Tech. report FSU 15-03

Solving PhaseLift by low-rank Riemannian optimization methods for complex semidefinite constraints

Wen Huang · K. A. Gallivan · Xiangxiong Zhang

Abstract A framework, PhaseLift, was recently proposed to solve the phase retrieval problem. In this framework, the problem is solved by optimizing a cost function over the set of complex Hermitian positive semidefinite matrices. This approach to phase retrieval motivates a more general consideration of optimizing cost functions on semidefinite Hermitian matrices where the desired minimizers are known to have low rank. This paper considers an approach based on an alternative cost function defined on a union of appropriate manifolds. It is related to the original cost function in a manner that preserves the ability to find a global minimizer and is significantly more efficient computationally. A rank-based optimality condition for stationary points is given and optimization algorithms based on state-of-the-art Riemannian optimization and dynamically reducing rank are proposed. Empirical evaluations are performed using the PhaseLift problem. The new approach is shown to be

This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office. This work was supported by grant FNRS PDR T.0173.13.

Wen Huang
Department of Mathematical Engineering, Université catholique de Louvain,
ICTEAM Institute, B-1348 Louvain-la-Neuve, Belgium
E-mail: wen.huang@uclouvain.be Tel.: +32-10-478005
Fax: +32-10-472180

K. A. Gallivan
Department of Mathematics, Florida State University,
208 Love Building, 1017 Academic Way, Tallahassee FL 32306-4510, USA
E-mail: kgallivan@fsu.edu

Xiangxiong Zhang
Department of Mathematics, Purdue University,
150 N. University Street, West Lafayette, IN 47907-2067
E-mail: zhan1966@purdue.edu

an effective method of phase retrieval with computational efficiency increased substantially compared to the algorithm used in original PhaseLift paper.

Keywords Riemannian Optimization · Low rank optimization · Complex optimization · Phase Retrieval · PhaseLift

1 Introduction

The phase retrieval problem concerns recovering a signal given the modulus of its Fourier transform. It is a key problem for many important applications, e.g., X-ray crystallography imaging [Har93], diffraction imaging [BDP+07], optics [Wal63] and microscopy [MISE08]. The continuous form of the problem recovers $x(t) : \mathbb{R}^s \to \mathbb{C}$ from $|\tilde{x}(u)|$, where $\tilde{x}(u) : \mathbb{R}^s \to \mathbb{C}$ is defined by

$$\tilde{x}(u) = \int_{\mathbb{R}^s} x(t) \exp(-2\pi u \cdot t\sqrt{-1}) dt,$$

and $\cdot$ denotes the Euclidean inner product. This paper considers the discrete form of the problem where an indexed set of complex numbers $\mathbf{x} \in \mathbb{C}^{n_1 \times n_2 \times \ldots \times n_s}$ is to be recovered from the modulus of its discrete Fourier transform $|\tilde{\mathbf{x}}(g_1, g_2, \ldots, g_s)|$, where $(g_1, g_2, \ldots, g_s) \in \Omega := G_1 \times G_2 \times \ldots G_s$ and Ω is a grid of an s-dimensional space. The discrete Fourier transform $\tilde{\mathbf{x}}$ is given by

$$\tilde{\mathbf{x}}(g_1, g_2, \ldots, g_s) = \frac{1}{\sqrt{n}} \sum_{i_1, i_2, \ldots, i_s} \mathbf{x}_{i_1 i_2 \ldots i_s} \exp\left(-2\pi\left(\frac{(i_1 - 1)g_1}{n_1} + \ldots + \frac{(i_s - 1)g_s}{n_s}\right)\sqrt{-1}\right), \quad (1.1)$$

where $n = n_1 n_2 \ldots n_s$, i_j is an integer satisfying $1 \leq i_j \leq n_j$ for $j = 1, \ldots s$, $\mathbf{x}_{i_1 i_2 \ldots i_s}$ denotes the corresponding entry of $\mathbf{x}$ and $\tilde{\mathbf{x}}(g_1, g_2, \ldots, g_s)$ denotes the corresponding entry of $\tilde{\mathbf{x}}$.

It is well-known that the solution of the phase retrieval is not unique. For example, when Ω is the uniform grid, i.e, $G_i = \{0, 1, \ldots, n_i - 1\}, i = 1, 2, \ldots, s$, if $\mathbf{x}$ is a solution, then

1. $\mathbf{y} = c\mathbf{x}$ is a solution where $c \in \mathbb{C}$ and $|c| = 1$;
2. $\mathbf{y} \in \mathbb{C}^{n_1 \times n_2 \times \ldots n_s}$ such that $\mathbf{y}_{i_1 i_2 \ldots i_s} = \mathbf{x}_{j_1 j_2 \ldots j_s}$ is a solution, where $j_k = i_k + a_k \mod n_k, k = 1, \ldots, s$ and $a_1, a_2, \ldots a_s$ are integers;
3. $\mathbf{y} \in \mathbb{C}^{n_1 \times n_2 \times \ldots n_s}$ such that $\mathbf{y}_{i_1, i_2, \ldots i_s} = \bar{\mathbf{x}}_{j_1 j_2 \ldots j_s}$ is a solution, where $j_k = -i_k \mod n_k, k = 1, \ldots, s$ and $\bar{\mathbf{x}}$ is the conjugate of $\mathbf{x}$.

These equivalent solutions are called trivial associates of $\mathbf{x}$ and infinitely many additional solutions may exist [San85].

Oversampling in the Fourier domain is a standard method to obtain a unique solution and it has been shown to almost always give a unique solution for multiple dimensional problems for real-valued and nonnegative signals [BS79, Hay82, San85]. Many algorithms based on alternating projection

[GS72] have been developed to solve phase retrieval problem using the oversampling framework [Fie78, Fie82, Els03, Bla04, Mar07, CMWL07]. While these algorithms are efficient and effective in some problem settings, they may not perform well in other settings. For details on the capabilities and difficulties of these algorithms see [CESV13] and the references therein.

In recent years other frameworks, using multiple structured illuminations, or the mathematically equivalent construct of masks, combined with convex programming, have been proposed to recover the phase exactly, e.g., PhaseLift [CESV13] and PhaseCut [WDM13]. It was later proved that a feasibility problem of two convex sets can be solved for PhaseLift in [CL13, DH14]. For the PhaseLift framework, four major results are of interest here. First, using a small number (related to s) of noiseless measurements of the modulus defined by certain carefully designed illuminations, the phase can be recovered exactly [CESV13]. Second, when these carefully designed measurements are not used, exact recovery is still possible using $O(n^2)$ noiseless measurements [CL13]. Third, the phase can be recovered exactly with high probability using $O(n \log n)$ noiseless measurements of the modulus [CSV13]. Finally, the stability of recovering the phase using noisy measurements is shown in [CSV13].

For the PhaseCut framework, it is known that if the phase can be recovered using PhaseLift, then it can also be recovered by a modified version of PhaseCut and that the PhaseCut is at least as stable as the weak formulation of PhaseLift for noisy measurements [WDM13]. The weak formulation is formally defined in [WDM13, Section 4.1], however, the idea of a weak formulation is also given earlier in the proof of [CESV13, Theorem 2.1]. Empirically, PhaseCut is observed to be more stable in the situation of sparse sampling of the modulus.

The problems in both PhaseLift and PhaseCut concern optimizing convex cost functions defined on a convex set of complex matrices, i.e.,

$$\min_{X \in \mathcal{D}_n} H(X), \tag{1.2}$$

where $H : \mathcal{D}_n \to \mathbb{R} : X \mapsto H(X)$, and $\mathcal{D}_n$ denotes the set of all n-by-n complex Hermitian positive semidefinite matrices. PhaseCut further requires that the diagonal entries of X are 1. However, the dimension of (1.2) is usually too large to be solved by standard convex programming techniques. For example, in order to recover an image of 100 by 100 pixels, i.e., $s = 2$ and $n_1 = n_2 = 100$, solving an optimization problem with an argument that is a 100^2 by 100^2 matrix is required. The complexity of solving PhaseLift and PhaseCut using standard semidefinite programming solvers, e.g., SDPT3 [TTT99], is discussed in [WDM13, Section 4.6].

Since the desired optimum, X_*, is known to be a rank-one matrix, a low-rank matrix approximation of the argument matrix is used in [CESV13] to save computations for PhaseLift. While this approximation has good empirical performance, no convergence proof is given in [CESV13]. For PhaseCut, a block coordinate descent algorithm is proposed in [WDM13] and the algorithm is shown to be computationally inexpensive for each iteration. However, the

block coordinate descent algorithm converges slowly, i.e., linear convergence [BV04, Section 9.4.3], and the overall computational cost can be unacceptably high.

This paper uses the framework of PhaseLift and an alternate cost function $F : \mathbb{C}^{n\times p} \to \mathbb{R} : Y \mapsto F(Y) = H(YY^*)$ defined by matrix factorization is considered. Even though F is not convex, it is shown to be a suitable replacement of the cost function H. Riemannian optimization methods on an appropriate quotient space are used for optimizing $\mathrm{t}F$. Using the cost function F with a small dimension p reduces storage and the computational complexity of each iteration. Convergence at superlinear rates is also guaranteed theoretically by known Riemannian optimization results. This new approach is shown to perform empirically much better than the low-rank approximate version of the algorithm used for PhaseLift in [CESV13] from the points of view of efficiency and effectiveness. Finally, note that the analysis and algorithm presented is not specific to the cost function used for phase retrieval in PhaseLift but for a general cost function defined on $\mathcal{D}_n$ and therefore the approach has potential for optimization in other applications where the global optimum is known to have low rank.

The idea of using low-rank factorization to solve positive semidefinite constrained problems is, of course, not new but all the research results of which the authors' are aware, are for real positive semidefinite matrix constraints. Burer and Monteiro [BM03] first investigate this approach for semidefinite programming (SDP) in which the cost function is linear. Journée et al. [JBAS10] use low-rank factorization for a more general problem in the sense that the cost function H is not necessary linear,

$$\begin{aligned} &\min_{X\in\mathbb{S}_n^+} H(X), \\ &\text{such that } \operatorname{tr}(A_iX) = b_i, i = 1, \dots m, \end{aligned}$$

where $\mathbb{S}_n^+$ denotes the set of all real n-by-n symmetric positive semidefinite matrices, $A_i \in \mathbb{R}^{n\times n}$, $A_i = A_i^T$ and $A_iA_j = 0$ for any $i \neq j$. The conditions that $A_i = A_i^T$ and $A_iA_j = 0$ for any $i \neq j$ implies the number of equality constraints m is at most n as pointed out in [JBAS10]. The complex problem (1.2) does not belong to this category of problem and details of important differences and a discussion of the geometry are given in Section 4.

The paper is organized as follows. Section 2 presents the notation used. The derivation of the optimization problem framework in PhaseLift is given in Section 3. The alternate cost function and optimality conditions are derived in Section 4. Riemannian optimization methods and the required geometric objects are presented in Section 5. In Section 6, the effectiveness of the methods are demonstrated with several numerical experiments and, finally, conclusions are given in Section 7.

2 Notation

For any $\mathbf{z} \in \mathbb{C}^{n_1 \times n_2 \times \ldots n_s}$, $\mathrm{vec}(\mathbf{z}) \in \mathbb{C}^n$, where $n = n_1 n_2 \ldots n_s$, denotes the vector form of $\mathbf{z}$, i.e.,

$$(\mathrm{vec}(\mathbf{z}))_k = \mathbf{z}_{i_1 i_2 \ldots i_s},$$

where $k = i_1 + \sum_{j=1}^{s-1} n_1 n_2 \ldots n_j (i_{j+1} - 1)$. $\mathrm{Re}(\cdot)$ denotes the real part of the argument and superscript $*$ denotes the conjugate transpose operator. A matrix or a vector with superscript $\check{}$ means it is a real matrix or a real vector. Given a vector v with length h, $\mathrm{Diag}(v)$ denotes an h-by-h diagonal matrix the diagonal entries of which are v. $\sqrt{v}$ denotes a vector with entries that are square root of corresponding entries in v.

$0_{s\times k}$ denotes an $s \times k$ zero matrix; $I_{s\times k}$ denotes a diagonal matrix with diagonal entries 1; and 0_s denote a vector with length s with entries all 0. $\mathrm{diag}(M)$ denotes a vector of the diagonal entries of $M \in \mathbb{C}^{s\times k}$ and $\mathrm{tr}(M)$ denotes the trace of M. If $s \geq k$, $M_\perp$ denotes an $s \times (s-k)$ matrix such that $M_\perp^* M_\perp = I_{(s-k)\times(s-k)}$ and $M_\perp^* M = 0_{(s-k)\times k}$. $M(:, 1:k)$ denotes a matrix that is formed by the first k columns of matrix M. $\mathrm{span}(M)$ denotes the column space of M. E_{ij} denotes a matrix with i-th row j-th column entry be 1 and other entries be 0.

Given an embedded submanifold $\mathcal{M} \subseteq \mathbb{C}^{s\times k}$, $\mathrm{T}_x \mathcal{M}$ and $\mathrm{N}_x \mathcal{M}$ denote the tangent space and normal space of $\mathcal{M}$ at $x \in \mathcal{M}$ respectively. $\mathcal{D}_k$ denotes set $\{X \in \mathbb{C}^{n\times n} | X = X^*, X \geq 0, \mathrm{rank}(X) \leq k\}, 1 \leq k \leq n$. Note that the statement $X \geq 0$ means that matrix X is positive semidefinite. $\mathrm{St}(k, s)$ denotes the complex compact Stiefel manifold $\{A \in \mathbb{C}^{s\times k} | A^* A = I_{k\times k}\}$ with $s \geq k$. $\mathrm{S}_+^{\mathbb{C}}(k, s)$ denotes the set of all Hermitian positive semidefinite $s \times s$ matrices of fixed rank k. When elements of $\mathrm{S}_+^{\mathbb{C}}(k, s)$ are restricted to be real, it is denoted by $\mathrm{S}_+^{\mathbb{R}}(k, s)$. $\mathbb{C}_*^{s\times k}$ denotes the complex noncompact Stiefel manifold, i.e., the set of all $s \times k$ full column rank complex matrices. $\mathcal{O}_s$ denotes the group of s-by-s unitary matrices.

Given a function $f(x)$ on $\mathcal{M}$ or $\mathbb{C}^{s\times k}$, $\mathrm{grad}\, f(x)$ denotes the gradient of f at x, $\mathrm{Hess}\, f(x)$ denotes the Hessian of f at x and $\mathrm{Hess}\, f(x)[\eta]$ denotes the action of $\mathrm{Hess}\, f(x)$ along direction η. These are Riemannian or Euclidean depending on the domain.

The standard Euclidean metric is denoted

$$g^E(Y, \hat{Y}) = \mathrm{Re}(\mathrm{tr}(Y^* \hat{Y})),$$

for $Y, \hat{Y} \in \mathbb{C}^{s\times t}$ and the orthogonal projection with respect to g^E from a point X to a convex set $\mathcal{L}$ is denoted $P_{\mathcal{L}}(X)$.

3 The PhaseLift Approach to Phase Retrieval

3.1 Exact Measurement Optimization Problems

The PhaseLift framework uses a grid, G_i, in each dimension, $i = 1, \ldots, s$, to define the discrete Fourier domain of interest in $\mathbb{C}^{t_1 \times t_2 \times \ldots t_s}$ and $G_i(j)$ denotes

the j-th entry in grid $G_i, i = 1, \ldots, s$. It is assumed that entries in G_i are increasing, i.e., $G_i(h) > G_i(k)$ if $h > k$.

The known random masks or illumination fields defined on the discrete signal domain are denoted $\mathbf{w}_r \in \mathbb{C}^{n_1 \times n_2 \times \ldots n_s}$, $r = 1, \ldots l$. It follows that $\tilde{\mathbf{x}} \in \mathbb{C}^{t_1 \times t_2 \times \ldots t_s}$ of (1.1) can be expressed as

$$(\tilde{\mathbf{x}}_i)_{j_1 j_2 \ldots j_s} = \frac{1}{\sqrt{n}} \sum_{i_1, i_2, \ldots, i_s} \Bigg((\mathbf{w}_i)_{i_1 i_2 \ldots i_s} \mathbf{x}_{i_1 i_2 \ldots i_s} \exp\left(-2\pi \left(\frac{G_1(j_1)(i_1 - 1)}{n_1} + \ldots + \frac{G_s(j_s)(i_s - 1)}{n_s} \right) \sqrt{-1} \right) \Bigg), \tag{3.1}$$

where the subscripts of $\tilde{\mathbf{x}}_i, \mathbf{w}_i$ and $\mathbf{x}$ denote the index in the appropriate discrete domain. Let x, $\tilde{x}_i$ and w_i denote $\mathrm{vec}(\mathbf{x})$, $\mathrm{vec}(\tilde{\mathbf{x}}_i)$ and $\mathrm{vec}(\mathbf{w}_i)$ respectively. Using the Kronecker product, (3.1) can be rewritten as

$$\tilde{x}_i = (\mathcal{F}_{n_s} \otimes \mathcal{F}_{n_{s-1}} \otimes \ldots \mathcal{F}_{n_1}) \operatorname{Diag}(w_i) x,$$

where $\mathcal{F}_{n_i} \in \mathbb{C}^{t_i \times n_i}, i = 1, \ldots, s$ denotes the one-dimensional Discrete Fourier Transform (DFT), i.e.,

$$(\mathcal{F}_{n_i})_{hk} = \frac{1}{\sqrt{n_i}} \exp(-2\pi G_i(h)(k-1)\sqrt{-1}/n_i), h = 1, \ldots, t_i, k = 1, \ldots, n_i.$$

Let Z_i denote $(\mathcal{F}_{n_s} \otimes \mathcal{F}_{n_{s-1}} \otimes \ldots \mathcal{F}_{n_1}) \operatorname{Diag}(w_i)$, Z denote $(Z_1^T \ Z_2^T \ \ldots \ Z_l^T)^T$ and $\tilde{x}$ denote $(\tilde{x}_1^T \ \tilde{x}_2^T \ \ldots \ \tilde{x}_l^T)^T$. The noiseless measurements $b \in \mathbb{R}^m$ are given by

$$b = \operatorname{diag}(\tilde{x}\tilde{x}^*) = \operatorname{diag}(Zxx^*Z^*),$$

where $m = nl$. The task is to recover $x \in \mathbb{C}^n$ that satisfies

$$b = \operatorname{diag}(Zxx^*Z^*),$$

given Z and b. This is a feasible problem of a quadratic equation that is equivalent to a feasible problem of a linear equation with rank constraints, i.e.,

$$\begin{aligned} &\text{find } X \in \mathbb{C}^{n \times n} \\ &\text{such that } b = \operatorname{diag}(ZXZ^*), X \in \mathcal{D}_n, \text{ and } \operatorname{rank}(X) = 1. \end{aligned} \tag{3.2}$$

The alternative problem suggested in [CESV13] considers an optimization problem that does not force the rank of matrix to be one but adds a nuclear norm penalty term to favor low-rank solutions

$$\min_{X \in \mathcal{D}_n} \|b - \operatorname{diag}(ZXZ^*)\|_2^2 + \kappa \operatorname{tr}(X), \tag{3.3}$$

where κ is a positive constant. Furthermore, it was proved in [CL13, DH14] that a feasibility problem of two convex sets can be solved instead of (3.3). Solving the feasibility problem in [CL13, DH14] is equivalent to solving (3.3) with $\kappa = 0$.

3.2 Noisy Measurement Optimization Problems

Measurements with noise, $b \in \mathbb{R}^m$, are assumed to have the form

$$b = \operatorname{diag}(Zxx^*Z^*) + \epsilon,$$

where $\epsilon \in \mathbb{R}^m$ is noise sampled from a distribution $p(:;\mu)$, and the task is to minimize the negative log-likelihood function

$$\begin{aligned} &\min_x -\log(p(b;\mu)) \\ &\text{such that } \mu = \operatorname{diag}(Zxx^*Z^*), \end{aligned}$$

which is equivalent to

$$\begin{aligned} &\min_X -\log(p(b;\mu)) \\ &\text{such that } \mu = \operatorname{diag}(ZXZ^*),\ X \in \mathcal{D}_n \text{ and } \operatorname{rank}(X) = 1. \end{aligned}$$

The alternate problem suggested in [CESV13] is

$$\begin{aligned} &\min_X -\log(p(b;\mu)) + \kappa \operatorname{tr}(X) \\ &\text{such that } \mu = \operatorname{diag}(ZXZ^*) \text{ and } X \in \mathcal{D}_n, \end{aligned} \tag{3.4}$$

or equivalently

$$\min_{X \in \mathcal{D}_n} -\log(p(b;\operatorname{diag}(ZXZ^*))) + \kappa \operatorname{tr}(X) \tag{3.5}$$

where κ is a positive constant. Problems (3.4) and (3.5) are preferred over Problem (3.2), since they are convex programming problems when the log-likelihood function is concave.

The Poisson and Gaussian distributions are given in [CESV13] as examples that have this kind of log-likelihood function. If $b = (b_1, \ldots, b_m)^T$ and $\mu = (\mu_1, \ldots, \mu_m)^T$ with $b_i, i = 1, \ldots, m$ drawn from a Poisson distribution with mean μ_i, then Problem (3.5) becomes

$$\min_{X \in \mathcal{D}_n} \sum_{i=1}^{m} (\mu_i - b_i \log \mu_i) + \kappa \operatorname{tr}(X),$$

where $\mu = \operatorname{diag}(ZXZ^*)$. If $b_i, i = 1, \ldots, m$ are drawn from Gaussian distribution with mean μ_k and variance δ_k, then Problem (3.5) becomes

$$\begin{aligned} &\min_{X \in \mathcal{D}_n} \sum_{i=1}^{m} \frac{1}{2\sigma_i^2}(b_i - \mu_i)^2 + \kappa \operatorname{tr}(X) \\ &= \min_{X \in \mathcal{D}_n} (b - \operatorname{diag}(ZXZ^*))^T \Gamma (b - \operatorname{diag}(ZXZ^*)) + \kappa \operatorname{tr}(X) \end{aligned} \tag{3.6}$$

where $\Gamma = \operatorname{Diag}(1/\sigma_1^2, \ldots, 1/\sigma_m^2)$.

4 Theoretical Results

This section presents theoretical results that motivate the design of algorithms for optimizing a cost function H defined on $\mathcal{D}_n$. The analysis does not rely on the convexity of the cost function H. First, Section 4.1 presents an alternate cost function F to replace H where H can be any of the cost functions discussed in Section 3. In order to make F be a suitable replacement, it is crucial to establish optimality conditions for H based on the properties of F. Since an n dimensional complex vector can be viewed as a $2n$ dimensional real vector, functions F and H can be viewed either on a complex space or on a real space. It follows that the optimality conditions can be obtained by two approaches. Section 4.2 presents the definitions of stationary points for real and complex versions of functions F and H from a Euclidean point of view. Section 4.3 discusses the properties of the domain $\mathcal{D}_n$ from a manifold point of view. Finally, the main results, optimality conditions for H based on the properties of F, are given in Section 4.4.

4.1 Equivalent Cost Function

The cost functions generically denoted H all satisfy

$$H : \mathcal{D}_n \to \mathbb{R} : X \mapsto H(X).$$

It is well-known that for any $X \in \mathcal{D}_n$, there exists $Y_n \in \mathbb{C}^{n\times n}$ such that $Y_n Y_n^* = X$. Furthermore, if X is rank p, then there exists $Y_p \in \mathbb{C}^{n\times p}$ such that $Y_p Y_p^* = X$. Throughout this paper, the subscript of Y is used to emphasize the column size of Y. Therefore, a surjective mapping between $\mathbb{C}^{n\times p}$ and $\mathcal{D}_p$ is given by

$$\alpha_p : \mathbb{C}^{n\times p} \to \mathcal{D}_p : Y_p \mapsto Y_p Y_p^*.$$

It is clear that α_p is not an injection. Specifically, given $X \in \mathcal{D}_p$, if Y_p satisfies $\alpha_p(Y_p) = Y_p Y_p^* = X$, then $Y_p O_p$ also satisfies $\alpha_p(Y_p O_p) = X$ for any $O_p \in \mathcal{O}_p$. Thus, if the desired solution of H is known to be at most rank p, then an alternate cost function to H can be used:

$$F_p : \mathbb{C}^{n\times p} \to \mathbb{R} : Y_p \mapsto H(\alpha_p(Y_p)) = H(Y_p Y_p^*).$$

The subscripts of F and α indicate the column size of the argument. The domain of F_p has lower dimension than that of H which may yield computational efficiency. Therefore, an alternate problem of (1.2) is considered

$$\min_{Y_p \in \mathbb{C}^{n\times p}} F_p(Y_p). \tag{4.1}$$

4.2 Stationary Points

In order to understand the Euclidean gradients and Hessians, and stationary points of F and H as well as important relationships between them, it is useful to consider the structure of the spaces and functions from a real point of view. This section develops that point of view and discusses the implications with respect to the complex forms of F and H.

Let a superscript $\check{}$ denote the mapping

$$\check{}: \mathbb{C}^{n\times p} \to \mathbb{R}^{2n\times 2p} : Y = \check{Y}_1 + \check{Y}_2\sqrt{-1} \mapsto \check{Y} = \begin{pmatrix} \check{Y}_1 & -\check{Y}_2 \\ \check{Y}_2 & \check{Y}_1 \end{pmatrix}$$

which is an isometry from $\mathbb{C}^{n\times p}$ to $\mathbb{R}^{2n\times 2p}$ [GW04]. It preserves the operations of addition and multiplication when the sizes of matrices allow. Note the mapping $\check{}$ is an injection but not a surjection.

For any matrix $X = \check{X}_1 + \check{X}_2\sqrt{-1} \in \mathcal{D}_n$ the mapping $\check{}$ yields

$$\check{X} = \begin{pmatrix} \check{X}_1 & -\check{X}_2 \\ \check{X}_2 & \check{X}_1 \end{pmatrix} = \hat{X} = \begin{pmatrix} \check{X}_1 & \check{X}_2^T \\ \check{X}_2 & \check{X}_3 \end{pmatrix} \in \mathbb{S}^+_{2n},$$

since $\check{X}_2^T = -\check{X}_2$ where $\check{X}_3 := \check{X}_1$. Thus, (1.2) can be formulated as a problem with real semidefinite constraints:

$$\begin{aligned} \min_{\hat{X}\in\mathbb{S}^+_{2n}} \ & \tilde{H}(\hat{X}) := H(\check{X}_1 + \check{X}_2\sqrt{-1}) \\ \text{such that } & \operatorname{tr}(A_k\hat{X}) = 0, k = 1, 2, \ldots, n(n+1), \end{aligned} \tag{4.2}$$

where A_k, $k = 1, \ldots, n(n+1)/2$ are given by

$$A_k = \begin{pmatrix} 0_{n\times n} & E_{ij} + E_{ji} \\ E_{ij} + E_{ji} & 0_{n\times n} \end{pmatrix}, \quad i = 1, \ldots, n, \ \ j = i, \ldots, n$$

and the $n(n+1)/2$ remaining A_k, are given by

$$A_k = \begin{pmatrix} E_{ij} + E_{ji} & 0_{n\times n} \\ 0_{n\times n} & -E_{ij} - E_{ji} \end{pmatrix}, \quad i = 1, \ldots, n, \ \ j = i, \ldots, n$$

where $E_{ij} \in \mathbb{R}^{n\times n}$ are the standard basis matrices.

The real semidefinite formulation (4.2) of the complex problem (1.2) is different than the related real problem on symmetric semidefinite matrices in [JBAS10]. The conditions given in [JBAS10] do not hold for (4.2) since the number of constraints $n(n+1)$ in (4.2) is greater than the size of the argument $2n$ for $n > 1$ which is required in their formulation.

Since $\tilde{H}$ in (4.2) is defined on a real space, [JBAS10, Definition 1] is applicable:

Definition 1 A stationary point of (4.2) is a symmetric matrix $\hat{X} \in \mathbb{R}^{2n\times 2n}$ for which there exists a vector $\delta = (\delta_1, \ldots, \delta_m)^T \in \mathbb{R}^m$ and a symmetric matrix $S \in \mathbb{R}^{2n\times 2n}$ such that the first-order optimality conditions hold:

$$\operatorname{tr}(A_i\hat{X}) = 0, \quad \hat{X} \geq 0, \quad S \geq 0,$$
$$S\hat{X} = 0, \quad S = \operatorname{grad}\tilde{H}(\hat{X}) - \sum_{i=1}^{m}\delta_i A_i,$$

where $m = n(n+1)$.

Similarly, let Y_p be denoted by $\check{Y}_1 + \check{Y}_2\sqrt{-1}$ and $\hat{Y}_p$ be denoted by

$$\begin{pmatrix} \check{Y}_1 & \check{Y}_3 \\ \check{Y}_2 & \check{Y}_4 \end{pmatrix}.$$

Problem (4.1) is equivalent to

$$\min_{\hat{Y}_p\in\mathbb{R}^{2n\times 2p}} \tilde{F}_p(\hat{Y}_p) := F_p(\check{Y}_1 + \check{Y}_2\sqrt{-1}), \tag{4.3}$$
$$\text{such that } \operatorname{tr}(B_i^T\hat{Y}_p) = 0, i = 1, \ldots, 2np,$$

where B_k, $k = 1, \ldots, np$ are given by

$$B_k = \begin{pmatrix} 0_{n\times p} & E_{ij} \\ E_{ij} & 0_{n\times p} \end{pmatrix}, \quad i = 1, \ldots, n, \;\; j = i, \ldots, p$$

and the np remaining B_k, are given by

$$B_k = \begin{pmatrix} E_{ij} & 0_{n\times p} \\ 0_{n\times p} & -E_{ij} \end{pmatrix}, \quad i = 1, \ldots, n, \;\; j = 1, \ldots, p$$

where $E_{ij} \in \mathbb{R}^{n\times p}$ are the standard basis matrices. The B_k are not unique but this choice is simple and has useful properties exploited later.

The relationship between $\operatorname{grad}\tilde{F}_p(\hat{Y}_p)$ and $\operatorname{grad}\tilde{H}(\hat{Y}_p\hat{Y}_p^*)$ is easily obtained and is given in Lemma 1.

Lemma 1 *The gradients of the functions $\tilde{F}_p$ in (4.3) and $\tilde{H}$ in (4.2) satisfy*

$$\operatorname{grad}\tilde{F}_p(\hat{Y}_P) = 2\operatorname{grad}\tilde{H}(\hat{Y}_p\hat{Y}_p^T)\hat{Y}_p$$

A stationary point of $\tilde{F}_p$ is defined to satisfies the first order KKT conditions:

Definition 2 $\hat{Y}_p$ is a stationary point of (4.3) if there exists a vector $\lambda = (\lambda_1, \ldots, \lambda_{2np})^T \in \mathbb{R}^{2np}$ such that

$$\operatorname{tr}(B_i^T\hat{Y}_p) = 0 \text{ and } \operatorname{grad}\tilde{F}_p(\hat{Y}_P) - \sum_{i=1}^{2np}\lambda_i B_i = 2\operatorname{grad}\tilde{H}(\hat{Y}_p\hat{Y}_p^T)\hat{Y}_p - \sum_{i=1}^{2np}\lambda_i B_i = 0_{2n\times 2p}, \tag{4.4}$$

where $\operatorname{tr}(B_i^T\hat{Y}_p) = 0, i = 1, \ldots, 2np$.

Lemma 2 shows a necessary and sufficient condition for a stationary point of $\tilde{F}_p$. In other words, it specifies the λ_i in the Definition 2.

Lemma 2 *$\hat{Y} \in \mathbb{R}^{2n\times 2p}$ is a stationary point of $\tilde{F}_p$ if and only if*

$$\mathrm{tr}(B_i^T \hat{Y}_p) = 0 \text{ and } P_{\Delta} \operatorname{grad} \tilde{F}_p(\hat{Y}_p) = 2P_{\Delta}(\operatorname{grad} \tilde{H}(\hat{Y}_p \hat{Y}_p^T)\hat{Y}_p) = 0, \qquad (4.5)$$

where Δ denotes the feasible set of (4.3).

Proof First, suppose (4.5) holds. It can be seen that Δ is a linear space, i.e.,

$$\Delta = \{ \begin{pmatrix} \check{M}_1 & -\check{M}_2 \\ \check{M}_2 & \check{M}_1 \end{pmatrix} | \check{M}_1, \check{M}_2 \in \mathbb{R}^{n\times p} \}$$

and the perpendicular space of Δ is

$$\Delta_{\perp} = \{ \begin{pmatrix} -\check{M}_1 & \check{M}_2 \\ \check{M}_2 & \check{M}_1 \end{pmatrix} | \check{M}_1, \check{M}_2 \in \mathbb{R}^{n\times p} \}$$

It follows that $\operatorname{grad} \tilde{F}_p(\hat{Y}_p) = P_{\Delta} \operatorname{grad} \tilde{F}_p(\hat{Y}_p) + P_{\Delta_\perp} \operatorname{grad} \tilde{F}_p(\hat{Y}_p)$. By the definitions of B_k and $\Delta_\perp$, there exist $\{\lambda_i\}_{i=1}^{2np}$ such that $\sum_{i=1}^{2np} \lambda_i B_i = P_{\Delta_\perp} \operatorname{grad} \tilde{F}_p(\hat{Y}_p)$. Therefore, $\hat{Y}$ is a stationary point of $\tilde{F}_p$ by Definition 2.

If $\hat{Y}_p$ is a stationary point of $\tilde{F}_p$, it follows that $\operatorname{grad} \tilde{F}_p(\hat{Y}_p) \in P_{\Delta_\perp} \operatorname{grad} \tilde{F}_p(\hat{Y}_p)$, which yields (4.5).

The complex forms and their gradients, Hessians and stationary points can also be characterized and related to the real formulations above. The gradient and the action of Hessian of F_p are easily computed and are given in Lemma 3 in terms of H.

Lemma 3 *The gradient of F_p at Y_p is given by*

$$\operatorname{grad} F_p(Y_p) = 2 \operatorname{grad} H(Y_p Y_p^*) Y_p \qquad (4.6)$$

and the action of the Hessian of F_p at Y_p on $\eta_p \in \mathbb{C}^{n\times p}$ is given by

$$\operatorname{Hess} F_p(Y_p)[\eta_p] = 2 \operatorname{grad} H(Y_p Y_p^*)\eta_p + 2(\operatorname{Hess} H(Y_p Y_p^*)[\eta_p Y_p^* + Y_p \eta_p^*])Y_p. \qquad (4.7)$$

Proof On one hand, it satisfies that for all $\eta_p \in \mathbb{C}^{n\times p}$

$$\mathrm{D}\, F_p(Y_p)[\eta_p] = g^E(\operatorname{grad} F_p(Y_p), \eta_p).$$

On the other hand, we have

$$\begin{aligned}
\mathrm{D}\, F_p(Y_p)[\eta_p] =& \mathrm{D}\, H(Y_p Y_p^*)[Y_p \eta_p^* + \eta_p Y_p^*] = g^E(\operatorname{grad} H(Y_p Y_p^*), Y_p \eta_p^* + \eta_p Y_p^*) \\
=& \mathrm{Re}(\mathrm{tr}(\operatorname{grad} H(Y_p Y_p^*)^* Y_p \eta_p^*)) + \mathrm{Re}(\mathrm{tr}(\operatorname{grad} H(Y_p Y_p^*)^* \eta_p Y_p^*)) \\
=& \mathrm{Re}(\mathrm{tr}(\eta_p^* \operatorname{grad} H(Y_p Y_p^*)^* Y_p)) + \mathrm{Re}(\mathrm{tr}(\eta_p^* \operatorname{grad} H(Y_p Y_p^*) Y_p)) \\
=& \mathrm{Re}(\mathrm{tr}(\eta_p^* (\operatorname{grad} H(Y_p Y_p^*) + \operatorname{grad} H(Y_p Y_p^*)^*) Y_p)) \\
=& g^E((\operatorname{grad} H(Y_p Y_p^*) + \operatorname{grad} H(Y_p Y_p^*)^*) Y_p, \eta_p),
\end{aligned}$$

which implies $\operatorname{grad} F_p(Y_p) = (\operatorname{grad} H(Y_pY_p^*) + \operatorname{grad} H(Y_pY_p^*)^*)Y_p$. Since H is defined on Hermitian matrices, $\operatorname{grad} H$ can be written as a Hermitian matrix. It follows that $\operatorname{grad} F_p(Y_p) = 2\operatorname{grad} H(Y_pY_p^*)Y_p$ which is (4.7). The action of Hessian (4.7) can be computed in a straightforward way.

Definition 3 expresses a stationary point of H in terms of a stationary point of $\tilde{H}$, the real form of H.

Definition 3 Matrix $X = \check{X}_1 + \check{X}_2\sqrt{-1} \in \mathcal{D}_n$ is a stationary point of H if

$$\tilde{X} = \begin{pmatrix} \check{X}_1 & -\check{X}_2 \\ \check{X}_2 & \check{X}_1 \end{pmatrix}$$

is a stationary point of $\tilde{H}$.

4.3 Structure of $\mathcal{D}_k$

The discussion so far has all been relative to the Euclidean spaces of matrices with size and rank constraints. The structure of $\mathcal{D}_k$ and its relationship to Riemannian manifolds are crucial to the development of the optimality conditions and efficient algorithms. $\mathcal{D}_k$ is not a manifold but a union of manifolds, i.e.,

$$\mathcal{D}_k = \cup_{p=0}^{k} \mathrm{S}_+^{\mathbb{C}}(p, n),$$

where $\mathrm{S}_+^{\mathbb{C}}(p, n)$ can be shown to be a manifold over $\mathbb{R}$, i.e., it has a real parameterization. More specifically, the conjugate congruence orbit, $\mathrm{CCO}(A) = \{PAP^* | P \in \mathrm{GL}(n, \mathbb{C})\}$, is not a manifold over $\mathbb{C}$ but a manifold over $\mathbb{R}$ [DTD11] for any $A \in \mathbb{C}^{n\times n}$ where $\mathrm{GL}(n, \mathbb{C})$ denotes the complex general linear group. By choosing A to be

$$E = \begin{pmatrix} I_p & 0 \\ 0 & 0 \end{pmatrix},$$

it follows that $\mathrm{S}_+^{\mathbb{C}}(p, n) = \mathrm{CCO}(E)$. Therefore, $\mathrm{S}_+^{\mathbb{C}}(p, n)$ is a manifold over $\mathbb{R}$. Note that throughout this paper, a manifold always refers to a manifold over $\mathbb{R}$ rather than $\mathbb{C}$ even though the original optimization domain is in $\mathbb{C}^{n\times n}$.

We also point out that $\mathrm{S}_+^{\mathbb{R}}(p, n)$ is well known to be a smooth manifold and there exist many geometries for this manifold : a submanifold embedded in $\mathbb{R}^{n\times n}$ [HM94, HS95, OHM06, KL07, VAV09, VV10], a quotient manifold of $\mathbb{R}^{n\times p}$ [AIDV09, BMS10, JBAS10, MBS11], a quotient manifold of the Stiefel manifold and $\mathrm{S}_+^{\mathbb{R}}(p, p)$ [MJBS09, BMS10, BS10] and a quotient manifold of the generalize linear group [VAV12].

Lemma 4 gives the tangent cone of a point $X \in \mathcal{D}_k$. A similar result has been given for real low-rank matrices in [Cas12, Theorem 6.6] and the proof in Lemma 4 follows the spirit of the proof of that theorem.

Lemma 4 *If $X \in \mathcal{D}_k$ has rank $p \leq k$ and thin singular value decomposition $X = U_p D_p U_p^*$ then the tangent cone of $\mathcal{D}_k$ at X denoted by $\mathrm{T}_X\,\mathcal{D}_k$ is given by*

$$\begin{aligned}\mathrm{T}_X\,\mathcal{D}_k = \{ \begin{pmatrix} U_p & (U_p)_\perp \end{pmatrix} \begin{pmatrix} S & A^* \\ A & R \end{pmatrix} \begin{pmatrix} U_p & (U_p)_\perp \end{pmatrix}^* | S \in \mathbb{C}^{p\times p}, S = S^*, \\ A \in \mathbb{C}^{(n-p)\times p}, R = R^*, R \in \mathbb{C}^{(n-p)\times(n-p)}, \mathrm{rank}(R) = k-p. \}\end{aligned} \tag{4.8}$$

Proof For any element $\eta_X \in \mathrm{T}_X\,\mathcal{D}_k$, there exists a smooth curve $\gamma(t) \subset \mathcal{D}_k$, such that $\gamma(0) = X$ and $\dot\gamma(0) = \eta_X$. By [Kat80, p120-122], there exist smooth curves $U(t) \subset \mathcal{O}_n$, $D(t) \subset \mathbb{R}^{n\times n}$, and $D(t)$ is diagonal matrix such that $\gamma(t) = U(t)D(t)U(t)^*$. Therefore, η_X can be written as

$$\eta_X = \dot U(0)D(0)U(0)^* + U(0)\dot D(0)U(0)^* + U(0)D(0)\dot U(0)^*.$$

Since it is well known that the tangent space of the complex Stiefel manifold at U_p is $\mathrm{T}_{U_p}\,\mathrm{St}(p,n) = \{U_p\Omega_u + (U_p)_\perp K_u | \Omega_u^* = -\Omega_u, K_u \in \mathbb{C}^{(n-p)\times p}\}$, the first p columns of $\dot U(0)$ can be written in the form of $U_p\Omega + (U_p)_\perp K$, where $\Omega^* = -\Omega$, $\Omega \in \mathbb{C}^{p\times p}$, $K \in \mathbb{C}^{(n-p)\times p}$. It follows that

$$\begin{aligned}\eta_X = &(U_p\Omega + (U_p)_\perp K)D_pU_p^* \\ &+ (U_p\ \ (U_p)_\perp)\dot D(0)(U_p\ \ (U_p)_\perp)^* + U_pD_p(U_p\Omega + (U_p)_\perp K)^*.\end{aligned}$$

Let $\mathrm{Diag}(\dot D_p, \dot D_{n-p})$ denote $\dot D(0)$ where $\dot D_p \in \mathbb{R}^{p\times p}$ and $\dot D_{n-p} \in \mathbb{R}^{(n-p)\times(n-p)}$. It follows that

$$\begin{aligned}\eta_X =& U_p(\Omega D_p + \dot D_p + D_p\Omega^*)U_p^* \\ &+ (U_p)_\perp K D_p U_p^* + U_pD_pK^*(U_p)_\perp + (U_p)_\perp \dot D_{n-p}(U_p)_\perp^* \\ =& \begin{pmatrix} U_p & (U_p)_\perp \end{pmatrix} \begin{pmatrix} \Omega D_p + \dot D_p + D_p\Omega^* & D_pK^* \\ KD_p & \dot D_{n-p} \end{pmatrix} \begin{pmatrix} U_p & (U_p)_\perp \end{pmatrix}^*\end{aligned} \tag{4.9}$$

Since $\gamma(t) \in \mathcal{D}_k$, $\dot D_{n-p}$ has at most rank $k-p$. Using $\mathcal{S}$ to denote the right hand side of (4.8), (4.9) yields $\mathrm{T}_X\,\mathcal{D}_k \subseteq \mathcal{S}$.

For any $\xi_X \in \mathcal{S}$, by definition there exists S_1, A_1 and R_1 such that

$$\xi_X = \begin{pmatrix} U_p & (U_p)_\perp \end{pmatrix} \begin{pmatrix} S_1 & A_1^* \\ A_1 & R_1 \end{pmatrix} \begin{pmatrix} U_p & (U_p)_\perp \end{pmatrix}^*,$$

where $S_1 = S_1^*$ and R_1 is a rank $k-p$ matrix. It follows that

$$\begin{aligned}\xi_X = &\begin{pmatrix} U_p & (U_p)_\perp \end{pmatrix} \begin{pmatrix} 0 & -D_p^{-1}A_1^* \\ A_1D_p^{-1} & 0 \end{pmatrix} \begin{pmatrix} D_p & 0_{p\times(n-p)} \\ 0_{(n-p)\times p} & 0_{(n-p)\times(n-p)} \end{pmatrix} \begin{pmatrix} U_p & (U_p)_\perp \end{pmatrix}^* \\ &+ \begin{pmatrix} U_p & (U_p)_\perp \end{pmatrix} \mathrm{Diag}(S_1, R_1) \begin{pmatrix} U_p & (U_p)_\perp \end{pmatrix}^* \\ &+ \begin{pmatrix} U_p & (U_p)_\perp \end{pmatrix} \begin{pmatrix} D_p & 0_{p\times(n-p)} \\ 0_{(n-p)\times p} & 0_{(n-p)\times(n-p)} \end{pmatrix} \begin{pmatrix} 0 & D_p^{-1}A_1^* \\ -A_1D_p^{-1} & 0 \end{pmatrix} \begin{pmatrix} U_p & (U_p)_\perp \end{pmatrix}^*.\end{aligned}$$

Since

$$\begin{pmatrix} U_p & (U_p)_\perp \end{pmatrix} \begin{pmatrix} 0 & -D_p^{-1}A_1^* \\ A_1 D_p^{-1} & 0 \end{pmatrix} \in \mathrm{T}_{(U_p \;\; (U_p)_\perp)}\, \mathcal{O}_n,$$

there exist a smooth curve $r_1(t)$ such that $r_1(0) = (U_p \;\; (U_p)_\perp)$ and

$$\dot{r}_1(0) = \begin{pmatrix} U_p & (U_p)_\perp \end{pmatrix} \begin{pmatrix} 0 & -D_p^{-1}A_1^* \\ A_1 D_p^{-1} & 0 \end{pmatrix}$$

Define $D_1(t)$ to be $\mathrm{Diag}(D_p + tS_1, tR_1)$ which is a smooth curve over $\mathbb{C}^{n\times n}$. Therefore, $\gamma_1(t) = r_1(t)D_1(t)r_1(t)^*$ is a smooth curve in $\mathcal{D}_k$ and satisfies $\dot{\gamma}_1(0) = \xi_X$, which implies $\xi_X \in \mathrm{T}_X\, \mathcal{D}$. Thus, $\mathcal{S} \subseteq \mathrm{T}_X\, \mathcal{D}_k$ holds.

When $k = p$, $\mathrm{T}_X\, \mathcal{D}_k$ is the tangent space of $\mathrm{S}_+^{\mathbb{C}}(p, n)$ at X, i.e.,

$$\begin{aligned} \mathrm{T}_X\, \mathrm{S}_+^{\mathbb{C}}(p,n) = \{ \begin{pmatrix} U_p & (U_p)_\perp \end{pmatrix} \begin{pmatrix} S & A^* \\ A & 0_{(n-p)\times(n-p)} \end{pmatrix} \begin{pmatrix} U_p & (U_p)_\perp \end{pmatrix}^* | A \in \mathbb{C}^{p\times(n-p)}, \\ S \in \mathbb{C}^{p\times p}, S = S^*. \} \end{aligned}$$

Finally, the tangent cone of a low-rank matrix X can be written as the summation of directions in $\mathrm{T}_X\, \mathrm{S}_+^{\mathbb{C}}(p, n)$ and directions pointing to a higher rank space, i.e.,

$$\mathrm{T}_X\, \mathcal{D}_k = \mathrm{T}_X\, \mathrm{S}_+^{\mathbb{C}}(p,n) + (U_p)_\perp R (U_p)_\perp^*. \tag{4.10}$$

This is the form that is convenient computationally and used in the implementation of the algorithms assessed below.

4.4 Euclidean Optimality Conditions

In this section, the characterizations of stationary points of F and H over $\mathcal{D}_n$, i.e., on a feasible subset of a Euclidean space, derived in Section 4.2 are used to derive the relationship between optimizing F and optimizing H over $\mathcal{D}_n$.

Theorem 1 develops a condition related to H satisfied by stationary points of F that follows from (4.6) in Lemma 3.

Theorem 1 *Y_p is a stationary point of F_p if and only if* $\mathrm{grad}\, H(Y_pY_p^*)Y_p = 0_{n\times p}$.

Let s denote the rank of Y_p and X denote $Y_pY_p^*$. From $\mathrm{grad}\, H(X) \in \mathrm{T}_X\, \mathcal{D}_p$ and (4.10), we have $\mathrm{grad}\, H(X)Y_p = 0_{n\times p}$ is equivalent to $P_{\mathrm{T}_X\, \mathrm{S}_+^{\mathbb{C}}(p,n)}(\mathrm{grad}\, H(X)) = 0_{n\times n}$. In other words, Theorem 1 shows that Y_p is a stationary point of F_p if and only if the component of $\mathrm{grad}\, H(X)$ in $\mathrm{T}_X\, \mathrm{S}_+^{\mathbb{C}}(s, n)$ is zero. Note that if it is only known that Y_p is a stationary point of F_p then no information concerning $\mathrm{grad}\, H(X)$ pointing to a higher rank space is given.

Since the condition $\mathrm{grad}\, H(X)Y_p = 0_{n\times p}$ is equivalent to the condition $\mathrm{grad}\, \tilde{H}(\tilde{Y}_p\tilde{Y}_p^T)\tilde{Y}_p = 0_{2n\times 2p}$, Theorem 1 can also be obtained from the real formulation of the cost function, i.e., Lemma 2, however, the information above

characterizing the stationary point of F_p in terms of the absence of a certain component of $\operatorname{grad} H(X)$ is more easily seen from the complex form.

Lemma 5 provides sufficient conditions for X to be a stationary point of H.

Lemma 5

1. *If* $\operatorname{grad} H(X)X = 0$ *and* $\operatorname{grad} H(X) \geq 0$, *then* $X \in \mathcal{D}_n$ *is a stationary point of* H.
2. *If* Y_p *is a stationary point of* F_p *and* $\operatorname{grad} H(Y_pY_p^*) \geq 0$, *then* $X = Y_pY_p^*$ *is a stationary point of* H.

Proof The conditions $\operatorname{grad} H(X)X = 0$ and $\operatorname{grad} H(X) \geq 0$ imply $\operatorname{grad} \tilde{H}(\tilde{X}) \geq 0$ and $\operatorname{grad} \tilde{H}(\tilde{X})\tilde{X} = 0$. Therefore, choosing $S = \operatorname{grad} \tilde{H}(\tilde{X})$ and $\delta = 0_m$ in Definition 1 yields the first statement. The second statement follows from Theorem 1 and the first statement.

Theorem 2 and [JBAS10, Theorem 7] show similar results under different frameworks. Both results suggest considering the cost function F_p if the desired minimizer of H is known to have rank smaller than p, as is the case with PhaseLift for phase retrieval. This is formalized in Theorem 2 and has critical algorithmic, efficiency and optimality implications when H has suitable structure such as convexity as in the case of PhaseLift. These implications for PhaseLIft are discussed in Section 6.1.

Theorem 2 *Suppose* $Y_p = K_sQ^*$ *is a rank deficient minimizer of* F_p, *where* $K_s \in \mathcal{C}_*^{n\times s}$ *and* $Q \in \operatorname{St}(s,p)$. *Then* $(K_s)_\perp^* \operatorname{grad} H(Y_pY_p^*)(K_s)_\perp$ *is a positive semidefinite matrix and, therefore,* $X = Y_pY_p^*$ *is a stationary point of* H.

Proof This is proved by contradiction. If $(K_s)_\perp^* \operatorname{grad} H(X)(K_s)_\perp$ is not a positive semidefinite matrix, then it has at least one negative eigenvalue. If μ and v denote a negative eigenvalue and the corresponding eigenvector then the semidefinite positive matrix $\eta = -(K_s)_\perp(v\mu v^*)(K_s)_\perp^*$ satisfies $g^E(\eta, \operatorname{grad} H(X)) < 0$. Thus, a smooth curve $\gamma(t)$, e.g., $\gamma(t) = X + t\eta$, can be chosen such that $\dot{\gamma}(0) = \eta$, $\gamma(t) \in \mathcal{D}_p$ for all $t \in [0, \delta)$ and $\gamma(0) = X$, where δ is a positive constant. The derivative $\frac{d}{dt}H(\gamma(t))|_{t=0}$ by definition is $g^E(\eta, \operatorname{grad} H(X))$ and, therefore, $\frac{d}{dt}H(\gamma(t))|_{t=0} < 0$.

Since a smooth eigenvalue value decomposition exists for any smooth curve in $\mathbb{C}^{n\times n}$ [Kat80, p120-122], there exists smooth curves $r(t) \subset \operatorname{St}(p,n)$ and $d(t) = (d_1(t), d_2(t), \ldots, d_p(t))^T \subset \mathbb{R}^p$ such that $\gamma(t) = r(t)\operatorname{Diag}(d(t))r(t)^*$. The derivative of $r(t)\operatorname{Diag}(\sqrt{d(t)})$ is

$$\eta(t) = \dot{r}(t)\operatorname{Diag}(\sqrt{d(t)}) + r(t)\operatorname{Diag}(\frac{\dot{d}_1(t)}{2\sqrt{d_1(t)}}, \ldots, \frac{\dot{d}_p(t)}{2\sqrt{d_p(t)}})$$

and the derivative of $r(t^2)\operatorname{Diag}(\sqrt{d(t^2)})$ is

$$\xi(t) = \dot{r}(t^2)2t\operatorname{Diag}(\sqrt{d(t^2)}) + r(t^2)\operatorname{Diag}(\dot{d}_1(t^2)\sqrt{\frac{t^2}{d_1(t^2)}}, \ldots, \dot{d}_p(t^2)\sqrt{\frac{t^2}{d_p(t^2)}}).$$

Two cases are considered. The first case assumes that for any i, the conditions $d_i(0) = 0$, $\dot{d}_i(0) = 0$ hold and the second case, which is true if the first case is not, assumes that there exists an i such that $d_i(0) = 0$ but $\dot{d}_i(0) \neq 0$.

In the first case, $\eta(0)$ is well-defined and it follows that

$$\frac{d}{dt}H(\gamma(t))|_{t=0} = \frac{d}{dt}F_p(r(t)\operatorname{Diag}(\sqrt{d(t)}))|_{t=0} = g^E(\eta(0), \operatorname{grad} F_p(Y_p)) = 0,$$

which contradicts that $\frac{d}{dt}H(\gamma(t))|_{t=0} < 0$.

In the second case, $\eta(0)$ is undefined but $\xi(0)$ is well-defined. It follows that

$$\begin{aligned}\frac{d^2}{dt^2}H(\gamma(t^2))|_{t=0} &= \frac{d^2}{dt^2}F_n(r(t^2)\operatorname{Diag}(\sqrt{d(t^2)}))_{t=0} \\ &= g^E(\xi(0), \operatorname{Hess} F_p(\tilde{Y}_p)[\xi(0)]) \geq 0. \end{aligned} \tag{4.11}$$

Let $a(t) = H(\gamma(t))$ and so $\dot{a}(0) < 0$. It follows that

$$\frac{d^2}{dt^2}H(\gamma(t^2))|_{t=0} = \frac{d^2}{dt^2}a(t^2) = (4t^2\ddot{a}(t^2) + 2\dot{a}(t^2))|_{t=0} = 2\dot{a}(0) < 0,$$

which conflicts with (4.11). Therefore, $(K_s)_\perp^* \operatorname{grad} H(Y_pY_p^*)(K_s)_\perp$ is a positive semidefinite matrix which is a contradiction with the initial assumption of the proof.

Let Q_s denote an orthonormal basis of $\operatorname{span}(K_s)$. $\operatorname{grad} H(X)$ can be written as

$$\operatorname{grad} H(X) = \begin{pmatrix} Q_s & (K_s)_\perp \end{pmatrix} \begin{pmatrix} S & A^* \\ A & R \end{pmatrix} \begin{pmatrix} Q_s & (K_s)_\perp \end{pmatrix}^*,$$

where $S \in \mathbb{C}^{s\times s}$, $S^* = S$, $A \in \mathbb{C}^{(n-s)\times p}$ and $R = (K_s)_\perp^* \operatorname{grad} H(X)(K_s)_\perp$. Theorem 1 implies $\operatorname{grad} H(X)K_s = 0_{n\times s}$. It follows that $S = 0_{s\times s}$ and $A = 0_{(n-s)\times s}$. Therefore, $R \geq 0$ implies $\operatorname{grad} H(X) \geq 0$ which means X is a stationary point of H by Lemma 5.

$(K_s)_\perp((K_s)_\perp^* \operatorname{grad} H(X)(K_s)_\perp)(K_s)_\perp^*$ is the component of $\operatorname{grad} H(X)$ that points to a higher rank space, i.e., $\operatorname{grad} H(X) - P_{\mathrm{T}_X \mathrm{S}_+^{\mathbb{C}}(s,n)}(\operatorname{grad} H(X)) = (K_s)_\perp((K_s)_\perp^* \operatorname{grad} H(X)(K_s)_\perp)(K_s)_\perp^*$. Therefore, the positive semidefiniteness of $(K_p)_\perp^* \operatorname{grad} H(X)(K_p)_\perp$ means that the descent direction, $-\operatorname{grad} H(X)$, does not have a component pointing to the higher rank space and stays in the positive semidefinite domain $\mathcal{D}_n$.

Theorem 2 can be obtained also by the approach in the proof of [JBAS10, Theorem 7] that exploits the second-order KKT condition for the low-rank factorization alternate cost function $\tilde{F}_p$.

5 A Riemannian Approach

5.1 Riemannian Optimization Preliminaries

Riemannian optimization is an active research area and recently many Riemannian optimization methods have been systemically analyzed and efficient libraries designed, e.g., Riemannian trust-region Newton method (RTR-Newton)

[Bak08], Riemannian Broyden family method including BFGS method and its limited-memory version (RBroyden family, RBFGS, LRBFGS) [RW12, Hua13, HGA14], Riemannian trust-region symmetric rank-one update method and its limited-memory version (RTR-SR1, LRTR-SR1) [Hua13, HAG14], Riemannian Newton method (RNewton) and Riemannian non-linear conjugate gradient method (RCG) [AMS08].

Journée et al. [JBAS10] have proposed a method that combines a Riemannian optimization method on a fixed rank manifold with a procedure of increasing rank for their semidefinite constrained problem setting. Specifically, given an iterate with rank r, a Riemannian optimization method is applied for a cost function on a manifold with rank r. If the limit point is not rank deficient, then either a descent direction to a higher rank space can be found or a desired stationary point is obtained. For the former case, a descent algorithm is applied to find a next descent iterate which is used to be the initial point for a Riemannian optimization method on a manifold with larger rank. For the latter case, the convergence rate can be obtained and depends on the Riemannian optimization algorithm. If the limit point is rank deficient, then all existing Riemannian convergence analyses are not applicable. This case was ignored in [JBAS10] since situations of a limit point being rank deficient were not encountered in their experiments.

If the rank of the desired minimizer is known, such as in the problems in PhaseLift, then both [BM03] and [JBAS10] suggest to choose the rank of initial point to be that rank. However, this, in fact, is not the appropriate response due to complexity considerations as the theory and algorithms derived in this section indicate, and the numerical experiments in Section 6 demonstrate. To see this, let r_* denote the desired rank of the global minimizer. As shown in Theorem 1, there may exist a stationary point Y_{r_*} of F_{r_*} for which $Y_{r_*}Y_{r_*}^*$ is not a stationary point of H. It follows that forcing iterates to be rank r_* is not appropriate and starting from a higher rank or moving to a higher rank to move to the minimizer of H is necessary. This is discussed further later in this section.

There is also a potential problem of using a rank increasing procedure. If Y_p is a stationary point of F_p, then $(Y_p, 0_{n\times(k-p)})$ is also a stationary point of F_k by Theorem 1. A procedure that increases rank starting from Y_p may find a point Y_k which is close to $(Y_p, 0_{n\times(k-p)})$. It follows that using Y_k to be an initial point of the iteration on the rank-k manifold may not work efficiently since Y_k may be too close to a stationary point. Therefore, an algorithm based on Riemannian optimization methods on a fixed rank manifold and a procedure to decrease rank is proposed in this section. Since it is known that the minimizer for PhaseLift phase retrieval has rank 1 this is sufficient to allow global minimization using only rank decreases. For completeness, a simple procedure to increase rank under the complex Hermitian semidefinite matrix framework is included.

5.2 Riemannian Optimization on Fixed Rank Manifold

Derivations for Riemannian objects of $\mathrm{S}_+^{\mathbb{R}}(p,n)$ have been given in [AIDV09]. This section includes derivations of Riemannian objects for the complex case, i.e., $\mathrm{S}_+^{\mathbb{C}}(p,n)$. Since the mapping α_p is not an injection, all the minimizers of F_p are degenerate, which causes difficulties in some algorithms, e.g., Riemannian and Euclidean Newton method. In order to overcome this difficulty, a function defined on a quotient manifold with fixed rank is considered. To this end, define the mapping β_p to be the mapping α_p restricted on $\mathbb{C}_*^{n\times p}$, i.e.,

$$\beta_p : \mathbb{C}_*^{n\times p} \to \mathrm{S}_+^{\mathbb{C}}(p,n) : Y \mapsto \alpha_p(Y) = YY^*.$$

and function G_p to be the function F_p restricted on $\mathbb{C}_*^{n\times p}$, i.e.,

$$G_p : \mathbb{C}_*^{n\times p} \to \mathbb{R} : Y \mapsto F_p(Y) = H(\beta_p(Y)).$$

Like α_p, the mapping β_p is a surjection but not a injection and there are multiple matrices in $\mathbb{C}_*^{n\times p}$ mapping to a single point in $\mathrm{S}_+^{\mathbb{C}}(p,n)$. Nevertheless, given a $X \in \mathrm{S}_+^{\mathbb{C}}(p,n)$, $\beta_p^{-1}(X)$ is a manifold while $\alpha_p^{-1}(X)$ is not a manifold. Therefore, using the mapping β_p, a quotient manifold can be used to remove the degeneracy by defining the equivalence class $\beta_p^{-1}(YY^*)$

$$[Y] = \{YO | O \in \mathcal{O}_p\}.$$

and the set

$$\mathbb{C}_*^{n\times p}/\mathcal{O}_p = \{[Y] | Y \in \mathbb{C}_*^{n\times p}\}.$$

This set can be shown to be a quotient manifold over $\mathbb{R}$ and the proof can be found in Appendix A. To clarify the notation, $\pi(Y)$ is used to denote $[Y]$ viewed as an element in $\mathbb{C}_*^{n\times p}/\mathcal{O}_p$ and $\pi^{-1}(\pi(Y))$ is used to denote $[Y]$ viewed as a subset of $\mathbb{C}_*^{n\times p}$. The function $m_p : \pi(Y) \mapsto YY^*$ is a diffeomorphism between $\mathbb{C}_*^{n\times p}/\mathcal{O}_p$ and $\mathrm{S}_+^{\mathbb{C}}(p,n)$.

An element of a quotient manifold is an equivalence class which is often cumbersome computationally . Fortunately, choosing a representative for an equivalence class and definitions of related mathematical objects have been developed in many papers in the literature of computation on manifolds, e.g., [AMS08]. The vertical space at $Y \in \pi^{-1}(\pi(Y))$, which is the tangent space of $\pi^{-1}(\pi(Y))$ at Y, is

$$\mathcal{V}_Y = \{Y\Omega | \Omega^* = -\Omega, \Omega \in \mathbb{C}^{p\times p}\}.$$

The horizontal space at Y, $\mathcal{H}_Y$, is defined to be a subspace of $\mathrm{T}_Y\,\mathbb{C}_*^{n\times p} = \mathbb{C}^{n\times p}$ that is orthogonal to $\mathcal{V}_Y$, i.e., satisfying $\mathcal{H}_A \oplus \mathcal{V}_A = \mathrm{T}_A\,\mathrm{GL}(n,\mathbb{C})$. Therefore, a Riemannian metric of $\mathbb{C}_*^{n\times p}$ is required to define the meaning of orthogonal. The standard Euclidean metric is

$$\hat{g}_Y(\eta_Y, \xi_Y) = \mathrm{Re}(\mathrm{tr}(\eta_Y^* \xi_Y)), \tag{5.1}$$

for all $\eta_Y, \xi_Y \in \mathrm{T}_Y\,\mathbb{C}_*^{n\times p}$ and $Y \in \mathbb{C}_*^{n\times p}$. The horizontal space is therefore

$$\begin{aligned}\mathcal{H}_Y =&\{V \in \mathbb{C}^{n\times p} | Y^*V = V^*Y\}\\ =&\{Y(Y^*Y)^{-1}S + Y_\perp K | S^* = S, S \in \mathbb{C}^{p\times p}, K \in \mathbb{C}^{(n-p)\times p}\}.\end{aligned}$$

The horizontal space $\mathcal{H}_Y$ is a representation of the tangent space $\mathrm{T}_{\pi(Y)}\,\mathbb{C}_*^{n\times p}/\mathcal{O}_p$. It is known that for any $\eta_{\pi(Y)} \in \mathrm{T}_{\pi(Y)}\,\mathbb{C}_*^{n\times p}/\mathcal{O}_p$, there exists a unique vector in $\mathcal{H}_Y$, called the horizontal lift of $\eta_{\pi(Y)}$ and denoted by $\eta_{\uparrow_Y}$, satisfying

$$\mathrm{D}\,\pi(Y)[\eta_{\uparrow_Y}] = \eta_{\pi(Y)},$$

see e.g., [AMS08]. Lemma 6 gives a relationship among horizontal lifts of a tangent vector $\eta_{\pi(Y)}$ when different representations in $\pi^{-1}(\pi(Y))$ are chosen. The result follows from [Hua13, Theorem 9.3.1].

Lemma 6 *A horizontal vector field $\hat{\eta}$ of $\mathbb{C}_*^{n\times p}$ is the horizontal lift of a vector field η on $\mathbb{C}_*^{n\times p}/\mathcal{O}_p$ if and only if, for each $Y \in \mathbb{C}_*^{n\times p}$, we have*

$$\hat{\eta}_{YO} = \hat{\eta}_Y O \text{ for all } O \in \mathcal{O}_p.$$

The orthogonal projections on to the horizontal space or the vertical space are also easily characterized.

Lemma 7 *The orthogonal projection to vertical space $\mathcal{V}_Y$ is*

$$P_Y^v(\eta) = Y\Omega,$$

where Ω is the skew symmetric matrix that solves the Sylvester equation,

$$\Omega Y^*Y + Y^*Y\Omega = Y^*\eta - \eta^*Y.$$

The orthogonal projection to Horizontal space $\mathcal{H}_Y$ is

$$P_Y^h(\eta) = \eta - Y\Omega.$$

Proof By definition of $\mathcal{H}_Y$ and $\mathcal{V}_Y$, $P_Y^h(\eta)$ satisfies that $Y^*P_Y^h(\eta) = P_Y^h(\eta)^*Y$ and can be expressed as $\eta - Y\Omega$. It follows that

$$\Omega Y^*Y + Y^*Y\Omega = Y^*\eta - \eta^*Y$$

which gives the desired results.

The metric $\hat{g}$ of $\mathbb{C}_*^{n\times p}$ defines a metric g of $\mathbb{C}_*^{n\times p}/\mathcal{O}_p$.

Lemma 8 *The mapping g defined by*

$$g_{\pi(Y)}(\eta_{\pi(Y)}, \xi_{\pi(Y)}) = \hat{g}_Y(\eta_{\uparrow_Y}, \xi_{\uparrow_Y}) = \mathrm{Re}(\mathrm{tr}(\eta_{\uparrow_Y}^*\xi_{\uparrow_Y})) \tag{5.2}$$

is a Riemannian metric on $\mathbb{C}_*^{n\times p}/\mathcal{O}_p$.

Proof The lifted metric is invariant on the chosen representation for $\pi(Y)$

$$\begin{aligned}\hat{g}_{YO}(\eta_{\uparrow_{YO}}, \xi_{\uparrow_{YO}}) &= \operatorname{Re}(\operatorname{tr}(\eta^*_{\uparrow_{YO}}\xi_{\uparrow_{YO}})) = \operatorname{Re}(\operatorname{tr}((\eta_{\uparrow_Y}O)^*(\xi_{\uparrow_Y}O))) \\ &= \operatorname{Re}(\operatorname{tr}(\eta^*_{\uparrow_Y}\xi_{\uparrow_Y})) = \hat{g}_Y(\eta_{\uparrow_Y}, \xi_{\uparrow_Y}).\end{aligned}$$

Finally, the desired cost function that removes the equivalence can be defined as

$$f_p : \mathbb{C}_*^{n\times p}/\mathcal{O}_p \to \mathbb{R} : \pi(Y) \mapsto f_p(\pi(Y)) = G_p(Y) = F_p(Y). \tag{5.3}$$

The function f_p in 5.3 has the important property that $\pi(Y)$ is a nondegenerate minimizer of f over $\mathbb{C}_*^{n\times p}/\mathcal{O}_p$ if and only if YY^* is a nondegenerate minimizer of H over $\mathrm{S}_+^{\mathbb{C}}(p, n)$.

The gradient and the action of the Hessian of (5.3) are given in Lemma 9.

Lemma 9 *The horizontal lift of the gradient of* (5.3) *at* Y *is*

$$(\operatorname{grad} f(\pi(Y)))_{\uparrow_Y} = P^h_Y(\operatorname{grad} F(Y)),$$

and the action of Hessian of (5.3) *at* $\pi(Y)$ *along* $\eta_{\pi(Y)} \in \mathrm{T}_{\pi(Y)}\,\mathbb{C}_*^{n\times p}/\mathcal{O}_p$ *satisfies*

$$(\operatorname{Hess} f(\pi(Y))[\eta_{\pi(Y)}])_{\uparrow_Y} = P^h_Y(\dot{M} - \eta_{\uparrow_Y}\Omega),$$

where $\dot{M} = \operatorname{Hess} F(Y)[\eta_{\uparrow_Y}]$, Ω *is the skew-symmetric matrix that solves*

$$\Omega Y^*Y + Y^*Y\Omega = Y^*M - M^*Y,$$

and $M = \operatorname{grad} F(Y)$.

Proof The directional derivative of f along any $\eta_{\pi(Y)} \in \mathrm{T}_{\pi(Y)}\,\mathbb{C}_*^{n\times p}/\mathcal{O}_p$ is

$$\begin{aligned}&\mathrm{D}\, f(\pi(Y))[\eta_{\pi(Y)}] = \mathrm{D}\, f(\pi(Y))[\mathrm{D}\,\pi(Y)[\eta_{\uparrow_Y}]] \\ &= \mathrm{D}\, F(Y)[\eta_{\uparrow_Y}] = \hat{g}_Y(\operatorname{grad} F(Y), \eta_{\uparrow_Y}) = \hat{g}_Y(P^h_Y(\operatorname{grad} F(Y)), \eta_{\uparrow_Y}).\end{aligned}$$

Additionally using the definition of gradient [AMS08, (3.31)], i.e.,

$$\mathrm{D}\, f(\pi(Y))[\eta_{\pi(Y)}] = g_{\pi(Y)}(\operatorname{grad} f(\pi(Y)), \eta_{\pi(Y)}),$$

and the equation

$$g_{\pi(Y)}(\operatorname{grad} f(\pi(Y)), \eta_{\pi(Y)}) = \hat{g}_Y((\operatorname{grad} f(\pi(Y)))_{\uparrow_Y}, \eta_{\uparrow_Y}),$$

yields the result. The action of the Hessian is computed in a straightforward way.

The retraction used in the Riemannian optimization methods is

$$R_{\pi(Y)}(\eta_{\pi(Y)}) = \pi(Y + \eta_{\uparrow_Y}). \tag{5.4}$$

Two vector transports are used for different Riemannian algorithms. One is the vector transport by parallelization [HAG14, Section 2.3.1] and the other one is the vector transport by differentiated retraction (5.4), which also is the vector transport by projection:

$$\left(\mathcal{T}_{\eta_{\pi(Y)}}(\xi_{\pi(Y)})\right)_{\uparrow_{\left(Y+\eta_{\uparrow_Y}\right)}} = P^h_{\left(Y+\eta_{\uparrow_Y}\right)}\xi_{\uparrow_Y}. \tag{5.5}$$

In summary, this section provides all the objects used in Riemannian optimization methods, i.e., the horizontal space, the projection to a horizontal space, the Riemannian metric, the retraction, the vector transport, the Riemannian gradient and action of the Riemannian Hessian.

5.3 Dynamic Rank Reduction

Since the domain of f_p, $\mathbb{C}_*^{n\times p}/\mathcal{O}_p$, is not closed, i.e., a sequence $\{W^{(i)}\}$ representing $\{\pi(W^{(i)})\}$ generated by an algorithm may have a limit point $\hat{W}$ with rank less than p, a simple well-known strategy for dynamically reducing rank is adapted and used. Since it is impossible in practice to check whether a limit point of iterates $\{W^{(i)}\}$ is a lower rank matrix or just close to one of lower rank, the idea suggested below makes more sense when the desired rank of the minimizer is known and the current iterate $W^{(i)}$ has a higher rank than the desired rank. This is the case with PhaseLift for phase retrieval.

The thin singular value decomposition of the i-th iterate is $W^{(i)} = U\Sigma V^*$ and $\Sigma = \operatorname{Diag}(\sigma_1, \sigma_2, \ldots, \sigma_p)$, where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$. Let $\tilde{\sigma}$ be $\|\operatorname{Diag}(\sigma_1, \ldots, \sigma_p)\|_F/\sqrt{p}$. If there exists $q < p$ such that $\sigma_q/\tilde{\sigma} > \delta$ and $\sigma_{q+1}/\tilde{\sigma} \leq \delta$ for a given threshold δ, then $\hat{W} = U(:, 1:q)\operatorname{Diag}(\sigma_1, \ldots \sigma_q)V(:, 1:q)^*$ is chosen to be the initial point for optimizing cost function f_q over $\mathbb{C}_*^{n\times q}/\mathcal{O}_q$. The details of reducing rank are given in Algorithm 1. Note that the step of decreasing the rank may produce an iterate that increases the cost function value. This facilitates global optimization by allowing nondescent steps.

Combining a Riemannian optimization method with the procedure of reducing rank gives Algorithm 2.

5.4 Dynamic Rank Increase

While dynamic rank increase is not required for PhaseLift phase retrieval, it is included for completeness in the discussion of optimization over the Hermitian positive semidefinite matrices with a maximum rank constraint. Suppose the limit point of $\{W^{(i)}\}$, $\hat{W}$, is full rank, i.e., rank is p. It is still possible to move

Algorithm 1 Reduce Rank

Require: $Y \in \mathbb{C}^{n\times p}$; threshold δ;
Ensure: $W \in \mathbb{C}^{n\times q}$;
1: Take thin singular value decomposition for Y, i.e., $Y = U\operatorname{Diag}(\sigma_1,\dots,\sigma_p)V^*$, where $U \in \mathbb{C}^{n\times p}$, $V \in \mathbb{C}^{p\times p}$ and $\sigma_1 \geq \dots \geq \sigma_p \geq 0$;
2: Set $\tilde{\sigma} = \|\operatorname{Diag}(\sigma_1,\dots,\sigma_p)\|_F/\sqrt{p}$;
3: **if** $\sigma_p/\tilde{\sigma} > \delta$ **then**
4: $q \leftarrow p$, $W \leftarrow Y$ and return;
5: **else**
6: Find q such that $\sigma_q/\tilde{\sigma} > \delta$ and $\sigma_{q+1}/\tilde{\sigma} \leq \delta$;
7: Let $W = U(:,1:q)\operatorname{Diag}(\sigma_1,\dots\sigma_q)V(:,1:q)^*$ and return;
8: **end if**

Algorithm 2 Rank Reduce Algorithm

Require: $p > 0$; $Y_p^{(0)} \in \mathbb{C}^{n\times p}$ a representation of initial point $\pi(Y_p^{(0)})$ for f; Stopping criterion threshold ϵ; rank reducing threshold δ; a Riemannian optimization method;
Ensure: W
1: **for** $k = 0, 1, 2, \dots$ **do**
2: Apply Riemannian method for cost function f over $\mathbb{C}_*^{n\times p}/\mathcal{O}_p$ with initial point $\pi(Y_p^{(k)})$ until i-th iterate $W^{(i)}$ satisfying $g(\operatorname{grad} f, \operatorname{grad} f) < \epsilon^2$ or the requirement of reducing rank with threshold δ;
3: **if** $g(\operatorname{grad} f, \operatorname{grad} f) < \epsilon_1^2$ **then**
4: Find a minimizer $W = W^{(i)}$ over $\mathbb{C}_*^{n\times p}/\mathcal{O}_p$ and return;
5: **else** {iterate in the Riemannian optimization method meets the requirements of reducing rank}
6: Apply Algorithm 1 with threshold δ and obtain an output $\hat{W} \in \mathbb{C}^{n\times q}$;
7: $p \leftarrow q$ and set $Y_p^{(k+1)} = \hat{W}$;
8: **end if**
9: **end for**

from $\hat{W}$ along a descent direction that increases the rank. Since $(\hat{W} \;\; 0_{n\times(k-p)})$ is always a stationary point of F_k, the gradient of F_k is not enough to provide a direction that reduces the cost function H and increases the rank. Instead, another equivalent cost function can be used

$$\hat{f}_k : \operatorname{St}(k,n) \times \mathbb{R}^k \to \mathbb{R} : (U_k, d_k) \mapsto H(U_k \operatorname{Diag}(d_k) U_k^*). \tag{5.6}$$

The reason is that given a rank p matrix $X \in \mathcal{D}_k$ and a direction V in the tangent cone at X, there may not exist a direction η_k for point $Y_k = (K_p \;\; 0_{n\times(k-p)}) \in \mathbb{C}^{n\times k}, Y_kY_k^* = X$ such that $\eta_kY_k^* + Y_k\eta_k^* = V$. This can be seen from the fact that V can be any vector in $\operatorname{T}_X\mathcal{D}_k$ but the image of $\eta_kY_k^* + Y_k\eta_k^*$ is $\operatorname{T}_{K_p}\operatorname{S}_+^{\mathbb{C}}(p,n)$ which is only a subset of $\operatorname{T}_X\mathcal{D}_k$. However, by the proof in Lemma 4, there exists $\dot{U}_k$ and $\dot{d}$ such that $\dot{U}_k\operatorname{Diag}(d_k)U_k^* + U_k\operatorname{Diag}(\dot{d}_k)U_k^* + U_k\operatorname{Diag}(d_k)\dot{U}_k^* = V$ for any $V \in \operatorname{T}_X\mathcal{D}_k$.

A simple strategy of increasing rank by 1 at a time is used, i.e., $k = p+1$, can be used. Let $\hat{U}\operatorname{Diag}(\hat{\sigma}_1,\dots,\hat{\sigma}_p)\hat{U}^*$ be thin singular value decomposition of $\hat{W}$. If there exists a vector $\hat{u} \in \mathbb{C}^n$, $\hat{u}^*\hat{u} = 1, \hat{u}^*\hat{U} = 0$ such that the negative gradient of $\hat{f}$ at $((\hat{U} \;\; \hat{u}), (\sigma_1,\dots,\sigma_p,0)^T) \in \operatorname{St}(k,n)\times\mathbb{R}^k$, denoted by $(\eta, v) \in \operatorname{T}_{\hat{U}}\operatorname{St}(k,n)\times\mathbb{R}^k$, satisfies that the k-th component of v

is positive, then applying one step of any algorithm for $\hat{f}$ with initial point $((\hat{U} \quad \hat{u}), (\sigma_1, \ldots, \sigma_p, 0)^T)$ is able to obtain a new point (P, ℓ) such that all entries in ℓ are all positive. It follows that a higher rank iterate that keeps function decreasing is obtained. An algorithm for optimizing f_k with initial point $\pi(P\sqrt{\ell})$ over the fixed rank quotient $\mathbb{C}_*^{n\times k}/\mathcal{O}_k$ can be applied.

To determine the new point, a Riemannian gradient on $\mathrm{St}(k,n) \times \mathbb{R}^k$ is needed. The metric of the product of manifolds $\mathrm{St}(k,n) \times \mathbb{R}^k$ is

$$g^P_{(U,d)}((\xi, v), (\hat{\xi}, \hat{v})) = \mathrm{Re}(\mathrm{tr}(\xi^* \hat{\xi})) + v^T \hat{v}, \tag{5.7}$$

where $\xi, \hat{\xi} \in \mathrm{T}_U \,\mathrm{St}(k,n)$ and $v, \hat{v} \in \mathbb{R}^k$. The Riemannian gradient of cost function $\hat{f}$ in (5.6) is given in the next lemma.

Lemma 10 *The Riemannian gradient of* (5.6) *with respect to metric* (5.7) *is*

$$\mathrm{grad}\, \hat{f}(U_k, d_k) = (M - U_k \,\mathrm{sym}(U_k^* M), v) \tag{5.8}$$

where $\mathrm{sym}(A) = (A + A^*)/2$,

$$M = 2\,\mathrm{grad}\, H(U_k \,\mathrm{Diag}(d_k) U_k^*) U_k \,\mathrm{Diag}(d_k)$$

and

$$v = \mathrm{diag}(U_k^* \,\mathrm{grad}\, H(U_k \,\mathrm{Diag}(d_k) U_k^*) U_k).$$

Proof The directional directive of $\hat{f}(U_k, d_k)$ along $(\eta_k, u) \in \mathrm{T}_{U_k} \,\mathrm{St}(k,n) \times \mathbb{R}^k$ is given by

$$\begin{aligned} \mathrm{D}\, \hat{f}(U_k, d_k)[(\eta_k, u)] &= \mathrm{Re}(\mathrm{tr}(\eta_k^* 2\,\mathrm{grad}\, H(U_k \,\mathrm{Diag}(d_k) U_k^*) U_k \,\mathrm{Diag}(d_k))) \\ &\quad + u^T \,\mathrm{diag}(U_k^* \,\mathrm{grad}\, H(U_k \,\mathrm{Diag}(d_k) U_k^*) U_k) = g^P((M, v), (\eta_k, u)). \end{aligned}$$

Therefore, $(P_{\mathrm{T}_{U_k} \mathrm{St}(k,n)}(M), v) = (M - U_k \,\mathrm{sym}(U_k^* M), v)$ is the Riemannian gradient by [AMS08, (3.31)].

Lemma 11 is the basic property for designing the algorithm. The details for increasing rank are given in Algorithm 3.

Lemma 11 *Let* $U \in \mathrm{St}(p,n)$ *and* $\Theta = (\sigma_1, \ldots, \sigma_p)^T$, $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p > 0$. *Matrix* A *denotes* $\mathrm{grad}\, H(U_k \,\mathrm{Diag}(d_k) U_k^*)$. *There exists a vector* $u \in \mathrm{span}(U_\perp)$ *such that* $u^* A u < 0$ *if and only if*

$$g^P((0_{n\times(p+1)}, (0, \ldots, 0, 1)^T), \mathrm{grad}\, \hat{f}(R, (\Theta^T, 0)^T)) < 0$$

where $(R, (\Theta^T, 0)^T) = ((U \quad u), (\Theta^T, 0)^T)$.

Proof The result is easily seen from (5.8) in Lemma 10.

Algorithm 3 Increase Rank

Require: Full rank matrix $Y \in \mathbb{C}_*^{n\times p}$;
Ensure: $(R, \Theta) \in \mathrm{St}(p+1, n) \times \mathbb{R}^{p+1}$ or Y;
1: Thin singular value decomposition for Y, i.e., $Y = U \operatorname{Diag}(\sigma_1, \ldots, \sigma_p) V^*$, where $U \in \mathbb{C}^{n\times p}$, $V \in \mathbb{C}^{p\times p}$ and $\sigma_1 \geq \ldots \geq \sigma_p > 0$;
2: (Approximately) solve the Rayleigh quotient problem

$$\min_{u\in\mathrm{span}(U_\perp), u^*u=1} u^*Au,$$

where $A = \operatorname{grad} H(U \operatorname{Diag}(\sigma_1^2, \ldots, \sigma_p^2)U^*)$.
3: **if** find u such that $u^*Au < 0$ **then**
4: return $R = (U \;\; u)$ and $\Theta = \operatorname{diag}(\sigma_1, \ldots, \sigma_p, 0)$;
5: **else**
6: return Y;
7: **end if**

Combining the ideas of Riemannian optimization methods and the procedure of increasing rank yield Algorithm 4.

This rank increase algorithm is related to the algorithm discussed in [JBAS10]. Algorithm 4 generalizes their algorithm to Hermitian positive semidefinite matrices. The literature on optimization over symmetric or Hermitian positive semidefinite matrices consists of algorithms that use either rank increase or rank decrease that are applied to problems where the rank of the optimizer can be bounded from above or below a priori. One may ask what if all ideas of Riemannian optimization methods on a fixed rank manifold and the procedure of increasing and reducing rank are combined. Note that since the procedure of decreasing rank does not necessarily give an iterate that reduce the cost function, combining the idea of increasing and reducing the rank may produce iterates whose ranks are not fixed eventually and thereby the iterates may be even not convergent. Recently, a rigorous definition of a rank adaptation strategy for optimization with rank inequality constraints based on the notion of rank-related Riemannian retractions has been developed and shown to be superior to the heuristics in the literature for problems such as weighted matrix approximation [ZHG$^+$14, ZHG$^+$]. Current work on the topic includes the adaptation of the strategy to problems with additional constraints such as Hermitian [Zho].

6 Experiments

In this section, numerical simulations for noiseless problems and those with Gaussian noise are used to illustrate the performance of the proposed method. The required Riemannian objects are derived in Section 6.1 and the experimental environment and parameters are defined in Section 6.2. Several Riemannian optimization algorithms are compared for a range of parameters in Section 6.3 and a representative selected for use in the more extensive set of numerical experiments used to demonstrate the efficiency of the Riemannian approach

Algorithm 4 Rank Increasing Algorithm

Require: $p > 0$; $Y_p^{(0)} \in \mathbb{C}^{n\times p}$ a representation of initial point $\pi(Y_p^{(0)})$ for f; Stopping criterion threshold ϵ; Riemannian optimization methods (a) and (b).
Ensure: W
for $k = 0, 1, 2, \ldots$ **do**
  Apply the Riemannian optimization method (a) for cost function f over $\mathbb{C}_*^{n\times p}/\mathcal{O}_p$ with initial point $\pi(Y_p^{(k)})$ until i-th iterate $W^{(i)}$ satisfying $g(\operatorname{grad} f, \operatorname{grad} f) < \epsilon^2$.
  $p \leftarrow p + 1$
  Apply Algorithm 3 with input $W^{(i)}$;
  if The output of Algorithm 3 is still $W^{(i)}$ **then**
    return $W = W^{(i)}$
  else {The output is (R, Θ)}
    Apply one step of the Riemannian optimization method (b) for cost function $\hat{f}$ over $\operatorname{St}(p, n) \times \mathbb{R}^p$ with initial point (R, Θ) and obtain next iterate $(\hat{R}, \hat{\Theta})$.
    Set $Y_p^{(k+1)} = \hat{R} \operatorname{diag}(\sqrt{\hat{\Theta}})$;
  end if
end for

to PhaseLift. In Section 6.4, the Riemannian approach is compared to the algorithm used in the convex programming approach of [CESV13, CSV13] that represents the current PhaseLift state-of-the-art. Finally, the peformance is evaluated for a complex-valued image from [CESV13] using masks that satisfy the assumptions of the PhaseLift framework and binary masks that do not. The latter are of interest from a practical point of view.

6.1 Gradient, and Action of Hessian and Complexity for PhaseLift

When the entries in the noise ϵ are drawn from the normal distribution with mean 0 and variance τ, the cost functions of (3.3) and (3.6) are essentially identical, i.e., for (3.3), $H_1(X) = \|b - \operatorname{diag}(ZXZ^*)\|_2^2 + \kappa \operatorname{tr}(X)$, and for (3.6), $H_2(X) = \frac{1}{\tau^2}\|b - \operatorname{diag}(ZXZ^*)\|_2^2 + \kappa \operatorname{tr}(X)$. Without loss of generality, only the cost function

$$H(X) = \frac{\|b - \operatorname{diag}(ZXZ^*)\|_2^2}{\|b\|_2^2} + \kappa \operatorname{tr}(X)$$

is considered. The Euclidean gradient and the action of the Euclidean Hessian of H are given in Lemma 12.

Lemma 12 *The Euclidean gradient of H is*

$$\operatorname{grad} H(X) = \frac{2}{\|b\|_2^2} Z^* \operatorname{Diag}(\operatorname{diag}(ZXZ^*) - b)Z + \kappa I_{n\times n}.$$

The action of the Euclidean Hessian at X along V is

$$\operatorname{Hess} H(X)[V] = \frac{2}{\|b\|_2^2} Z^* \operatorname{Diag}(\operatorname{diag}(ZVZ^*))Z,$$

where $V = V^*$.

Proof The directional derivative of $H(X)$ along direction V, $\mathrm{D}\,H(X)[V]$, is by definition equal to $g^E(V, \operatorname{grad} H(X))$. In addition, we have

$$\begin{aligned}\mathrm{D}\,H(X)[V] =& \frac{2}{\|b\|_2^2}(\operatorname{diag}(ZXZ^*) - b)^* \operatorname{diag}(ZVZ^*) + \kappa \operatorname{tr}(V)\\ =& \frac{2}{\|b\|_2^2} \operatorname{tr}(\operatorname{Diag}(\operatorname{diag}(ZXZ^*) - b)(ZVZ^*)) + \kappa \operatorname{tr}(V)\\ =& \frac{2}{\|b\|_2^2} \operatorname{Re}(\operatorname{tr}(Z^* \operatorname{Diag}(\operatorname{diag}(ZXZ^*) - b)ZV)) + \kappa \operatorname{tr}(V)\\ =& g^E(V, \frac{2}{\|b\|_2^2} Z^* \operatorname{Diag}(\operatorname{diag}(ZXZ^*) - b)Z + \kappa I_{n\times n}),\end{aligned}$$

which means $\operatorname{grad} H(X) = \frac{2}{\|b\|_2^2} Z^* \operatorname{Diag}(\operatorname{diag}(ZXZ^*) - b)Z + \kappa I_{n\times n}$. From this it is straightforward to compute the action of the Euclidean Hessian on a matrix.

The gradients and actions of Hessians of functions F_p and f_p, Euclidean and Riemannian respectively, constructed using Lemmas 3 and 9, are

$$\operatorname{grad} F(Y) = \frac{4}{\|b\|_2^2} Z^* \operatorname{Diag}(\operatorname{diag}(ZYY^*Z^*) - b)ZY + \kappa Y,$$

the action of the Hessian of F at Y on $\eta \in \mathbb{C}^{n\times p}$ is given by

$$\begin{aligned}\operatorname{Hess} F(Y)[\eta] =& \frac{4}{\|b\|_2^2} Z^* \operatorname{Diag}(\operatorname{diag}(ZYY^*Z^*) - b)Z\eta + \kappa\eta\\ &+ \frac{4}{\|b\|_2^2} Z^* \operatorname{Diag}(\operatorname{diag}(Z(\eta Y^* + Y\eta^*)Z^*))ZY,\end{aligned}$$

the horizontal lift of the gradient of (5.3) at Y is

$$(\operatorname{grad} f(\pi(Y)))_{\uparrow_Y} = P_Y^h(\frac{4}{\|b\|_2^2} Z^* \operatorname{Diag}(\operatorname{diag}(ZYY^*Z^*) - b)ZY + \kappa Y),$$

and the action of Hessian of (5.3) at $\pi(Y)$ along $\eta_{\pi(Y)} \in \mathrm{T}_{\pi(Y)} \mathbb{C}_*^{n\times p}/\mathcal{O}$ satisfies

$$(\operatorname{Hess} f(\pi(Y))[\eta_{\pi(Y)}])_{\uparrow_Y} = P_Y^h(\dot{M} - \eta_{\uparrow_Y}\Omega),$$

where $\dot{M} = \frac{4}{\|b\|_2^2} Z^* \operatorname{Diag}(\operatorname{diag}(ZYY^*Z^*) - b)Z\eta_{\uparrow_Y} + \frac{4}{\|b\|_2^2} Z^* \operatorname{Diag}(\operatorname{diag}(Z(\eta_{\uparrow_Y} Y^* + Y\eta_{\uparrow_Y}^*)Z^*))ZY + \kappa\eta_{\uparrow_Y}$, Ω is the skew-symmetric matrix that solves

$$\Omega Y^*Y + Y^*Y\Omega = Y^*M - M^*Y,$$

and $M = \frac{4}{\|b\|_2^2} Z^* \operatorname{Diag}(\operatorname{diag}(ZYY^*Z^*) - b)ZY + \kappa Y$.

The complexities of evaluations of the function value, gradient and action of Hessian of F_p are all of the same order, $O(lpns \max_i(\log(n_i))$. The complexities of evaluations of the function value, gradient and action of Hessian of f_p are $O(lpns \max_i(\log(n_i))$, $O(lpns \max_i(\log(n_i)) + O(np^2) + O(p^3)$ and

$O(lpns\max_i(\log(n_i)) + O(np^2) + O(p^3)$ respectively. If $p << n$ then all these complexities are dominated by $O(lpns\max_i(\log(n_i))$.

For the optimization problems in the PhaseLift framework for phase retrieval, Theorem 2 is extremely important due to the following reasons. First, the cost function H in PhaseLift is convex over a convex domain $\mathcal{D}_n$. Therefore, finding a stationary point of H by using the cost function F is sufficient to find a global minimizer of H. Second, the rank of the desired minimizer of H in PhaseLift is one. It follows that by using a low-rank factorization-based cost function F_p with small $p > 1$ it is possible to find the desired unique rank-one minimizer of H and optimizing F_p with small $p > 1$. (This approach also has lower storage and computational complexity compared to optimizing H.) The theorem guarantees that any minimizer, Y_p, of F_p with rank less than p must have rank 1 and $Y_pY_p^*$ must be the global minimizer of H. Stationary points, including local minimizers, of F_p with rank p can be discarded if found and the algorithm restarted appropriately. If an X with numerical rank $1 < r < p$ is encountered when iterating using F_p then X is not a stationary point and the rank reduction strategy increases efficiency by removing the unnecessary directions from X and continuing the iteration on F_r.

Even though stationary points with rank 1 that are not local minimizers of F_1 may exist, their presence tends to simply slow the algorithm rather than stopping the iteration at the saddle point. As expected, therefore, running with $p > 1$ avoids this issue completely. There is no theorem guaranteeing that the iterates generated by optimizing F_p with adapting but remaining greater than 1 always converge to an approximation of the rank-one minimizer of H in PhaseLift, such convergence occurred in all of the experiments below and if it were to occur Theorem 2 allows detection and restarting as discussed above.

6.2 Data, Parameters and Notations

All codes are written in Matlab and all experiments are performed in Matlab R2014a on a 64 bit Ubuntu system with 3.6 GHz CPU (Intel (R) Core (TM) i7-4790).

Unless indicated in the description of the experiments, the following test data are used. A complex number $a + b\sqrt{-1}$ is said to be drawn from a distribution in this paper if both a and b are drawn from the distribution independently. The entries of the true solution x_* and Gaussian masks $w_i, i = 1, \ldots l$ are drawn from the standard normal distribution. The entries of x_* are further normalized by $\|x_*\|_2$ and the $w_i, i = 1, \ldots, l$ are further normalized by $\sqrt{n}$. For the noiseless problem, the measurement b is set to be $\mathrm{diag}(Zx_*x_*^*Z^*)$ and for Gaussian noise problem, the measurement b is set to be $\mathrm{diag}(Zx_*x_*^*Z^*) + \epsilon$, where the entries of $\epsilon \in \mathbb{R}^m$ are drawn from the normal distribution with mean 0 and variance τ that is specified later for each experiment.

The initial iterate $Y_p^{(0)}$ is generated by orthonormalizing a complex n-by-p matrix with entries drawn from the standard normal distribution. Note the initial iterate is chosen such that its singular values are identical. This choice

Table 1 Notation for reporting the experimental results.

$iter$	summation of numbers of iterations in Step 2 of Algorithm 2
nf	number of function evaluations
ng	number of gradient evaluations
nH	number of operations of the form $\mathcal{H}\eta$
f_f	the function value of the final iterate
t	average wall time (seconds)

of initial point minimizes the influence of magnitudes of singular values of the initial point. In other words, if a bias of magnitudes of singular values is shown during iteration, one knows that the bias is generated by the algorithm and the surface of the cost function not the initial iterate.

The stopping criterion of Algorithm 2 requires the norm of gradient to less than 10^{-6} and the minimum number of iterations at each rank is 10. Since Algorithm 2 eventually optimizes the cost function f_p with $p = 1$, the penalty term $\mathrm{tr}(X) = \mathrm{tr}(YY^*)$ for minimizing the rank is not necessary. Therefore, κ is set to be 0 for Algorithm 2.

To obtain sufficiently stable timing results, an average time is taken of several runs with identical parameters for a total runtime of at least 1 minute. The notation used when reporting the experimental results is given in Table 1.

6.3 A Representative Riemannian Method and Choices of Initial Point Size and Rank Reducing Threshold

Practically, the dimension of the domain is usually large so RBFGS and RTR-SR1 are not applicable since the Hessian approximation requires too much storage. The Riemannian optimization methods used in the experiments are representative of the state-of-the-art and have satisfactory convergent rate. The methods are: RNewton, RTR-Newton, LRBFGS, LRTR-SR1 and RCG. Note that in RNewton and RTR-Newton solving the linear system or local model uses a truncated CG method [Ste83, CGT00] which only requires the action of Hessian. Therefore, RNewton and RTR-Newton, like all of the other methods, are applicable to a large scale problem. The line search algorithm used with RNewton and LRBFGS is the back tracking which finds an Armijo point [AMS08, Definition 4.2.2]. RCG uses the line search algorithm of [NW06, Algorithm 3.5] to find a point satisfying the strong Wolfe conditions with $c_1 = 10^{-4}$ and $c_2 = 10^{-1}$. The vector transports in LRBFGS and LRTR-SR1 use vector transport by parallelization [HAG14, Section 2.3.1] and RCG uses vector transport by differentiated retraction.

The initial p, denoted p_0, and the rank reducing parameter δ are set to 8 and 0.9 respectively. The parameters l and s are 6 and 2 respectively. The experimental results are reported in Table 2 with several values of n_1 and n_2 for the noiseless problem. LRBFGS is the fastest algorithm among them. RNewton

Table 2 Comparison of RNewton (1), RTR-Newton (2), LRBFGS (3), LRTR-SR1 (4) and RCG (5) for average of 10 random runs. The subscript ν indicates a scale of 10^{ν}.

(n_1, n_2)		(32, 32)	(32, 64)	(64, 64)	(64, 128)	(128, 128)	(128, 256)	(256, 256)
(1)	$iter$	2.34_{1}	2.58_{1}	2.92_{1}	3.26_{1}	3.8_{1}	4.56_{1}	5.14_{1}
	nf	3.24_{1}	3.36_{1}	3.58_{1}	4.12_{1}	4.58_{1}	5.74_{1}	5.88_{1}
	ng	2.34_{1}	2.58_{1}	2.92_{1}	3.26_{1}	3.8_{1}	4.56_{1}	5.14_{1}
	nH	4.56_{2}	3.39_{2}	3.86_{2}	4.00_{2}	4.21_{2}	4.88_{2}	5.73_{2}
	f_f	1.25_{-13}	2.17_{-13}	1.37_{-14}	4.83_{-13}	1.59_{-14}	2.32_{-12}	3.79_{-13}
	t	2.54	3.25	9.28	1.72_{1}	3.02_{1}	7.57_{1}	1.98_{2}
(2)	$iter$	2.82_{1}	2.66_{1}	2.84_{1}	3.06_{1}	3.42_{1}	3.62_{1}	3.36_{1}
	nf	2.82_{1}	2.66_{1}	2.84_{1}	3.06_{1}	3.42_{1}	3.62_{1}	3.36_{1}
	ng	2.82_{1}	2.66_{1}	2.84_{1}	3.06_{1}	3.42_{1}	3.62_{1}	3.36_{1}
	nH	3.00_{2}	6.13_{2}	4.36_{2}	484	5.27_{2}	5.33_{2}	5.77_{2}
	f_f	2.11_{-14}	2.86_{-13}	4.53_{-13}	4.69_{-13}	3.03_{-13}	9.74_{-14}	3.45_{-14}
	t	1.83	5.45	8.23	1.92_{1}	3.42_{1}	6.07_{1}	1.42_{2}
(3)	$iter$	9.6_{1}	1.04_{2}	1.16_{2}	1.29_{2}	1.40_{2}	1.77_{2}	1.87_{2}
	nf	9.78_{1}	1.06_{2}	1.20_{2}	133	1.45_{2}	1.84_{2}	191
	ng	9.6_{1}	1.04_{2}	1.16_{2}	1.29_{2}	1.40_{2}	1.77_{2}	1.87_{2}
	f_f	6.40_{-12}	6.82_{-12}	7.61_{-12}	1.04_{-11}	1.58_{-11}	2.24_{-11}	3.55_{-11}
	t	6.00_{-1}	1.03	1.90	3.37	6.86	1.59_{1}	3.27_{1}
(4)	$iter$	1.45_{2}	1.44_{2}	1.56_{2}	1.71_{2}	1.88_{2}	2.29_{2}	2.43_{2}
	nf	1.45_{2}	1.44_{2}	1.56_{2}	1.71_{2}	1.88_{2}	2.29_{2}	2.43_{2}
	ng	1.45_{2}	1.44_{2}	1.56_{2}	1.71_{2}	1.88_{2}	2.29_{2}	2.43_{2}
	f_f	1.24_{-11}	1.08_{-11}	1.50_{-11}	3.67_{-11}	3.10_{-11}	4.82_{-11}	2.04_{-10}
	t	9.97_{-1}	1.64	3.16	6.20	1.22_{1}	3.14_{1}	6.81_{1}
(5)	$iter$	8.74_{1}	86	9.16_{1}	9.42_{1}	1.00_{2}	1.09_{2}	1.16_{2}
	nf	2.66_{2}	2.59_{2}	2.77_{2}	2.89_{2}	3.11_{2}	3.45_{2}	3.73_{2}
	ng	2.55_{2}	250	266	2.80_{2}	3.02_{2}	3.36_{2}	3.65_{2}
	f_f	3.11_{-12}	3.47_{-12}	5.42_{-12}	7.94_{-12}	1.16_{-11}	1.60_{-11}	2.53_{-11}
	t	1.18	2.00	3.74	7.18	1.47_{1}	3.63_{1}	9.54_{1}

and RTR-Newton require least number of iteration but the relative expensive action of Hessian evaluation makes them relatively slow. The performance of RCG depends significantly on the choice of parameters c_1 and c_2 used in the strong Wolfe conditions. We tested a few of them and RCG is always slower than LRBFGS. (A representative result is reported.) Therefore, LRBFGS is chosen to be the representative Riemannian method for the more detailed comparisons with a standard nonmanifold method.

Table 3 presents the experimental results of the representative Riemannian method LRBFGS with $l = 6$, $s = 2$ and several values of p_0 and δ. The noiseless problem is used. The mean and the standard derivation of computational time of 100 runs of LRBFGS are reported. In addition, Figures 1 present an example that shows the relationships among iteration number, computational time and cost function values with $\delta = 0.9$ and several values of p_0.

The average computational time and the standard derivation of $p_0 = 1$ are much larger relatively than the other starting ranks. This clearly shows that the performance simply optimizing over matrices with the fixed optimal rank is not a reliable and efficient method. Additionally, note that, when the initial point is close to the global rank-one minimizer, then Algorithm 2 with $p_0 = 1$ is fast, otherwise Algorithm 2 with $p_0 = 1$ is usually very slow. Figure 1 illustrates this point by using $p_0 = 1$ to generate a random initial iterate. As expected, using $p_0 > 1$ significantly improves the performance of the algorithm. It allows the algorithm to search on a larger dimensional space and find a more reasonable initial point for Algorithm 2 when p finally reduces to 1. The values $p_0 = 8$ and $\delta = 0.9$ are therefore chosen for use with LRBFGS in the later comparisons.

Table 3 The mean and the standard derivation of computational time of 100 runs of LRBFGS with variant p_0 and δ and output format is (mean)/(the standard derivation). Since δ does not take effect for $p_0 = 1$, the row corresponding to $p_0 = 1$ has only one result.

	δ	0.95	0.9	0.85	0.8	0.75	0.7
p_0	1	3.33/2.47					
	2	2.67/2.28	2.67/2.28	2.68/2.29	2.67/2.31	2.66/2.28	2.62/2.28
	4	$1.76/5.43_{-1}$	$1.80/4.74_{-1}$	$1.79/3.91_{-1}$	$1.81/3.89_{-1}$	$1.84/3.83_{-1}$	$1.83/2.87_{-1}$
	8	$1.67/1.72_{-1}$	$1.82/1.65_{-1}$	$2.05/2.23_{-1}$	$2.34/2.93_{-1}$	$2.63/3.30_{-1}$	$2.97/4.00_{-1}$

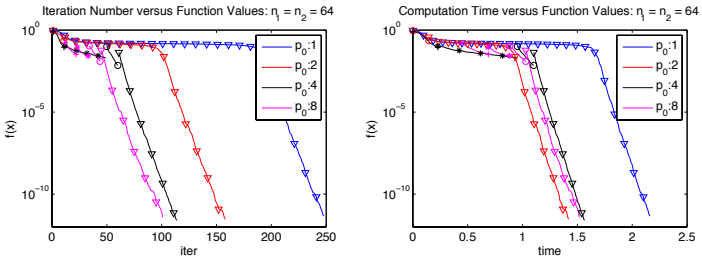

Fig. 1 The figures show relationships among iteration, computational time and cost function value for $n_1 = n_2 = 64$ and $\delta = 0.9$. The curves generated by different p_0 are displayed in different colors. Markers $\triangledown$, o, $*$, $+$, $\triangle$, x, $\triangleright$ and $\triangleleft$ on the curves indicate the ranks, from 1 to 8 respectively, of the corresponding iterations.

6.4 Comparisons with a Standard Low-rank Method

Candes et al., [CESV13,CSV13] use a Matlab library TFOCS [BCG11] that contains a variety of accelerated first-order methods given in [Nes04] and, in particular, the method based on FISTA [BT09] is used to optimize the cost functions in PhaseLift. FISTA [BT09] works as follows. Given an initial point $X^{(0)}$, set $B^{(0)} = X^{(0)}$ and $\theta^{(0)} = 1$, and inductively define

$$X^{(i)} = P_{\mathcal{D}_n}(B^{(i-1)} - t^{(i)} \operatorname{grad} H(B^{(i-1)})), \tag{6.1}$$

$$\begin{aligned} \theta^{(i)} &= 2(1 + \sqrt{1 + \frac{4}{(\theta^{(i-1)})^2}})^{-1}, \\ \beta^{(i)} &= \theta^{(i)}((\theta^{(i-1)})^{-1} - 1), \\ B^{(i)} &= X^{(i)} + \beta^{(i)}(X^{(i)} - X^{(i-1)}), \end{aligned} \tag{6.2}$$

where $t^{(i)}$ is an appropriate step size, e.g., by back tracking. For large scale problems, matrix $X^{(i)}$ is stored by its low-rank approximation computed via projection, i.e., $\sum_{j=1}^{k} \max(\sigma_j^{(i)}, 0) v_j^{(i)} (v_j^{(i)})^*$, where $\sum_{j=1}^{n} \sigma_j^{(i)} v_j^{(i)} (v_j^{(i)})^*$ is an eigenvalue decomposition of $X^{(i)}$ and eigenvalues satisfies $\sigma_1^{(i)} \geq \sigma_2^{(i)} \geq \ldots \geq \sigma_n^{(i)}$. The orthogonal projection (6.1) is obtained by using "eigs" with function handle providing matrix vector multiplication since the matrix vector multiplication of $(B^{(i-1)} - t^{(i)} \operatorname{grad} H(B^{(i-1)}))v$ is cheap for any $v \in \mathbb{C}^n$. The low-rank

subtraction (6.2) is computed exactly by doubling the storage. LR-FISTA is used to denote the low-rank version of FISTA.

As in [CESV13], the difference between the true solution and the minimizer is measured by the relative mean-square error (RMSE) that is defined to be $\min_{a:|a|=1} \|ax - x_*\|_2/\|x_*\|_2$ and RMSE in dB is defined by $10\log_{10}(\text{RMSE})$. The scale of the noise is measured by the signal-to-noise ratio (SNR) in dB that is defined to be $\text{SNR} = 10\log_{10}(\|b\|_2^2/\|b-\hat{b}\|_2^2)$, where $b = \text{diag}(Zx_*x_*^*Z^*)$ and $\hat{b}$ is the noise measurements.

The stopping criterion of LR-FISTA requires that the Frobenius norm of the relative difference between $X^{(i)}$ and $X^{(i-1)}$ is less than 10^{-6} or the number of iterations is greater than 2000, i.e., $\|X^{(i)} - X^{(i-1)}\|_F/\|X^{(i)}\|_F < 10^{-6}$ or $iter > 2000$.

In practice, the choice of κ needs careful consideration. The standard golden section search [Kie53] is used by Candes et al. [CESV13] to find the best κ that gives the smallest RMSE. This method can be used only when the true solution x_* is known. In addition, Candes et al. indicate that one would have to find the best κ via a strategy like cross validation or generalized cross validation. However, since Algorithm 2 was designed with rank in mind the default choice for problems with and without noise is $\kappa = 0$. κ is also chosen to be 0 in LR-FISTA for noiseless measurements in order to recover the exact solution. The effect of using $\kappa > 0$ in both algorithms is discussed below when Gaussian noisy measurements are used.

Tables 4 and 5 report experimental results of comparisons of Algorithm 2 and LR-FISTA for the noiseless and Gaussian noise problems (3.3) and (3.6) respectively. For the Gaussian noise problem, τ is 10^{-4} and the corresponding SNR is 31.05 dB in this experiment. Multiple examples with different random seeds and different SNR show similar results. First, increasing k for LR-FISTA usually does not improve the performance in the sense of efficiency and effectiveness for both noiseless and Gaussian noise problems. Therefore, $k = 1$ is used in the later comparison. Second, increasing κ usually does not reduce the RMSE. When it does, the RMSE values are not reduced significantly. Therefore, $\kappa = 0$ is used in the later comparisons for Gaussian noise problems. Third, Algorithm 2 outperforms LR-FISTA significantly in the sense that Algorithm 2 provides similar accuracy usually while requiring fewer operations of all types (cost function evaluation, gradients etc.) and yielding a significantly smaller computational time.

Figure 2 shows the relationships between RMSE and SNR for both Algorithm 2 and LR-FISTA methods. Clearly, increasing the number of masks, l, improves the accuracy, i.e., reduces the RMSE. In addition, the RMSE given by Algorithm 2 is similar to that given by LR-FISTA. All the curves indicate that increasing SNR in dB reduces the RMSE in dB linearly, which is consistent with the report of [CESV13, Figure 4] for a 1-dimensional problem.

Table 4 Comparisons of Algorithm 2 and LR-FISTA for the noiseless PhaseLift problem (3.3) with $n_1 = n_2 = 64$ and several values of k. ♯ represents the number of iterations reach the maximum.

noiseless	Algorithm 2	LR-FISTA				
		1	2	4	8	16
iter	124	1022	377	601	1554	$2000^{\sharp}$
nf	129	2212	804	1278	3360	4322
ng	124	1106	402	639	1680	2161
f_f	4.62_{-12}	8.18_{-12}	4.50_{-11}	4.64_{-12}	1.54_{-11}	1.27_{-9}
RMSE	6.34_{-6}	1.01_{-5}	1.74_{-5}	1.46_{-5}	1.10_{-4}	2.56_{-3}
t	2.12	1.27_{2}	5.25_{1}	9.35_{1}	3.48_{2}	6.86_{2}

Table 5 Comparisons of Algorithm 2 and LR-FISTA for the noise PhaseLift problem (3.6) with SNR be 31.05 dB, $n_1 = n_2 = 64$ and several values of k and κ. ♯ represents the number of iterations reach the maximum.

noise	κ	Algorithm 2	LR-FISTA				
			1	2	4	8	16
iter	10^{-2}	84	409	$2000^{\sharp}$	$2000^{\sharp}$	$2000^{\sharp}$	$2000^{\sharp}$
	10^{-4}	122	978	$2000^{\sharp}$	$2000^{\sharp}$	$2000^{\sharp}$	$2000^{\sharp}$
	10^{-6}	128	1027	$2000^{\sharp}$	$2000^{\sharp}$	$2000^{\sharp}$	$2000^{\sharp}$
	0	138	1070	$2000^{\sharp}$	$2000^{\sharp}$	$2000^{\sharp}$	$2000^{\sharp}$
nf	10^{-2}	86	886	4280	4284	4290	4280
	10^{-4}	129	2116	4296	4316	4300	4318
	10^{-6}	132	2210	4266	4312	4336	4316
	0	143	2306	4308	4322	4314	4320
ng	10^{-2}	84	526	3468	3376	3242	3371
	10^{-4}	122	1105	2148	2158	2150	2159
	10^{-6}	128	1105	2712	2156	2168	2158
	0	138	1153	2154	2161	2157	2160
f_f	10^{-2}	1.63_{-1}	1.63_{-1}	1.77_{-1}	2.24_{-1}	2.75_{-1}	3.04_{-1}
	10^{-4}	1.80_{-3}	1.80_{-3}	1.81_{-3}	2.19_{-3}	4.55_{-3}	7.01_{-3}
	10^{-6}	1.84_{-5}	1.84_{-5}	1.91_{-5}	2.35_{-5}	3.55_{-5}	7.62_{-5}
	0	4.08_{-7}	4.08_{-7}	1.16_{-6}	6.27_{-6}	2.51_{-5}	8.89_{-5}
RMSE	10^{-2}	1.80_{-1}	1.80_{-1}	2.64_{-1}	3.60_{-1}	4.19_{-1}	4.45_{-1}
	10^{-4}	2.63_{-3}	2.63_{-3}	6.46_{-3}	2.17_{-2}	4.98_{-2}	6.57_{-2}
	10^{-6}	6.72_{-4}	6.72_{-4}	1.09_{-3}	2.10_{-3}	3.53_{-3}	6.27_{-3}
	0	6.70_{-4}	6.70_{-4}	1.09_{-3}	2.18_{-3}	4.01_{-3}	7.29_{-3}
t	10^{-2}	1.59	5.17_{1}	3.79_{2}	4.48_{2}	5.73_{2}	9.45_{2}
	10^{-4}	2.06	1.21_{2}	3.05_{2}	3.17_{2}	4.80_{2}	7.85_{2}
	10^{-6}	2.13	1.27_{2}	2.75_{2}	3.01_{2}	4.64_{2}	7.04_{2}
	0	2.20	1.34_{2}	2.63_{2}	2.98_{2}	4.32_{2}	6.91_{2}

6.5 Performance of PhaseLift on a Complex-valued Image

The complex-valued 2D image shown in Figure 3 is used in [CESV13] to illustrate the performance of PhaseLift for noiseless measurements when the masks are either Gaussian or binary and for noisy measurements when the masks are Gaussian. Gaussian masks are as defined in Section 6.2. A binary set of masks contain a mask that is all 1 (which yields the original image) and several other masks comprising elements that are 0 or 1 with equal probability.

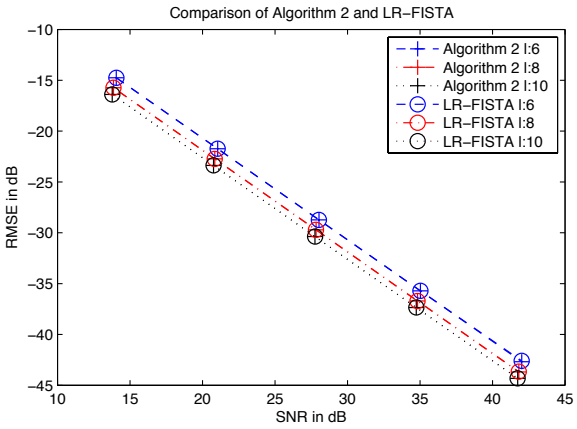

Fig. 2 Comparison of the Algorithm 2 performance and the LR-FISTA performance for Gaussian noise problems with $n_1 = n_2 = 64$, $k = 1$ and $p_0 = 4$.

It is empirically shown in [CESV13] that the noiseless measurements given by either Gaussian masks or binary masks enable retrieval of the phase using LR-FISTA in the sense that the differences between original image and reconstructed images are almost invisible to the eye. Quantitative error for the noiseless experiments are not given. For noisy measurements, only the Gaussian masks are considered and they give a satisfactory quantitative error in reconstructed phase and image. The implication, not stated in [CESV13], that the binary masks require an unacceptably large computational in the presence of noise is demonstrated below.

In all cases, the computation times are seen to be in hours for solving the PhaseLift optimization problem using LR-FISTA in [CESV13] (the details of the computational platform are not given). The experimental results below demonstrate that due to the efficiency of Algorithm 2, both phase retrieval by PhaseLift for both Gaussian and binary masks on this image is computationally practical. Times required for LR-FISTA, implemented using the same computational library primitives as the Riemannian algorithms for fair comparison, are included to verify its computational impracticality implied in [CESV13].

Since the computational time for LR-FISTA on this image is potentially very large, the default setting of stopping criterion given above is modified to include the additional condition of stopping when the computational time is greater than 1 hour.

Table 6 shows the RMSE and computational times of Algorithm 2 and LR-FISTA with varying number and types of masks. In all experiments LR-FISTA is stopped after 1 hour of computation, i.e., it does not achieve $\|X^{(i)} - X^{(i-1)}\|_F / \|X^{(i)}\|_F < 10^{-6}$ for any experiment. Additionally, after 1 hour the

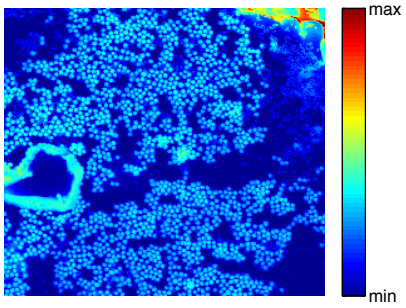

Fig. 3 Image of the absolute value of the complex-valued image.

retrieval is not satisfactory and the RMSE is consistently and significantly worse than the RMSE achieved by Algorithm 2 with exception of the single high noise Gaussian mask experiment where both achieve the same RMSE in dB.

The RMSE achieved by Algorithm 2 shows the influence of the number and the type of masks as expected. Using a few Gaussian masks is sufficient to recover satisfactory phase and achieve acceptable RMSE values.

The quality of recovery by binary masks suffers from noise and an insufficient number of masks. The ultimate quality of retrieval as the SNR reduces from high noise to noiseless is determined by the number of masks used. For example, for 32 masks -10 dB is achieved with an SNR of 40 dB and does not improve for noiseless. For a fixed SNR, increasing the number of binary masks, as expected, reduces the RMSE. Note that the computational time as the number of binary masks increases does not increase that rapidly. So there is more work required to understand the PhaseLift cost function surface for binary masks and its implications for the convergence of Riemannian algorithms and local/global minima structure.

Figures 4, 5 and 6 present the reconstructed images and their errors from Algorithm 2. The errors in the images shown **are magnified 10 times for display purposes**. As expected, there are almost no visible errors in the reconstructed images given by Gaussian masks, and the errors of images given by reconstructed phases using binary masks are visible especially when the number of binary masks is 6.

7 Conclusion

In this paper, the recently proposed PhaseLift framework for solving the phase retrieval problem has motivated the consideration of cost functions H on the

Table 6 RMSE and computational time (second) results with varying number and types of masks are shown in format RMSE/TIME. ♯ represents the computational time reaching 1 hour, i.e., 3.6_3 seconds.

	Algorithm 2			LR-FISTA		
SNR (dB)	20	40	inf	20	40	inf
6 Gaussian masks	$-20.8/4.30_1$	$-40.8/4.50_1$	$-54.3/4.19_1$	-20.8/♯	-34.3/♯	-34.3/♯
6 binary masks	$-1.41/7.90_2$	$-8.88/4.24_2$	$-9.64/4.42_2$	-0.84/♯	-3.03/♯	-3.04/♯
32 binary masks	$-6.56/6.84_2$	$-25.2/7.36_2$	$-25.9/6.54_2$	-2.17/♯	-2.35/♯	-2.38/♯

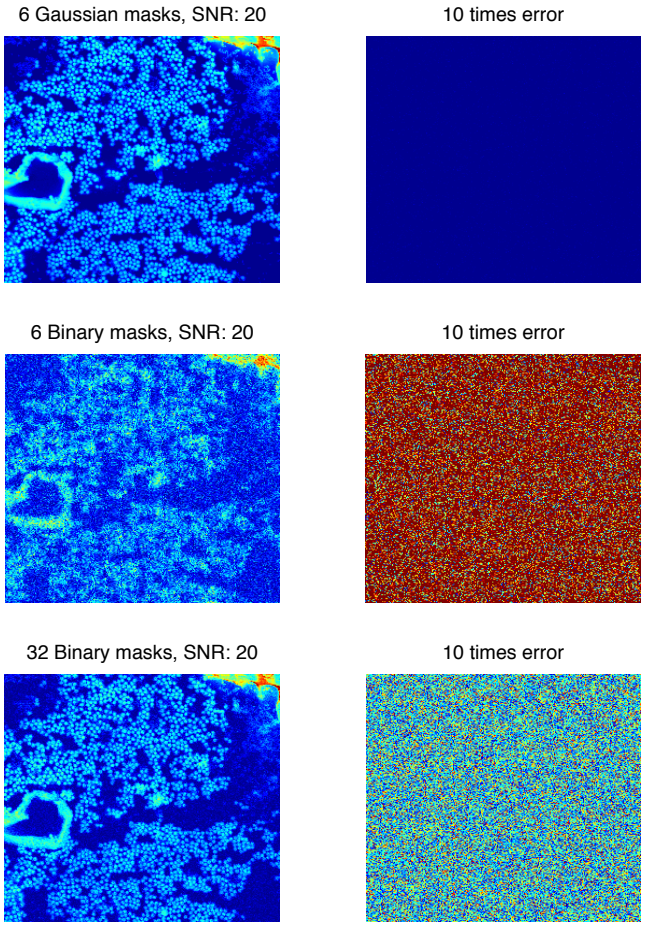

Fig. 4 Reconstructions via PhaseLift with varying number and types of masks. For the same number and types of masks, the reconstructions of noisy measurements with SNR 20 are shown. The error is shown in the images by magnifying 10 times.

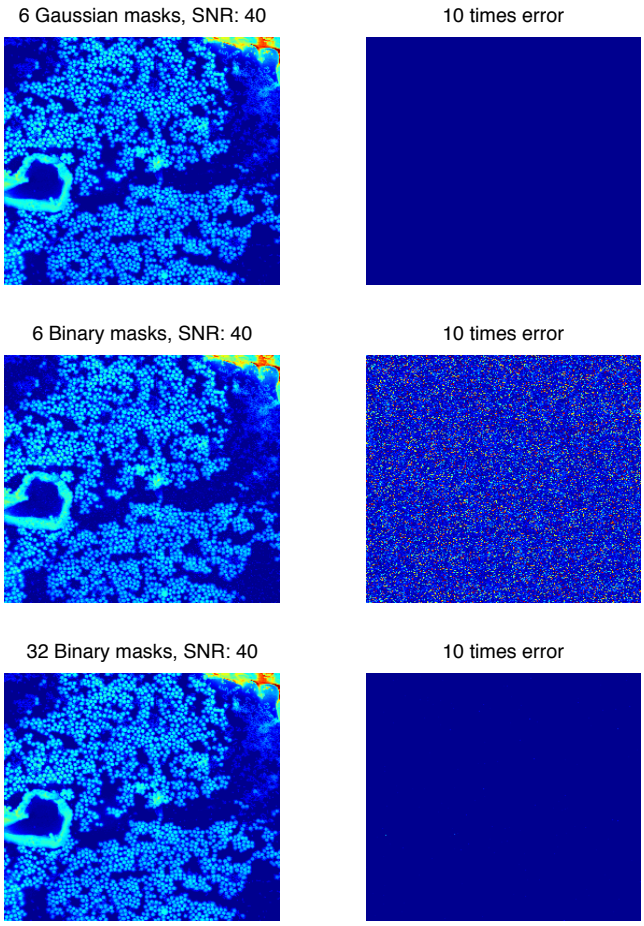

Fig. 5 Reconstructions via PhaseLift with varying number and types of masks. For the same number and types of masks, the reconstructions of noisy measurements with SNR 40 are shown. The error is shown in the images by magnifying 10 times.

set of complex Hermitian positive semidefinite matrices $\mathcal{D}_n$ that include the PhaseLift cost function.

An alternate cost function F related to factorization is used to replace the cost function H, i.e., $F(Y) = H(YY^*)$. The optimality conditions of H are related to the properties of F and the important optimality condition, Theorem 2, shows that if Y_p is a rank deficient minimizer of F_p, then $Y_pY_p^*$ is a stationary point of H. For general problems defined on $\mathcal{D}_n$, if r_*, the rank of the desired minimizer of cost function H, is low, the optimality condition

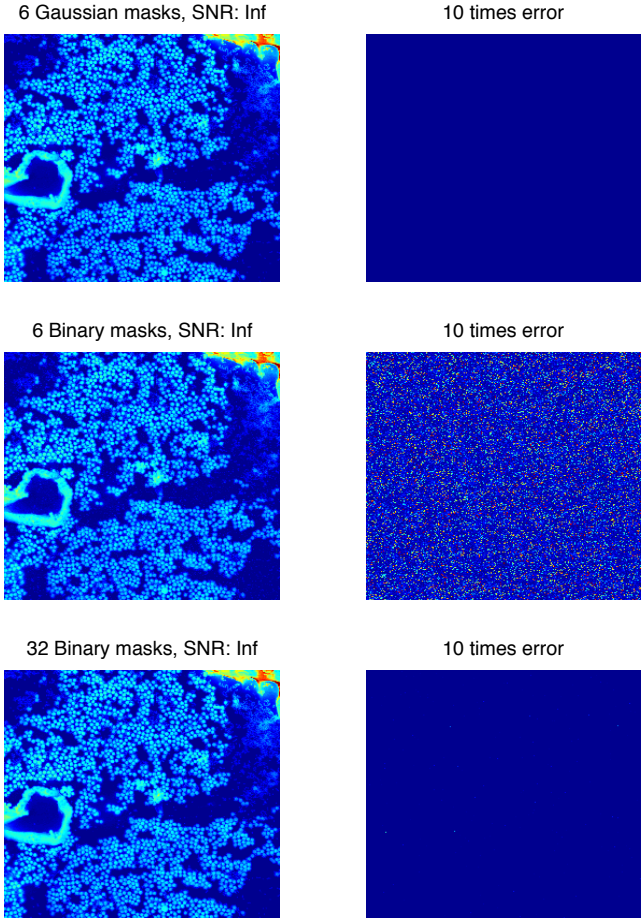

Fig. 6 Reconstructions via PhaseLift with varying number and types of masks. For the same number and types of masks, the reconstructions of noiseless measurements are shown. The error is shown in the images by magnifying 10 times.

suggests the use of the alternate cost function F with $p > r_*$. If r_* is small, then a small p can be used and optimization on F_p can be more efficient than optimization on H.

Additionally, Algorithm 2 and Algorithm 4 based on optimization on a fixed rank manifold and dynamically reducing or increasing rank are developed for optimizing the cost function F. For optimization on a fixed rank manifold, recently developed state-of-the-art Riemannian optimization methods on a quotient space are used and a superlinear convergent rate is obtained.

For the case of the noiseless phase retrieval problem in the PhaseLift framework, obtaining a rank-one minimizer Y_p of F_p with $p > 1$ is shown to be equivalent to obtaining the rank-one global minimizer $Y_p Y_p^*$ of H. Empirically, in finite precision arithmetic, choosing $p_0 > 1$ and using Algorithm 2 always yields a (approximately) rank-one minimizer for both the noiseless and noisy phase retrieval problems. The computational time of Algorithm 2 with LRBFGS is demonstrated to be significantly smaller than that of LR-FISTA with accuracy is at least as good as LR-FISTA when the latter manages to converge in an acceptable amount of time. In [CESV13], it is pointed out that the algorithm LR-FISTA may be too slow for large-scale images and development of a fast algorithm is a future research. The unacceptably large computational time of LR-FISTA is empirically verified here and Algorithm 2 using the limited memory forms of the Riemannian fixed-rank optimization algorithms, specifically LRBFGS for the phase retrieval problems, are clearly seen to be practical in terms of computational time and recovery quality for a wide range of problem sizes including those for which LR-FISTA fails.

8 Acknowledgement

We thank Stefano Marchesini at Lawrence Berkeley Notional Laboratory for providing the gold balls data set and granting permission to use it.

References

AIDV09. P.-A. Absil, M. Ishteva, L. De Lathauwer, and S. Van Huffel. A geometric Newton method for Oja's vector field. *Neural Computationomputation*, 21(5):1415–33, May 2009. doi:10.1162/neco.2008.04-08-749.

AMS08. P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, Princeton, NJ, 2008.

Bak08. C. G. Baker. *Riemannian manifold trust-region methods with applications to eigenproblems*. PhD thesis, Florida State University, Department of Computational Science, 2008.

BCG11. S. Becker, E. J. Cand, and M. Grant. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, 3:165–218, 2011.

BDP+07. O. Bunk, A. Diaz, F. Pfeiffer, C. David, B. Schmitt, D. K. Satapathy, and J. F. van der Veen. Diffractive imaging for periodic samples: retrieving one-dimensional concentration profiles across microfluidic channels. *Acta crystallographica. Section A, Foundations of crystallography*, 63(Pt 4):306–314, July 2007. doi:10.1107/S0108767307021903.

Bla04. R. Blankenbecler. Three-dimensional image reconstruction. II. Hamiltonian method for phase recovery. *Physical Review B*, 69(6):064108, February 2004. doi:10.1103/PhysRevB.69.064108.

BM03. S. Burer and R. D. C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, February 2003. doi:10.1007/s10107-002-0352-8.

BMS10. S. Bonnabel, G. Meyer, and R. Sepulchre. Adaptive filtering for estimation of a low-rank positive semidefinite matrix. *Proceedings of the 19th Internaltional Symposium on Mathematical Theory of Networks and Systems (MTNS 2010)*, 2010.

BS79. Y. M. Bruck and L. G. Sodin. On the ambiguity of the image recontruction problem. *Optics Communications*, 30(3):304–308, 1979.

BS10. S. Bonnabel and R. Sepulchre. Rank-preserving geometric means of positive semi-definite matrices. *Proceedings of the 19th Internaltional Symposium on Mathematical Theory of Networks and Systems (MTNS 2010)*, pages 209–214, 2010.

BT09. A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, January 2009. doi:10.1137/080716542.

BV04. S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004. doi:10.1017/CBO9780511804441.

Cas12. T. P. Cason. *Role Extraction in Networks*. PhD thesis, Université catholique de Louvain, Department of Engineering Sciences, 2012.

CESV13. E. J. Candès, Y. C. Eldar, T. Strohmer, and V. Voroninski. Phase retrieval via matrix completion. *SIAM Journal on Imaging Sciences*, 6(1):199–225, 2013. arXiv:1109.0573v2.

CGT00. A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-region methods*. MPS/SIAM Series on Optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

CL13. E. J. Candès and X. Li. Solving quadratic equations via phaselift when there are about as many equations as unknowns. *Foundations of Computational Mathematics*, June 2013. doi:10.1007/s10208-013-9162-z.

CMWL07. C. Chen, J. Miao, C. w. Wang, and T. K. Lee. Application of optimization technique to noncrystalline x-ray diffraction microscopy: Guided hybrid input-output method. *Physical Review B*, 76(6):064113, August 2007. doi:10.1103/PhysRevB.76.064113.

CSV13. E. J. Candès, T. Strohmer, and V. Voroninski. PhaseLift : Exact and stable signal recovery from magnitude measurements via convex programming. *Communications on Pure and Applied Mathematics*, 66(8):1241–1274, 2013.

DH14. L. Demanet and P. Hand. Stable optimizationless recovery from phaseless linear measurements. *Journal of Fourier Analysis and Applications*, 20(1):199–221, 2014.

DTD11. F. De Terán and F. M. Dopico. The equation $XA + AX^* = 0$ and the dimension of *congruence obits. *Electronic Journal of Linear Algebra*, 22:448–465, 2011.

Els03. V. Elser. Solution of the crystallographic phase problem by iterated projections. *Section A: Foundations of Crystallography*, pages 201–209, 2003.

Fie78. J. R. Fienup. Reconstruction of an object from the modulus of its Fourier transform. *Optics letters*, 3(1):27–29, 1978.

Fie82. J. R. Fienup. Phase retrieval algorithms: a comparison. *Applied optics*, 21(15):2758–69, August 1982.

GS72. R. W. Gerchberg and W. O. Saxton. A practical algorithm for the determination of phase from image and diffraction plane pictures. *Optik*, 35:237–246, 1972.

GW04. M. X. Goemans and D. P. Williamson. Approximation algorithms for max-3-cut and other problems via complex semidefinite programming. *Journal of Computer and System Sciences*, 68(2):442–470, March 2004. doi:10.1016/j.jcss.2003.07.012.

HAG14. W. Huang, P.-A. Absil, and K. A. Gallivan. A Riemannian symmetric rank-one trust-region method. *Mathematical Programming*, February 2014. doi:10.1007/s10107-014-0765-1.

Har93. R. W. Harrison. Phase problem in crystallography. *Journal of the Optical Society of America A*, 10(5):1046–1055, May 1993. doi:10.1364/JOSAA.10.001046.

Hay82. M. H. Hayes. The reconstruction of a multidimensional sequence from the phase or magnitude of its fourier transform. *IEEE Transactions on Acoustics speech and signal processing*, 30(2):140–154, 1982.

HGA14. W. Huang, K. A. Gallivan, and P.-A. Absil. A Broyden class of quasi-Newton methods for Riemannian optimization. *Submitted for publication*, 2014.

HM94. U. Helmke and J. B. Moore. *Optimization and dynamical systems*. Spfinger-Verlag, June 1994. doi:10.1109/JPROC.1996.503147.

HS95. U. Helmke and M. A. Shayman. Critical points of matrix least squares distance functions. *Linear Algebra and its Applications*, 19(1995):1–19, 1995.

Hua13. W. Huang. *Optimization algorithms on Riemannian manifolds with applications*. PhD thesis, Florida State University, Department of Mathematics, 2013.

JBAS10. M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5):2327–2351, 2010.

Kat80. T. Kato. *Perturbation theory For linear operators*. Springer-Verlag, New York, NY, corrected edition, 1980.

Kie53. J. Kiefer. Sequential minimax search for a maximum. *Proceedings of the American Mathematical Society*, 4:502–506, 1953.

KL07. Othmar Koch and Christian Lubich. Dynamical low-rank approximation. *SIAM Journal on Matrix Analysis and Applications*, 29(2):434–454, January 2007. doi:10.1137/050639703.

Lee11. J. M. Lee. *Introduction to smooth manifolds*, volume 36 of *Graduate Texts in Mathematics*. Springer New York, New York, NY, January 2011. doi:10.1016/B978-0-12-387667-6.00013-0.

Mar07. S. Marchesini. Invited article: a unified evaluation of iterative projection algorithms for phase retrieval. *Review of scientific instruments*, 78(1):011301, January 2007. doi:10.1063/1.2403783.

MBS11. G. Meyer, S. Bonnabel, and R. Sepulchre. Regression on fixed-rank positive semidefinite matrices : a Riemannian approach. *Journal of Machine Learning Research*, 12:593–625, 2011.

MISE08. J. Miao, T. Ishikawa, Q. Shen, and T. Earnest. Extending X-ray crystallography to allow the imaging of noncrystalline materials, cells, and single protein complexes. *Annual review of physical chemistry*, 59:387–410, January 2008. doi:10.1146/annurev.physchem.59.032607.093642.

MJBS09. G. Meyer, M. Journée, S. Bonnabel, and R. Sepulchre. From subspace learning to distance learning: a geometrical optimization approach. *Proceedings of the IEEE/SP 15th Workshop on Statistical Signal Processing*, pages 385–388, 2009.

Nar73. R. Narasimhan. *Analysis on real and complex manifolds*. MASSON & CIE, EDITEUR, second edition, 1973.

Nes04. Y. Nesterov. *Introductory lectures on convex programming: a basic course*, volume I. Springer, 2004.

NW06. J. Nocedal and S. J. Wright. *Numerical optimization*. Springer, second edition, 2006.

OHM06. R. Orsi, U. Helmke, and J. B. Moore. A Newton-like method for solving rank constrained linear matrix inequalities. *Automatica*, 42(11):1875–1882, November 2006. doi:10.1016/j.automatica.2006.05.026.

RW12. W. Ring and B. Wirth. Optimization methods on Riemannian manifolds and their application to shape space. *SIAM Journal on Optimization*, 22(2):596–627, January 2012. doi:10.1137/11082885X.

San85. J. L. C. Sanz. Mathematical considerations for the problem of Fourier transform phase retrieval from magnitude. *SIAM Journal on Applied Mathematics*, 45(4):651–664, 1985.

Ste83. T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637, 1983.

TTT99. K. C. Toh, M. J. Todd, and R. H. Tutuncu. SDPT3 a matlab software package for semidefinite programming. *Optimization methods and software*, 11:545–581, 1999.

VAV09. B. Vandereycken, P.-A. Absil, and S. Vandewalle. Embedded geometry of the set of symmetric positive semidefinite matrices of fixed rank. *Proceedings of the IEEE 15th Workshop on Statistical Signal Processing*, pages 389–392, 2009.

VAV12. B. Vandereycken, P.-A. Absil, and S. Vandewalle. A Riemannian geometry with complete geodesics for the set of positive semidefinite matrices of fixed rank. *IMA Journal of Numerical Analysis*, 33(2):481–514, July 2012. doi:10.1093/imanum/drs006.

VV10. B. Vandereycken and S. Vandewalle. A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations. *SIAM Journal on Matrix Analysis and Applications*, 31(5):2553–2579, January 2010. doi:10.1137/090764566.

Wal63. A. Walther. The question of phase retrieval in optics. *Optica Acta: International Journal of Optics*, 10(1):41–49, January 1963. doi:10.1080/713817747.

WDM13. I. Waldspurger, A. DAspremont, and S. Mallat. Phase recovery, maxcut and complex semidefinite programming. *Mathematical Programming*, December 2013. doi:10.1007/s10107-013-0738-9.

ZHG+. G. Zhou, W. Huang, K. A. Gallivan, P. Van Dooren, and P.-A. Absil. Rank-constrained optimization: A Riemannian manifold approach. In *Submitted for publication*.

ZHG+14. G. Zhou, W. Huang, K. A. Gallivan, P. Van Dooren, and P.-A. Absil. A Riemannian optimization technique for rank inequality constraints, 2014.

Zho. G. Zhou. *Rank-constrained optimization: A Riemannian manifold approach*. PhD thesis, Department of Mathematics, Florida State University, in preparation.

A Proof that $\mathcal{C}_*^{n\times p}/\mathcal{O}_p$ is a smooth manifold over $\mathbb{R}$

Let $\mathcal{G}$ and $\mathcal{M}$ be a Lie group and a smooth manifold over $\mathbb{R}$ respectively. The following definitions and theorems, which can be found in [Lee11], are also used.

Definition 4 An *action* of $\mathcal{G}$ on $\mathcal{M}$ is a map $\mathcal{G}\times\mathcal{M}\to\mathcal{M}$, written as $(g,p)\mapsto g\cdot p$, that satisfies (i) $g_1\cdot(g_2\cdot p)=(g_1g_2)\cdot p$ for all $g_1,g_2\in\mathcal{G}$ and $p\in\mathcal{M}$ and (ii) $e\cdot p=p$ for all $p\in\mathcal{M}$, where e is the identity. In addition, the group action is *smooth* if the map is smooth.

Definition 5 Let $\mathcal{G}$ act smoothly on $\mathcal{M}$.

- The group action is *free* if $g\cdot p=p$ for any $p\in\mathcal{M}$ implies $g=e$;
- The group action is *proper* if the action map $\mathcal{G}\times\mathcal{M}\to\mathcal{M}\times\mathcal{M}$ satisfies that the pre-image of any compact set is compact.

Theorem 3 *[Lee11, Corollary 21.6] Every continuous action by a compact Lie group on a manifold is proper.*

Theorem 4 *[Lee11, Theorem 21.10] Suppose $\mathcal{G}$ is a Lie group acting smoothly, freely, and properly on a smooth manifold $\mathcal{M}$. Then the orbit space $\mathcal{M}/\mathcal{G}$ is a topological manifold and has a unique smooth structure with the property that the quotient map $\pi:\mathcal{M}\to\mathcal{M}/\mathcal{G}$ is a smooth submersion.*

Now we are ready to prove $\mathcal{C}_*^{n\times p}/\mathcal{O}_p$ is a smooth manifold over $\mathbb{R}$, where $\mathcal{C}_*^{n\times p}$ is the complex noncompact Stiefel manifold and $\mathcal{O}_p$ is the group of p-by-p unitary matrices.

Proof The complex noncompact Stiefel manifold $\mathcal{C}_*^{n\times p}$ and the group $\mathcal{O}_p$ can be viewed as a manifold and a group over $\mathbb{R}$ respectively [Nar73, Page 55]. Therefore, Theorems 3 and 4 are applicable.

By Theorem 4, we need only show that the group action $(O_p,Y_p)\mapsto Y_pO_p$ is smooth, free, and proper. The smoothness follows from the smoothness of matrix multiplication. Since Y_p is full rank, $Y_pO_p=Y_p$ implies $O_p=I_p$. Therefore, the action is free. The properness follows from the compactness of $\mathcal{O}_p$ and Theorem 3.