})$ is the real Stiefel manifold defined by $\mathrm{St}(p,n,\mathbb{R}) = \left\{Y \in \mathbb{R}^{n\times p} \,|\, Y^TY = I_p\right\}$, and where the superscript T denotes the transposition of a matrix. The objective function $F_{\mathbb{R}}$ of the problem is

$$\begin{aligned} F_{\mathbb{R}}(U,V) &= -\operatorname{tr}(U^TAVN), \\ (U,V) &\in \mathrm{St}(p,m,\mathbb{R}) \times \mathrm{St}(p,n,\mathbb{R}), \end{aligned} \tag{1}$$

where $N = \operatorname{diag}(\mu_1,\dots,\mu_p),\ \mu_1 > \cdots > \mu_p > 0$. The solution to the present optimization problem is a pair of matrices whose columns are left and right singular vectors associated with the p largest singular values of A, respectively.

We turn to the SVD of an $m\times n$ complex matrix A, which is expressed as

$$A = U\Sigma V^H, \quad U \in U(m),\ V \in U(n),\ \Sigma = \begin{pmatrix}\Sigma_1\\ 0\end{pmatrix},$$

where $\Sigma_1 = \operatorname{diag}(\sigma_1,\dots,\sigma_n),\ \sigma_1 \geq \cdots \geq \sigma_n \geq 0$, and where the superscript H denotes the Hermitian conjugation of a matrix. The non-negative real numbers $\sigma_1,\dots,\sigma_n$ are called the singular values of A and the j-th columns of U and V are the left and right singular vectors associated with σ_j, respectively. In a similar manner to the real case discussed in [1], [7], we consider a truncated complex SVD of A. The problem is to find the left and right singular vectors associated with p largest singular values of A, where p is an arbitrarily fixed integer not greater than n.

The truncated complex SVD is naturally formulated to be an optimization problem on $\mathrm{St}(p,m,\mathbb{C}) \times \mathrm{St}(p,n,\mathbb{C})$, where $\mathrm{St}(p,n,\mathbb{C})$ is the complex Stiefel manifold defined by $\mathrm{St}(p,n,\mathbb{C}) = \left\{Y \in \mathbb{C}^{n\times p} \,|\, Y^HY = I_p\right\}$. With the same matrix N as described above, the objective function (1) would be replaced by $-\operatorname{tr}(U^HAVN)$ with $(U,V) \in \mathrm{St}(p,m,\mathbb{C}) \times \mathrm{St}(p,n,\mathbb{C})$. However, this function is no longer real-valued in general, and not appropriate as an objective function. An alternative objective function is $f(U,V) = -|\operatorname{tr}(U^HAVN)|$. However, as will be shown in Thm. 1, another real-valued function $F(U,V) = -\operatorname{Re}(\operatorname{tr}(U^H AVN))$ is an appropriate objective function, where $\operatorname{Re}(\cdot)$ denotes the real part of the quantity in the parentheses. Further, the function F is better than f from both theoretical and numerical viewpoints. Indeed, if we try to use f as an objective function, we end up with choosing the square of f in computing the gradient and the Hessian with respect to U and V. In contrast with this, if F is chosen as an objective function, its gradient and Hessian can be calculated without squaring F, and further F consists only of the real part of $\operatorname{tr}(U^H AVN)$ while f includes both the real and imaginary parts. For this reason, F is a better choice for computing requisites. See also the remark to be made below Problem 3.

Now, we deal with the following optimization problem on $\operatorname{St}(p,m,\mathbb{C}) \times \operatorname{St}(p,n,\mathbb{C})$.

Problem 1:

$$\begin{aligned} &\text{minimize} \quad F(U,V) = -\operatorname{Re}(\operatorname{tr}(U^H AVN)), \\ &\text{subject to} \quad (U,V) \in \operatorname{St}(p,m,\mathbb{C}) \times \operatorname{St}(p,n,\mathbb{C}), \end{aligned}$$

where $N = \operatorname{diag}(\mu_1,\ldots,\mu_p)$, $\mu_1 > \cdots > \mu_p > 0$.

Although the objective function F consists of the real part of $\operatorname{tr}(U^H AVN)$, it works well for finding the truncated SVD of A, as is shown in the following theorem.

Theorem 1: Let (U_*, V_*) be an optimal solution to Problem 1. Then, the j-th columns of U_* and V_* are the left and right singular vectors of A associated with the j-th dominant singular value, respectively. In addition, the p largest singular values $\sigma_1 \geq \cdots \geq \sigma_p$ can be calculated through the formula $U_*^H AV = \operatorname{diag}(\sigma_1,\ldots,\sigma_p)$, *i.e.*, (15) with $s_j = \sigma_j$.

To prove the theorem, we put Problem 1 into the form of a real problem. We denote $A = B + iC$, $U = X + iY$, and $V = Z + iW$, where $B,\ C \in \mathbb{R}^{m\times n}$ are the real and imaginary parts of $A \in \mathbb{C}^{m\times n}$, respectively, and where $X,\ Y \in \mathbb{R}^{m\times p}$ and $Z,\ W \in \mathbb{R}^{n\times p}$ are those of $U \in \mathbb{C}^{m\times p}$ and $V \in \mathbb{C}^{n\times p}$, respectively. The conditions $U^H U = V^H V = I_p$ for $(U,V) \in \operatorname{St}(p,m,\mathbb{C}) \times \operatorname{St}(p,n,\mathbb{C})$ are written out in terms of X,Y,Z,W, and the objective function $-\operatorname{Re}(\operatorname{tr}(U^H AVN))$ are expressed in terms of B,C,X,Y,Z,W as well. Hence, Problem 1 can be put equivalently in the real form as follows:

Problem 2:

$$\begin{aligned} &\text{maximize} \quad G(X,Y,Z,W) \\ &\quad = \operatorname{tr}((X^T BZ - X^T CW + Y^T BW + Y^T CZ)N), \\ &\text{subject to} \quad X^T X + Y^T Y = Z^T Z + W^T W = I_p, \\ &\qquad\qquad X^T Y - Y^T X = Z^T W - W^T Z = 0. \end{aligned}$$

We here introduce the Lagrangian of Problem 2 by

$$\begin{aligned} &L(X,Y,Z,W,\Lambda,\Omega,\Gamma,\Delta) \\ =&G(X,Y,Z,W) + \operatorname{tr}(\Lambda(X^T X + Y^T Y - I_p)) \\ &+ \operatorname{tr}(\Omega(Z^T Z + W^T W - I_p)) \\ &+ \operatorname{tr}(\Gamma(X^T Y - Y^T X)) + \operatorname{tr}(\Delta(Z^T W - W^T Z)), \end{aligned}$$

where $\Lambda, \Omega \in \operatorname{Sym}(p)$ and $\Gamma, \Delta \in \operatorname{Skew}(p)$ are the Lagrange multiplier matrices, and where $\operatorname{Sym}(p)$ and $\operatorname{Skew}(p)$ are the sets of all $p\times p$ real symmetric and skew-symmetric matrices, respectively. Note here that $X^T X + Y^T Y - I_p, Z^T Z + W^T W - I_p \in \operatorname{Sym}(p)$ and $X^T Y - Y^T X, Z^T W - W^T Z \in \operatorname{Skew}(p)$.

Let L_X denote the partial derivative of L with respect to X, and so on. Since $L_X = L_Y = 0$, $L_Z = L_W = 0$, $L_\Lambda = L_\Omega = L_\Gamma = L_\Delta = 0$ at an optimal solution $(X_*, Y_*, Z_*, W_*, \Lambda_*, \Omega_*, \Gamma_*, \Delta_*)$, we have

$$(BZ_* - CW_*)N + 2X_*\Lambda_* + 2Y_*\Gamma_* = 0, \tag{2}$$
$$(BW_* + CZ_*)N + 2Y_*\Lambda_* - 2X_*\Gamma_* = 0, \tag{3}$$
$$(B^T X_* + C^T Y_*)N + 2Z_*\Omega_* + 2W_*\Delta_* = 0, \tag{4}$$
$$(B^T Y_* - C^T X_*)N + 2W_*\Omega_* - 2Z_*\Omega_* = 0, \tag{5}$$
$$X_*^T X_* + Y_*^T Y_* = Z_*^T Z_* + W_*^T W_* = I_p, \tag{6}$$
$$X_*^T Y_* - Y_*^T X_* = Z_*^T W_* - W_*^T Z_* = 0. \tag{7}$$

We return to the proof of the theorem in the complex form. Let $U_* = X_* + iY_*$ and $V_* = Z_* + iW_*$. Note that rewriting (6) and (7) into the complex forms results in $U_*^H U_* = V_*^H V_* = I_p$. Adding (2) to (3) multiplied by i, we obtain

$$AV_*N + 2U_*(\Lambda_* - i\Gamma_*) = 0. \tag{8}$$

Since $U_*^H U_* = I_p$, it follows from (8) that

$$\Lambda_* - i\Gamma_* = -\frac{1}{2}U_*^H AV_*N. \tag{9}$$

Substituting (9) into (8) yields

$$AV_* = U_*U_*^H AV_*. \tag{10}$$

Also, since $\Lambda_* \in \operatorname{Sym}(p)$ and $\Gamma_* \in \operatorname{Skew}(p)$, we obtain

$$(\Lambda_* - i\Gamma_*)^H = \Lambda_*^T + i\Gamma_*^T = \Lambda_* - i\Gamma_*,$$

which implies that $\Lambda_* - i\Gamma_*$ is a Hermitian matrix. Therefore, the right-hand side of (9) is also Hermitian, so that we have

$$U_*^H AV_*N = NV_*^H A^H U_*. \tag{11}$$

In a similar manner, (4) and (5) are put together to eventually give rise to

$$A^H U_* = V_*V_*^H A^H U_*, \tag{12}$$

$$V_*^H A^H U_*N = NU_*^H AV_*. \tag{13}$$

From (11) and (13), it follows that

$$U_*^H AV_*N^2 = N^2 U_*^H AV_*. \tag{14}$$

Since N^2 is a diagonal matrix, (14) implies that $U_*^H AV_*$ is a diagonal matrix as well, which we express as

$$U_*^H AV_* = \operatorname{diag}(s_1,\ldots,s_p). \tag{15}$$

From (15) and its Hermitian conjugate, (11) is found to take the form

$$\operatorname{diag}(s_1\mu_1,\ldots,s_p\mu_p) = \operatorname{diag}(\bar{s}_1\mu_1,\ldots\bar{s}_p\mu_p),$$

which implies that s_j's are real numbers. In addition, (10) and (12) are put together to imply that

$$\begin{aligned} A^H AV_* &= (A^H U_*)(U_*^H AV_*) = V_*(V_*^H A^H U_*)(U_*^H AV_*) \\ &= V_* \operatorname{diag}(s_1,\ldots,s_p)^2 = V_* \operatorname{diag}(s_1^2,\ldots,s_p^2), \end{aligned}$$

which means that s_j^2 are eigenvalues of $A^H A$ and that the j-th column of V_* is the corresponding eigenvector. Then, the objective function G regarded as the function of $(U,V) \in$

$\mathrm{St}(p,m,\mathbb{C}) \times \mathrm{St}(p,n,\mathbb{C})$ is evaluated at an optimal solution (U_*, V_*) as

$$G(U_*, V_*) = \mathrm{tr}(U_*^H A V_* N) = \sum_{j=1}^{p} s_j \mu_j.$$

Since $\mu_1 > \cdots > \mu_p > 0$ and since $(U,V) = (U_*, V_*)$ are supposed to maximize $G(U,V)$, s_j's should be the p largest singular values among all the singular values of A and be ordered as $s_1 \geq \cdots \geq s_p \geq 0$. Therefore, s_j is the j-th largest singular value of A and the j-th column of V_* is the corresponding right singular vector. Similarly, the j-th column of U_* is the left singular vector associated with s_j. This completes the proof.

B. *Realization of* $\mathrm{St}(p,n,\mathbb{C})$ *as the Intersection of the Real Stiefel Manifold and the Quasi-Symplectic Set*

An $n \times n$ complex matrix $D = E + iF \in \mathbb{C}^{n\times n}$, $E, F \in \mathbb{R}^{n\times n}$, can be expressed as a $2n \times 2n$ real matrix

$$\tilde{D} = \begin{pmatrix} E & F \\ -F & E \end{pmatrix}, \tag{16}$$

and vice versa. A $2n \times 2n$ matrix $\hat{D}$ has the form (16) if and only if

$$J_n \hat{D} = \hat{D} J_n, \quad J_n := \begin{pmatrix} 0 & I_n \\ -I_n & 0 \end{pmatrix}.$$

Further, if $D = E + iF$ is unitary, then the corresponding real matrix $\tilde{D}$ given in (16) becomes orthogonal, and the condition $J_n \tilde{D} = \tilde{D} J_n$ is equivalently written as $\tilde{D}^T J_n \tilde{D} = J_n$, which implies that $\tilde{D}$ is a symplectic matrix. Let $\mathrm{Sp}(n,\mathbb{R})$ denote the real symplectic group defined by

$$\mathrm{Sp}(n,\mathbb{R}) = \left\{ \tilde{D} \in \mathbb{R}^{2n\times 2n} \mid \tilde{D}^T J_n \tilde{D} = J_n \right\}.$$

Then, the map

$$\psi^{(n)} : U(n) \to O(2n) \cap \mathrm{Sp}(n,\mathbb{R});$$
$$\psi^{(n)}(E + iF) = \begin{pmatrix} E & F \\ -F & E \end{pmatrix}, \tag{17}$$

gives an isomorphism between $U(n)$ and $O(2n) \cap \mathrm{Sp}(n,\mathbb{R})$.

We generalize the mapping $\psi^{(n)}$ into the rectangular matrix case. We first define $\mathcal{SP}(p,q)$ for integers p, q as

$$\mathcal{SP}(p,q) = \left\{ \tilde{D} \in \mathbb{R}^{2q\times 2p} \mid \tilde{D} J_p = J_q \tilde{D} \right\},$$

which we call the quasi-symplectic set. Note that if $p = q = n$, then $U(n) \simeq O(2n) \cap \mathrm{Sp}(n,\mathbb{R}) = O(2n) \cap \mathcal{SP}(n,n)$, though $\mathcal{SP}(n,n)$ itself is not identical to $\mathrm{Sp}(n,\mathbb{R})$. The set $\mathbb{C}^{n\times p}$ of all $n \times p$ complex matrices is isomorphic to $\mathcal{SP}(p,n)$ with the isomorphism

$$\phi^{(p,n)} : \mathbb{C}^{n\times p} \to \mathcal{SP}(p,n);\ \phi^{(p,n)}(E + iF) = \begin{pmatrix} E & F \\ -F & E \end{pmatrix}, \tag{18}$$

where $E, F \in \mathbb{R}^{n\times p}$. Then, the map $\phi^{(p,n)}|_{\mathrm{St}(p,n,\mathbb{C})}$, which is the restriction of $\phi^{(p,n)}$ to the complex Stiefel manifold $\mathrm{St}(p,n,\mathbb{C})$, gives a real expression of $\mathrm{St}(p,n,\mathbb{C})$, which we denote by

$$\mathrm{Stp}(p,n) := \mathrm{St}(2p, 2n, \mathbb{R}) \cap \mathcal{SP}(p,n). \tag{19}$$

We introduce the set $\mathcal{SP}(n)$ as the collection of $\mathcal{SP}(p,q)$ over all positive integers $p, q \leq n$:

$$\mathcal{SP}(n) = \cup_{\substack{0<p\leq n \\ 0<q\leq n}} \mathcal{SP}(p,q).$$

Also, we define the map ϕ as the collection of $\phi^{(p,q)}$:

$$\phi(B) = \phi^{(p,q)}(B), \qquad B \in \mathbb{C}^{q\times p}.$$

In what follows, for a square or rectangular complex matrix B, we denote the matrix $\phi(B) \in \mathcal{SP}(n)$ by $\tilde{B} = \phi(B)$. Then, matrix operations on matrices without and with tilde are related as follows:

$$B + C \longleftrightarrow \tilde{B} + \tilde{C}, \quad BD \longleftrightarrow \tilde{B}\tilde{D}, \quad B^H \longleftrightarrow \tilde{B}^T, \tag{20}$$

and the traces of matrices with and without tilde are related by

$$2\,\mathrm{Re}(\mathrm{tr}(E)) = \mathrm{tr}(\tilde{E}), \tag{21}$$

where B, C, D are complex matrices of appropriate size for addition and multiplication, and where E is a square complex matrix. Note that the set $\mathcal{SP}(n)$ is closed under the operations in the right-hand sides of (20).

We are now in a position to deal with the complex Stiefel manifold $\mathrm{St}(p,n,\mathbb{C})$ in the real form $\mathrm{Stp}(p,n)$ given in (19). On account of (21), the objective function $F(U,V) = -\mathrm{Re}(\mathrm{tr}(U^H AVN))$ in Problem 1 is now rewritten as

$$-\mathrm{Re}(\mathrm{tr}(U^H AVN)) = -\frac{1}{2}\mathrm{tr}(\tilde{U}^T \tilde{A} \tilde{V} \tilde{N}). \tag{22}$$

We remain to use the same symbol F to denote the function of $(\tilde{U}, \tilde{V}) \in \mathrm{Stp}(p,m) \times \mathrm{Stp}(p,n)$ in the right-hand side of (22). Thus, we are led to the following optimization problem on $\mathrm{Stp}(p,m) \times \mathrm{Stp}(p,n)$.

Problem 3:

$$\begin{aligned} &\text{minimize} \quad F(\tilde{U}, \tilde{V}) = -\frac{1}{2}\mathrm{tr}(\tilde{U}^T \tilde{A} \tilde{V} \tilde{N}), \\ &\text{subject to} \quad (\tilde{U}, \tilde{V}) \in \mathrm{Stp}(p,m) \times \mathrm{Stp}(p,n), \end{aligned} \tag{23}$$

where $\tilde{N} = \begin{pmatrix} N & 0 \\ 0 & N \end{pmatrix}$.

We note that the fact that Problem 1 is naturally put into Problem 3 as a real expression shows another merit in choosing $F(U,V) = -\mathrm{Re}(\mathrm{tr}(U^H AVN))$ as an objective function rather than $f(U,V) = -|\mathrm{tr}(U^H AVN)|$.

III. RIEMANNIAN GEOMETRY OF $\mathrm{Stp}(p,m) \times \mathrm{Stp}(p,n)$

In order to apply Newton's method to Problem 3, we need the gradient and the Hessian of the objective function F together with a retraction [1] on the product manifold $\mathrm{Stp}(p,m) \times \mathrm{Stp}(p,n)$. In this section, we deal with the Riemannian geometry of $\mathrm{Stp}(p,m) \times \mathrm{Stp}(p,n)$ by employing the results in [7].

A. Tangent Spaces and the Orthogonal Projection

The tangent space to $\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$ at $(\tilde{U},\tilde{V})$ is given by

$$T_{(\tilde{U},\tilde{V})}\left(\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)\right) = \Big\{(\tilde{\xi},\tilde{\eta})\in\mathcal{SP}(p,m)\times\mathcal{SP}(p,n)\,| \\ \tilde{\xi}^T\tilde{U}+\tilde{U}^T\tilde{\xi}=\tilde{\eta}^T\tilde{V}+\tilde{V}^T\tilde{\eta}=0\Big\}.$$

We proceed to endow $\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$ with a Riemannian metric. The Euclidean space $\mathbb{R}^{2n\times 2p}$ is endowed with the standard inner product

$$(M_1,M_2)=\mathrm{tr}(M_1^TM_2),\qquad M_1,M_2\in\mathbb{R}^{2n\times 2p}.$$

When restricted to the subspace $\mathcal{SP}(p,n)$ of $\mathbb{R}^{2n\times 2p}$, the inner product takes the form

$$(\tilde{B},\tilde{C})=\mathrm{tr}(\tilde{B}^T\tilde{C})=2\,\mathrm{tr}(B_1^TC_1+B_2^TC_2), \tag{24}$$

for $\tilde{B}=\begin{pmatrix}B_1 & B_2\\ -B_2 & B_1\end{pmatrix}, \tilde{C}=\begin{pmatrix}C_1 & C_2\\ -C_2 & C_1\end{pmatrix}\in\mathcal{SP}(p,n)$. Getting rid of the factor 2 in the right-hand side of (24), we define the inner product on $\mathcal{SP}(p,n)$ to be

$$\langle\tilde{B},\tilde{C}\rangle=\frac{1}{2}\,\mathrm{tr}(\tilde{B}^T\tilde{C}),\qquad \tilde{B},\tilde{C}\in\mathcal{SP}(p,n).$$

Then, the manifold $\mathrm{Stp}(p,n)$ as a submanifold of $\mathcal{SP}(p,n)$ is endowed with the induced metric. Further, the product manifold $\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$ is endowed with the product metric, which is expressed as

$$\langle(\tilde{\xi}_1,\tilde{\eta}_1),(\tilde{\xi}_2,\tilde{\eta}_2)\rangle_{(\tilde{U},\tilde{V})}=\frac{1}{2}\left(\mathrm{tr}(\tilde{\xi}_1^T\tilde{\xi}_2)+\mathrm{tr}(\tilde{\eta}_1^T\tilde{\eta}_2)\right), \tag{25}$$

for $(\tilde{\xi}_1,\tilde{\eta}_1),(\tilde{\xi}_2,\tilde{\eta}_2)\in T_{(\tilde{U},\tilde{V})}\left(\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)\right)$.

If we regard $\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$ as a Riemannian submanifold of $\mathcal{SP}(p,m)\times\mathcal{SP}(p,n)$, we can exploit a previous result in [7] to obtain the following proposition.

Proposition 1: For any $(\tilde{B},\tilde{C})\in\mathcal{SP}(p,m)\times\mathcal{SP}(p,n)$, the orthogonal projection operator $P_{(\tilde{U},\tilde{V})}$ onto the tangent space $T_{(\tilde{U},\tilde{V})}(\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n))$ at $(\tilde{U},\tilde{V})$ is given by

$$P_{(\tilde{U},\tilde{V})}(\tilde{B},\tilde{C})=\left(P_{\tilde{U}}(\tilde{B}),P_{\tilde{V}}(\tilde{C})\right),$$

where

$$P_{\tilde{U}}(\tilde{B})=\tilde{B}-\tilde{U}\,\mathrm{sym}\left(\tilde{U}^T\tilde{B}\right), \tag{26}$$

$$P_{\tilde{V}}(\tilde{C})=\tilde{C}-\tilde{V}\,\mathrm{sym}\left(\tilde{V}^T\tilde{C}\right), \tag{27}$$

and where $\mathrm{sym}(\tilde{B}):=(\tilde{B}+\tilde{B}^T)/2$ denotes the symmetric part of $\tilde{B}$.

Proof: On account of the right-hand sides of (26) and (27), it is easy to verify that $P_{(\tilde{U},\tilde{V})}(\tilde{B},\tilde{C})\in\mathcal{SP}(p,m)\times\mathcal{SP}(p,n)$. The remaining task is to show that

$$P_{\tilde{U}}(\tilde{B})^T\tilde{U}+\tilde{U}^TP_{\tilde{U}}(\tilde{B})=P_{\tilde{V}}(\tilde{C})^T\tilde{V}+\tilde{V}^TP_{\tilde{V}}(\tilde{C})=0 \tag{28}$$

and

$$\langle(\tilde{B},\tilde{C})-P_{(\tilde{U},\tilde{V})}(\tilde{B},\tilde{C}),(\tilde{\xi},\tilde{\eta})\rangle_{(\tilde{U},\tilde{V})}=0 \tag{29}$$

for any $(\tilde{\xi},\tilde{\eta})\in T_{(\tilde{U},\tilde{V})}(\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n))$. Equation (28) is an easy consequence of $\tilde{U}^T\tilde{U}=\tilde{V}^T\tilde{V}=I_{2p}$, and (29) results from the fact that the trace of the product of symmetric and skew-symmetric matrices is zero. ∎

B. Retraction

In each iteration of a Riemannian optimization method on a manifold M, for a given search direction $\eta\in T_xM$ at a current point $x\in M$, a search should be performed on a curve emanating from x in the direction of η. For this purpose, it is necessary to find a retraction [1] on the manifold M in question, which is a map from TM to M.

On the real Stiefel manifold $\mathrm{St}(p,n,\mathbb{R})$, there exists a retraction based on the QR decomposition, which is denoted by $R^{\mathbb{R}}$ and defined to be

$$R_Y^{\mathbb{R}}(\xi)=\mathrm{qf}(Y+\xi),\quad Y\in\mathrm{St}(p,n,\mathbb{R}),\ \xi\in T_Y\mathrm{St}(p,n,\mathbb{R}),$$

where $R_Y^{\mathbb{R}}$ is the restriction of $R^{\mathbb{R}}$ to $T_Y\mathrm{St}(p,n,\mathbb{R})$, and where $\mathrm{qf}(\cdot)$ denotes the Q-factor of the QR decomposition of the matrix in the parentheses [1], [7]. However, this $R^{\mathbb{R}}$ cannot apply for the case of $\mathrm{Stp}(p,n)$. This is because even if $\tilde{B}\in\mathcal{SP}(p,n)$, $\mathrm{qf}(\tilde{B})$ no longer belongs to $\mathcal{SP}(p,n)$ in general. An alternative approach is to start with the QR-based retraction $R^{\mathbb{C}}$ on $\mathrm{St}(p,n,\mathbb{C})$, and then to return to $\mathcal{SP}(p,n)$. Here, $R^{\mathbb{C}}$ is defined by

$$R_U^{\mathbb{C}}(\xi)=\mathrm{qf}(U+\xi),\quad U\in\mathrm{St}(p,n,\mathbb{C}),\ \xi\in T_U\mathrm{St}(p,n,\mathbb{C}), \tag{30}$$

where the qf in (30) denotes the Q-factor of the complex QR decomposition. That is, if a full-rank $n\times p$ complex matrix M is decomposed into

$$M=QR,\qquad Q\in\mathrm{St}(p,n,\mathbb{C}),R\in S^+_{\text{upp}}(p),$$

then $\mathrm{qf}(M)=Q$, where $S^+_{\text{upp}}(p)$ denotes the set of all $p\times p$ upper triangular matrices with strictly positive diagonal entries. We then define the QR-based retraction R^{Stp} on $\mathrm{Stp}(p,n)$ as follows:

$$R_{\tilde{U}}^{\mathrm{Stp}}(\tilde{\xi})=\phi\left(R^{\mathbb{C}}_{\phi^{-1}(\tilde{U})}(\phi^{-1}(\tilde{\xi}))\right)=\phi\left(R_U^{\mathbb{C}}(\xi)\right), \\ \tilde{U}\in\mathrm{Stp}(p,n),\ \tilde{\xi}\in T_{\tilde{U}}\mathrm{Stp}(p,n),$$

where ϕ is defined in (18).

A retraction $\tilde{R}$ on $\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$ is immediately defined as

$$\tilde{R}_{(\tilde{U},\tilde{V})}(\tilde{\xi},\tilde{\eta})=\left(R_{\tilde{U}}^{\mathrm{Stp}}(\tilde{\xi}),R_{\tilde{V}}^{\mathrm{Stp}}(\tilde{\eta})\right), \tag{31}$$

$$(\tilde{U},\tilde{V})\in\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n), \\ (\tilde{\xi},\tilde{\eta})\in T_{(\tilde{U},\tilde{V})}(\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)).$$

C. The Gradient and the Hessian

The objective function (23) in Problem 3 is quite similar to the function (1) which is investigated in [7]. The only difference is the factor $1/2$ in (23). However, because of the factor $1/2$ in the metric (25) on $\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$, the gradient and the Hessian of the current objective function F on $\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$ have the same forms as those given in [7].

Proposition 2: For $(\tilde{U},\tilde{V})\in\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$, let $\tilde{S}_1$ and $\tilde{S}_2$ be defined to be $\tilde{S}_1=\mathrm{sym}\left(\tilde{U}^T\tilde{A}\tilde{V}\tilde{N}\right)$ and

$\tilde{S}_2 = \mathrm{sym}\left(\tilde{V}^T\tilde{A}^T\tilde{U}\tilde{N}\right)$, respectively. Then, the gradient of (23) at $(\tilde{U},\tilde{V}) \in \mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$ is expressed as

$$\mathrm{grad}\, F(\tilde{U},\tilde{V}) = \left(\tilde{U}\tilde{S}_1 - \tilde{A}\tilde{V}\tilde{N}, \tilde{V}\tilde{S}_2 - \tilde{A}^T\tilde{U}\tilde{N}\right). \quad (32)$$

Further, let $(\tilde{\xi},\tilde{\eta})$ be a tangent vector at $(\tilde{U},\tilde{V}) \in \mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$. The Hessian of (23) at $(\tilde{U},\tilde{V})$ is viewed as a linear transformation of the tangent space and given by

$$\begin{aligned}&\mathrm{Hess}\, F(\tilde{U},\tilde{V})[(\tilde{\xi},\tilde{\eta})]\\ =&\Big(\tilde{\xi}\tilde{S}_1 - \tilde{A}\tilde{\eta}\tilde{N} - \tilde{U}\,\mathrm{sym}\left(\tilde{U}^T(\tilde{\xi}\tilde{S}_1 - \tilde{A}\tilde{\eta}\tilde{N})\right),\\ &\tilde{\eta}\tilde{S}_2 - \tilde{A}^T\tilde{\xi}\tilde{N} - \tilde{V}\,\mathrm{sym}\left(\tilde{V}^T\left(\tilde{\eta}\tilde{S}_2 - \tilde{A}^T\tilde{\xi}\tilde{N}\right)\right)\Big). \quad (33)\end{aligned}$$

IV. NEWTON'S METHOD AND A NEW COMPLEX SVD ALGORITHM

In this section, we develop a new complex SVD algorithm based on the Riemannian Newton method.

A. Newton's Method for Problem 3

We apply the Riemannian Newton method [1] to Problem 3. For a tangent vector $(\tilde{\xi},\tilde{\eta})$ to $\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$ at $(\tilde{U}_k,\tilde{V}_k) \in \mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$, Newton's equation takes the form

$$\mathrm{Hess}\, F(\tilde{U}_k,\tilde{V}_k)[(\tilde{\xi},\tilde{\eta})] = -\,\mathrm{grad}\, F(\tilde{U}_k,\tilde{V}_k). \quad (34)$$

On substituting (32) and (33) into (34), Newton's equation for (23) can be easily written out. Further, the QR-based retraction $\tilde{R}$ on $\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$ has been given in (31). On the basis of these arrangements, Newton's method for Problem 3 is described as Algorithm 1.

Algorithm 1 Newton's method for Problem 3

1: Choose an initial point $(\tilde{U}_0,\tilde{V}_0) \in \mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$.
2: **for** $k = 0,1,2,\ldots$ **do**
3: Compute the search direction $(\tilde{\xi}_k,\tilde{\eta}_k) \in T_{(\tilde{U}_k,\tilde{V}_k)}(\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n))$ by solving Newton's equations

$$\begin{cases}\tilde{\xi}_k\tilde{S}_{1,k} - \tilde{A}\tilde{\eta}_k\tilde{N} - \tilde{U}_k\,\mathrm{sym}\left(\tilde{U}_k^T(\tilde{\xi}_k\tilde{S}_{1,k}\right.\\ \left.- \tilde{A}\tilde{\eta}_k\tilde{N})\right) = \tilde{A}\tilde{V}_k\tilde{N} - \tilde{U}_k\tilde{S}_{1,k},\\ \tilde{\eta}_k\tilde{S}_{2,k} - \tilde{A}^T\tilde{\xi}_k\tilde{N} - \tilde{V}_k\,\mathrm{sym}\left(\tilde{V}_k^T\left(\tilde{\eta}_k\tilde{S}_{2,k}\right.\right.\\ \left.\left.-\tilde{A}^T\tilde{\xi}_k\tilde{N}\right)\right) = \tilde{A}^T\tilde{U}_k\tilde{N} - \tilde{V}_k\tilde{S}_{2,k},\end{cases}$$

where $\tilde{S}_{1,k} = \mathrm{sym}(\tilde{U}_k^T\tilde{A}\tilde{V}_k\tilde{N})$ and $\tilde{S}_{2,k} = \mathrm{sym}(\tilde{V}_k^T\tilde{A}^T\tilde{U}_k\tilde{N})$.
4: Compute the next iterate

$$(\tilde{U}_{k+1},\tilde{V}_{k+1}) := \tilde{R}_{(\tilde{U}_k,\tilde{V}_k)}(\tilde{\xi}_k,\tilde{\eta}_k),$$

where $\tilde{R}$ is the QR-based retraction on $\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$ defined in (31).
5: **end for**

Algorithm 1 is quite similar to Algorithm 4.3 in [7]. In [7], Newton's equation are divided into a collection of sub-equations by putting $p = 1$ and treating the equation on $\mathrm{St}(1,m,\mathbb{R})\times\mathrm{St}(1,n,\mathbb{R}) = S^{m-1}\times S^{n-1}$. This makes Newton's equation into a vector equation which is easy to solve. However, this division method does not result in an easy-to-perform algorithm for the present Newton's equation. This is because even for $p = 1$, Newton's equations in Algorithm 1 are still matrix equations for $2m\times 2$ and $2n\times 2$ matrices, $\tilde{\xi}_k$ and $\tilde{\eta}_k$, which are still difficult to solve. Furthermore, as we can observe from (17), treating matrices on $\mathrm{Stp}(p,n)$ needs twice as much computer memory as those on $\mathrm{St}(p,n,\mathbb{C})$. Also, addition and multiplication of matrices on $\mathrm{Stp}(p,n)$ need about twice as much computation time as those on $\mathrm{St}(p,n,\mathbb{C})$. To avoid these difficulties, we shall put Algorithm 1 in the complex form in the next subsection.

B. Newton's Method for Problem 1

Through the map ϕ^{-1}, Newton's method for Problem 3 can be translated into Newton's method for Problem 1 on $\mathrm{St}(p,m,\mathbb{C})\times\mathrm{St}(p,n,\mathbb{C})$. In the process of translation, the relations (20) are used together with the relation for $B \in \mathbb{C}^{p\times p}$ and $\tilde{B} = \phi(B) \in \mathcal{SP}(p,p)$,

$$\mathrm{sym}(\tilde{B}) = \frac{\tilde{B}+\tilde{B}^T}{2} \longleftrightarrow \frac{B+B^H}{2} = \mathrm{her}(B),$$

where $\mathrm{her}(\cdot)$ denotes the Hermitian part of the matrix in the parentheses. Further, the retraction $\tilde{R}$ given in (31) on $\mathrm{Stp}(p,m)\times\mathrm{Stp}(p,n)$ corresponds to the retraction R on $\mathrm{St}(p,m,\mathbb{C})\times\mathrm{St}(p,n,\mathbb{C})$ defined by

$$\begin{aligned}&R_{(U,V)}(\xi,\eta) = \left(R_U^{\mathbb{C}}(\xi), R_V^{\mathbb{C}}(\eta)\right) = (\mathrm{qf}(U+\xi), \mathrm{qf}(V+\eta)),\\ &(U,V) \in \mathrm{St}(p,m,\mathbb{C})\times\mathrm{St}(p,n,\mathbb{C}),\\ &(\xi,\eta) \in T_{(U,V)}(\mathrm{St}(p,m,\mathbb{C})\times\mathrm{St}(p,n,\mathbb{C})). \quad (35)\end{aligned}$$

Thus, Algorithm 1 is translated to Algorithm 2 for Problem 1, which provides Newton's method for Problem 1.

Algorithm 2 Newton's method for Problem 1

1: Choose an initial point $(U_0,V_0) \in \mathrm{St}(p,m,\mathbb{C})\times\mathrm{St}(p,n,\mathbb{C})$.
2: **for** $k = 0,1,2,\ldots$ **do**
3: Compute the search direction $(\xi_k,\eta_k) \in T_{(U_k,V_k)}(\mathrm{St}(p,m,\mathbb{C})\times\mathrm{St}(p,n,\mathbb{C}))$ by solving Newton's equations

$$\begin{cases}\xi_k S_{1,k} - A\eta_k N - U_k\,\mathrm{her}\left(U_k^H(\xi_k S_{1,k}\right.\\ \left.-A\eta_k N)\right) = AV_kN - U_kS_{1,k},\\ \eta_k S_{2,k} - A^H\xi_k N - V_k\,\mathrm{her}\left(V_k^H\left(\eta_k S_{2,k}\right.\right.\\ \left.\left.-A^H\xi_k N\right)\right) = A^HU_kN - V_kS_{2,k},\end{cases}$$

where $S_{1,k} = \mathrm{her}(U_k^H AV_kN)$ and $S_{2,k} = \mathrm{her}(V_k^H A^H U_k N)$.
4: Compute the next iterate

$$(U_{k+1},V_{k+1}) := R_{(U_k,V_k)}(\xi_k,\eta_k),$$

where R is the QR-based retraction on $\mathrm{St}(p,m,\mathbb{C})\times\mathrm{St}(p,n,\mathbb{C})$ defined in (35).
5: **end for**

Though Newton's equations in Algorithm 2 are not easy to solve, the problem can be divided into p subproblems which are easy to solve, as is done in [7]. To this end, we treat Newton's equations with $p = 1$ at first. If $p = 1$, then N is a positive real number, and hence we may put $N = 1$ without loss of generality. Furthermore, one has $U_k^H \xi_k = V_k^H \eta_k = 0$, and $S_{1,k} = S_{2,k} = \mathrm{Re}(U_k^H A V_k N) = \mathrm{Re}(U_k^H A V_k)$, where U_k, V_k, ξ_k, η_k are column vectors and S_k is a scalar. In what follows, we replace U_k, V_k, S_k with the lower case symbols u_k, v_k, s_k, respectively, since they are no longer matrices. Then, Newton's equations with $p = 1$ are written out as

$$s_k \xi_k - A\eta_k + u_k \mathrm{Re}(u_k^H A \eta_k) = A v_k - s_k u_k, \quad (36)$$

$$s_k \eta_k - A^H \xi_k + v_k \mathrm{Re}(v_k^H A^H \xi_k) = A^H u_k - s_k v_k. \quad (37)$$

If $s_k \neq 0$, (36) yields

$$\xi_k = s_k^{-1} \left(A(\eta_k + v_k) - u_k \mathrm{Re}(u_k^H A \eta_k) \right) - u_k. \quad (38)$$

Substituting (38) into (37) and simplifying the resulting equation, we obtain the equation for η_k without ξ_k:

$$\begin{aligned}&(s_k^2 I_n - A^H A)\eta_k + (A^H u_k - s_k v_k) \mathrm{Re}(u_k^H A \eta_k) \\ &+ v_k \mathrm{Re}(v_k^H A^H A \eta_k) = A^H A v_k - v_k \mathrm{Re}(v_k^H A^H A v_k). \quad (39)\end{aligned}$$

Let $B_k := s_k^2 I_n - A^H A \in \mathbb{C}^{n \times n}$ and $a_k := A^H u_k - s_k v_k$, $b_k := A^H u_k$, $c_k := A^H A v_k$, $d_k := A^H A v_k - v_k \mathrm{Re}(v_k^H A^H A v_k) \in \mathbb{C}^n$. In terms of these matrices and vectors, (39) is rewritten as

$$B_k \eta_k + a_k \mathrm{Re}(b_k^H \eta_k) + v_k \mathrm{Re}(c_k^H \eta_k) = d_k. \quad (40)$$

We decompose (40) into its real and imaginary parts by introducing real vectors such as $\eta_k = \eta_k^1 + i\eta_k^2$, $\eta_k^1, \eta_k^2 \in \mathbb{R}^n$. The resultant equation is expressed as

$$\begin{aligned}&B_k^1 \eta_k^1 - B_k^2 \eta_k^2 + a_k^1 (b_k^1)^T \eta_k^1 + a_k^1 (b_k^2)^T \eta_k^2 \\ &+ v_k^1 (c_k^1)^T \eta_k^1 + v_k^1 (c_k^2)^T \eta_k^2 \\ &+ i \left(B_k^1 \eta_k^2 + B_k^2 \eta_k^1 + a_k^2 (b_k^1)^T \eta_k^1 + a_k^2 (b_k^2)^T \eta_k^2 \right. \\ &\quad \left. + v_k^2 (c_k^1)^T \eta_k^1 + v_k^2 (c_k^2)^T \eta_k^2 \right) = d_k^1 + i d_k^2.\end{aligned}$$

This equation is put in the real form

$$\boldsymbol{A}_k \boldsymbol{\eta}_k = \boldsymbol{d}_k,$$

where the bold symbols are $\boldsymbol{\eta}_k = \begin{pmatrix} \eta_k^1 \\ \eta_k^2 \end{pmatrix}$, $\boldsymbol{d}_k = \begin{pmatrix} d_k^1 \\ d_k^2 \end{pmatrix}$, $\boldsymbol{A}_k = \begin{pmatrix} B_k^1 + a_k^1 (b_k^1)^T + v_k^1 (c_k^1)^T & -B_k^2 + a_k^1 (b_k^2)^T + v_k^1 (c_k^2)^T \\ B_k^2 + a_k^2 (b_k^1)^T + v_k^2 (c_k^1)^T & B_k^1 + a_k^2 (b_k^2)^T + v_k^2 (c_k^2)^T \end{pmatrix}$.
If $\boldsymbol{A}_k$ is invertible, one has

$$\begin{pmatrix} \eta_k^1 \\ \eta_k^2 \end{pmatrix} = \boldsymbol{\eta}_k = \boldsymbol{A}_k^{-1} \boldsymbol{d}_k,$$

so that $\eta_k = \eta_k^1 + i\eta_k^2$ is found as well. Once η_k is computed, ξ_k is given by (38). By introducing $\boldsymbol{a}_k = \begin{pmatrix} a_k^1 \\ a_k^2 \end{pmatrix}$, $\boldsymbol{b}_k = \begin{pmatrix} b_k^1 \\ b_k^2 \end{pmatrix}$, $\boldsymbol{c}_k = \begin{pmatrix} c_k^1 \\ c_k^2 \end{pmatrix}$, $\boldsymbol{v}_k = \begin{pmatrix} v_k^1 \\ v_k^2 \end{pmatrix}$, these equations take a simpler form. Now we are led to Algorithm 3.

C. Complex SVD Algorithm Based on the Riemannian Newton Method

Let $(U_{\mathrm{app}}, V_{\mathrm{app}})$ be a sufficiently accurate approximate solution to Problem 1 with a general p. We denote by $(\cdot)_j$ the j-th column of the matrix in the parentheses. Then, for each $j \in \{1, \dots, p\}$, the pair $((U_{\mathrm{app}})_j, (V_{\mathrm{app}})_j)$ can be considered to be in the convergence region of $((U_*)_j, (V_*)_j)$ for Algorithm 3, where (U_*, V_*) is an optimal solution to Problem 1. Then, we can solve each of these p subproblems by applying Algorithm 3, and eventually solve Problem 1 after collecting the solutions to the subproblems. We now propose a new complex SVD algorithm as Algorithm 4. Since the problem is divided into p subproblems, Algorithm 4 can be performed by parallel p iterations of Algorithm 3.

Algorithm 3 Newton's method for Problem 1 with $p = 1$

1: Choose an initial point $(u_0, v_0) \in \mathrm{St}(1, m, \mathbb{C}) \times \mathrm{St}(1, n, \mathbb{C})$.
2: **for** $k = 0, 1, 2, \dots$ **do**
3: Compute the search direction $(\xi_k, \eta_k) \in T_{(u_k, v_k)}(\mathrm{St}(1, m, \mathbb{C}) \times \mathrm{St}(1, n, \mathbb{C}))$ by

$$\eta_k = \begin{pmatrix} I_n & iI_n \end{pmatrix} \left(\boldsymbol{B}_k + \begin{pmatrix} \boldsymbol{a}_k & \boldsymbol{v}_k \end{pmatrix} \begin{pmatrix} \boldsymbol{b}_k & \boldsymbol{c}_k \end{pmatrix}^T \right)^{-1} \boldsymbol{d}_k,$$

$$\xi_k = s_k^{-1} \left(A(\eta_k + v_k) - u_k \mathrm{Re}(u_k^H A \eta_k) \right) - u_k,$$

where $s_k = \mathrm{Re}(u_k^H A v_k)$, $\boldsymbol{b}_k = \begin{pmatrix} \mathrm{Re}(A^H u_k) \\ \mathrm{Im}(A^H u_k) \end{pmatrix}$, $\boldsymbol{a}_k = \boldsymbol{b}_k - s_k \begin{pmatrix} \mathrm{Re}(v_k) \\ \mathrm{Im}(v_k) \end{pmatrix}$, $\boldsymbol{c}_k = \begin{pmatrix} \mathrm{Re}(A^H A v_k) \\ \mathrm{Im}(A^H A v_k) \end{pmatrix}$, $\boldsymbol{d}_k = \boldsymbol{c}_k - \mathrm{Re}(v_k^H A^H A v_k) \begin{pmatrix} \mathrm{Re}(v_k) \\ \mathrm{Im}(v_k) \end{pmatrix}$, $\boldsymbol{B}_k = s_k^2 I_{2n} - \begin{pmatrix} \mathrm{Re}(A^H A) & -\mathrm{Im}(A^H A) \\ \mathrm{Im}(A^H A) & \mathrm{Re}(A^H A) \end{pmatrix}$, and where $\mathrm{Im}(\cdot)$ denotes the imaginary part of the quantity in the parentheses.
4: Compute the next iterate

$$(u_{k+1}, v_{k+1}) := R_{(u_k, v_k)}(\xi_k, \eta_k) = \left(\frac{u_k + \xi_k}{\|u_k + \xi_k\|}, \frac{v_k + \eta_k}{\|v_k + \eta_k\|} \right),$$

where $\|\cdot\|$ denotes the standard norm on $\mathbb{C}^m$ and $\mathbb{C}^n$.
5: **end for**

Algorithm 4 Complex SVD algorithm based on Newton's method for Problem 1

Require: A sufficiently accurate approximate solution $(U_{\mathrm{app}}, V_{\mathrm{app}}) \in \mathrm{St}(p, m, \mathbb{C}) \times \mathrm{St}(p, n, \mathbb{C})$ for Problem 1.
1: **for** $j = 1, 2, \dots, p$ **do**
2: Set $(u_0, v_0) := ((U_{\mathrm{app}})_j, (V_{\mathrm{app}})_j)$,
3: Perform Steps 2–5 in Algorithm 3.
4: **end for**
5: Stack the vectors $u_1, \dots, u_p$ and $v_1, \dots, v_p$ to form U and V, respectively:

$$U = (u_1, \dots, u_p), \; V = (v_1, \dots, v_p),$$

where each (u_j, v_j) is obtained by Step 3.

A way to obtain an initial approximate solution is to use the MATLAB's svd function. Another method to obtain an approximate solution is to apply the conjugate gradient method for Problem 1 as in [7], which we omit to discuss in this paper. Our method for obtaining initial approximate solutions is as follows: We first make up several test matrices A of which the exact SVDs, hence an optimal solution (U_*, V_*), are available in advance. Then, approximate initial solutions $(U_{\rm app}, V_{\rm app})$ are made by adding a pair of matrices with small random elements to the exact solution (U_*, V_*).

We set $m = 300$, $n = 10$, and $p = 5$, and then form unitary matrices $U_{\rm SVD} \in U(m)$ and $V_{\rm SVD} \in U(n)$ with randomly chosen elements and fix them in what follows. We proceed to compute $A_j = U_{\rm SVD}\Sigma_j V_{\rm SVD}^H$, $\Sigma_j = \begin{pmatrix} D_j \\ 0 \end{pmatrix}$, for the $n \times n$ diagonal matrices D_j given below:

$$D_1 = \mathrm{diag}(10, 9, \ldots, 1),$$

$$D_2 = \mathrm{diag}(100, 99, \ldots, 92, 1),$$

$$D_3 = \mathrm{diag}(100, 99, \ldots, 96, 5, 4, \ldots, 1),$$

$$D_4 = \mathrm{diag}(1000, 999, \ldots, 992, 1),$$

$$D_5 = \mathrm{diag}(9.64, 8.97, 8.19, 7.77, 5.55, 5.02, 4.23, 4.10, 3.60, 0.29),$$

where the singular values of A_5 (or the diagonal elements of D_5) are randomly chosen out of the interval $[0, 10]$. The condition numbers of the matrices A_j are $\mathrm{cond}(A_1) = 10$, $\mathrm{cond}(A_2) = \mathrm{cond}(A_3) = 100$, $\mathrm{cond}(A_4) = 1000$, $\mathrm{cond}(A_5) = 32.92$, respectively. From the very definition of A_j, the columns of $U_{\rm SVD}$ and $V_{\rm SVD}$ are exactly the left and right singular vectors of A_j, $j = 1, \ldots, 5$. Therefore, the $(U_{\rm opt}, V_{\rm opt})$ defined by

$$U_{\rm opt} = U_{\rm SVD} I_{m,p}, \ V_{\rm opt} = V_{\rm SVD} I_{n,p}$$

is an optimal solution to Problem 1 with $A = A_j$, where $I_{n,p}$ is defined to be $I_{n,p} = \begin{pmatrix} I_p \\ 0 \end{pmatrix} \in \mathbb{R}^{n\times p}$. An approximate initial solution $(U_{\rm app}, V_{\rm app}) \in \mathrm{St}(p, m, \mathbb{C}) \times \mathrm{St}(p, n, \mathbb{C})$ is then formed by

$$U_{\rm app} = \mathrm{qf}(U_{\rm opt} + U_{\rm rand}), \ V_{\rm app} = \mathrm{qf}(V_{\rm opt} + V_{\rm rand}),$$

where $U_{\rm rand} \in \mathbb{C}^{m\times p}$ and $V_{\rm rand} \in \mathbb{C}^{n\times p}$ are randomly chosen matrices with elements less than 0.05 in absolute values. For example, the difference in the values of the objective function is $F(U_{\rm app}, V_{\rm app}) - F(U_{\rm opt}, V_{\rm opt}) = 12.30$ for the matrix A_1. We apply Algorithm 4 with $(U_{\rm app}, V_{\rm app})$ as initial data to obtain Fig. 1, which shows that the differences between the values $F(U_k, V_k)$ and the minimum values $F(U_{\rm opt}, V_{\rm opt})$ of F decrease rapidly against the iteration number k for any test matrices A_j. For $A = A_1$, the computer decides that $F(U_6, V_6) - F(U_{\rm opt}, V_{\rm opt}) = 0$ within a machine epsilon at $k = 6$. For $A = A_2, A_3, A_4, A_5$, the computer decides that the current iterate at $k = 4, 3, 5, 4$ is an optimal solution within a machine epsilon, respectively. We here note that the fact that each graph in Fig. 1 ends at some iteration number means that the difference reaches 0 within computer accuracy at the next iterate.

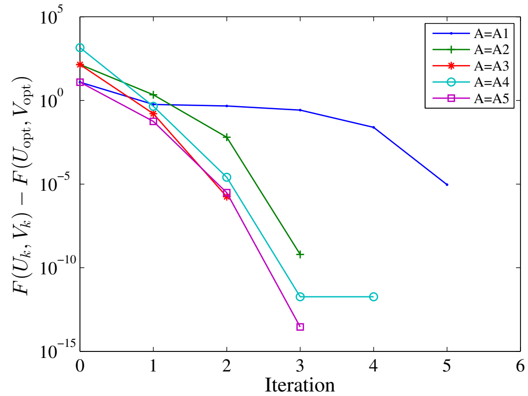

Fig. 1. The differences between the optimal and the current values of the objective functions with $A = A_1, A_2, \ldots, A_5$.

V. CONCLUSIONS

We have formulated the complex SVD problem as an optimization problem on $\mathrm{St}(p, m, \mathbb{C}) \times \mathrm{St}(p, n, \mathbb{C})$. The problem has been further rewritten as that on the real form $\mathrm{Stp}(p, m) \times \mathrm{Stp}(p, n)$ of $\mathrm{St}(p, m, \mathbb{C}) \times \mathrm{St}(p, n, \mathbb{C})$. In developing Newton's method for the problem on $\mathrm{Stp}(p, m) \times \mathrm{Stp}(p, n)$, the results obtained in [7] for the real SVD case have been extensively used. Consequently, we have developed Newton's method for the problem on $\mathrm{St}(p, m, \mathbb{C}) \times \mathrm{St}(p, n, \mathbb{C})$. Though Newton's equation in the algorithm is difficult to solve, the division of the problem into p subproblems makes Newton's equations easy to solve.

We have performed numerical experiments with the presented algorithm. The results show that the proposed algorithm can improve a given approximate SVD within computer accuracy.

REFERENCES

[1] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, NJ, 2008.
[2] P.-A. Absil, R. Sepulchre, P. Van Dooren, and R. Mahony, A Newton algorithm for invariant subspace computation with large basins of attraction, *Proc. of the 42nd IEEE Conference on Decision and Control*, pp. 2352–2357, 2003.
[3] G. Adams, A. M. Finn, and M. F. Griffin, A Fast Implementation of the Complex Singular Value Decomposition on the Connection Machine, *IEEE Trans. Acoust. Speech Signal Process*, pp. 1129–1132, 1991.
[4] A. Edelman, T. A. Arias, and S. T. Smith, The geometry of algorithms with orthogonality constraints, *SIAM J. Matrix Anal. Appl.*, **20**(2), pp. 303–353, 1998.
[5] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Third ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
[6] N. Hemkumar and J. Cavallaro, A Systolic VLSI Architecture for Complex SVD, *Proc. of the IEEE International Symposium on Circuits and Systems*, pp. 1061–1064, 1992.
[7] H. Sato and T. Iwai, A Riemannian Optimization Approach to the Matrix Singular Value Decomposition, *SIAM J. Opt.*, **23**(1), pp. 188–212, 2013.
[8] S.-G. Wang and L. Bai, Flexible Robust Sliding Mode Control for Uncertain Stochastic Systems with Time-Varying Delay and Structural Uncertainties, *Proc. of the 51st IEEE Conference on Decision and Control*, pp. 1536–1541, 2012.
[9] C.-Z. Zhan, Y.-L. Chen, and A.-Y. Wu, Iterative Superlinear-Convergence SVD Beamforming Algorithm and VLSI Architecture for MIMO-OFDM Systems, *IEEE Trans. Signal Process.*, **60**(6), pp. 3264–3277, 2012.