Journal of Scientific Computing (2019) 79:128–147
https://doi.org/10.1007/s10915-018-0857-9

Fast Rank-One Alternating Minimization Algorithm for Phase Retrieval

Jian-Feng Cai[1] · Haixia Liu[2] · Yang Wang[1]

Received: 28 October 2017 / Revised: 18 September 2018 / Accepted: 12 October 2018 /
Published online: 22 October 2018

Abstract
The *phase retrieval* problem is a fundamental problem in many fields, which is appealing for investigation. It is to recover the signal vector $\tilde{\mathbf{x}} \in \mathbb{C}^d$ from a set of N measurements $b_n = |\mathbf{f}_n^* \tilde{\mathbf{x}}|^2,\ n = 1, \ldots, N$, where $\{\mathbf{f}_n\}_{n=1}^N$ forms a frame of $\mathbb{C}^d$. Existing algorithms usually use a least squares fitting to the measurements, yielding a quartic polynomial minimization. In this paper, we employ a new strategy by splitting the variables, and we solve a bi-variate optimization problem that is quadratic in each of the variables. An alternating gradient descent algorithm is proposed, and its convergence for any initialization is provided. Since a larger step size is allowed due to the smaller Hessian, the alternating gradient descent algorithm converges faster than the gradient descent algorithm (known as the Wirtinger flow algorithm) applied to the quartic objective without splitting the variables. Numerical results illustrate that our proposed algorithm needs less iterations than Wirtinger flow to achieve the same accuracy.

Keywords Phase retrieval · Alternating minimization · Alternating gradient descent · Rank-one · Non-convex optimizaton

JFC was supported in part by HKRGC Grant 16300616. Part of the research work of YW was completed while the author was at Department of Mathematics, Michigan State University. This research was supported in part by the National Science Foundation Grant DMS-1043032 and AFOSR Grant FA9550-12-1-0455 and HKRGC Grant 16306415.

✉ Haixia Liu
liuhaixia@hust.edu.cn

Jian-Feng Cai
jfcai@ust.hk

Yang Wang
yangwang@ust.hk

[1] Department of Mathematics, The Hong Kong University of Science and Technology, Kowloon, Hong Kong

[2] School of Mathematics and Statistics, Huazhong University of Science and Technology, Wuhan, China

1 Introduction

Let $f(x) \in L^2(\mathbb{R}^d)$. It is well known that the map $f \mapsto \widehat{f}$, where $\widehat{f}$ denotes the Fourier transform of f, is an isometry in $L^2(\mathbb{R}^d)$ and hence f can be uniquely reconstructed from $\widehat{f}$. In many applications such as X-ray crystallography, however, we can only measure the magnitude $|\widehat{f}|$ of the Fourier transform. This raises the following question: Is it still possible to reconstruct f from $|\widehat{f}|$? This is the classic *phase retrieval* problem.

The phase retrieval problem has a natural generalization to finite dimensional Hilbert spaces. Such an extension has important applications in imaging, optics, communication, audio signal processing and more [10,15,16,19,24]. It is in this finite Hilbert space setting that phase retrieval has become one of the growing areas of research in recent years.

Let $\mathbf{H}$ be a (real or complex) Hilbert space of finite dimension. Without loss of generality we identify $\mathbf{H}$ with $\mathbb{H}^d$ where $\mathbb{H} = \mathbb{R}$ or $\mathbb{H} = \mathbb{C}$. A set of elements $\mathcal{F} = \{\mathbf{f}_n\}$ in $\mathbf{H}$ is called a *frame* if it spans $\mathbf{H}$. Given this frame any vector $\mathbf{x} \in \mathbf{H}$ can be reconstructed from the inner products $\{\langle \mathbf{x}, \mathbf{f}_n \rangle\}$. Often it is convenient to identify the frame $\mathcal{F}$ with the corresponding *frame matrix* $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N]$. The phase retrieval problem in $\mathbf{H}$ is:

The Phase Retrieval Problem *Let $\mathcal{F} = \{\mathbf{f}_n\}$ be a frame in $\mathbf{H}$. Can we reconstruct any $\mathbf{x} \in \mathbf{H}$ up to a unimodular scalar from $\{|\langle \mathbf{x}, \mathbf{f}_n \rangle|\}$,and if so, how?*

$\mathcal{F}$ is said to be *phase retrievable (PR)* if the answer is affirmative. There is an alternative formulation. Consider the equivalence relation $\sim$ on $\mathbf{H}$: $\mathbf{x}_1 \sim \mathbf{x}_2$ if there is a constant $b \in \mathbb{H}$ with $|b| = 1$ such that $\mathbf{x}_1 = b\mathbf{x}_2$. Let $\overline{\mathbf{H}} := \mathbf{H}/\sim$. We shall use $\underline{\mathbf{x}}$ to denote the equivalent class containing $\mathbf{x}$. For any given frame $\mathcal{F} = \{\mathbf{f}_n : 1 \le n \le N\}$ in $\mathbf{H}$ define the map $\mathbf{M}_{\mathcal{F}} : \overline{\mathbf{H}} \longrightarrow \mathbb{R}_+^N$ by

$$\mathbf{M}_{\mathcal{F}}(\underline{\mathbf{x}}) = [|\langle \mathbf{x}, \mathbf{f}_1 \rangle|^2, \ldots, |\langle \mathbf{x}, \mathbf{f}_N \rangle|^2]^T. \tag{1.1}$$

The phase retrieval problem asks whether an $\underline{\mathbf{x}} \in \overline{\mathbf{H}}$ is uniquely determined by $\mathbf{M}_{\mathcal{F}}(\underline{\mathbf{x}})$, i.e. whether $\mathbf{M}_{\mathcal{F}}$ is *injective* on $\overline{\mathbf{H}}$.

Many challenging and fundamental problems in phase retrieval remain open. For example, for phase retrieval in $\mathbb{C}^d$ it is still unknown what is the minimal number of vectors needed for a set of vectors $\mathcal{F}$ to be phase retrievable. A challenging problem of very practical importance is the computational efficiency of phase retrieval algorithms. So far the existing phase retrieval algorithms can be loosely divided into four categories: (A) Using frames with very large N, in the order of $N \ge O(d^2)$, (B) Convex relaxation algorithms using random frames, (C) Non-convex optimization with a quartic [8,21], Poisson log likelihood [11] or quadratic [12,20,25–27] objective functions with random frames, and (D) Constructing special frames $\mathcal{F}$ that allow for fast and robust phase retrieval reconstruction of $\mathbf{x}$.

The first category is based on the fact that each $|\langle \mathbf{x}, \mathbf{f}_n \rangle|^2$ is a linear combination of monomials $x_i^* x_j$. The reconstruction of $\mathbf{x}$ can be attained by solving for these monomials, provided that there are enough equations, i.e. N is large enough. We will need $N \ge \frac{1}{2}d(d+1)$ in the real case and $N \ge d^2$ in the complex case. The reconstruction then becomes solving a system of linear equations if we treat all monomials as independent variables. One can also obtain robustness results under such framework. The weakness of this approach is that when d is large the number of variables and the number of measurements needed will explode, making it generally impractical and slow. Several constructions for special frames were designed (e.g. [3]) with which one can compute $\mathbf{x}$ efficiently ("painless reconstruction"). But this does not reduce the number of required measurements.

The second category of methods employs convex relaxation techniques like those of compressive sensing [2,6,7,9,13,23]. By considering $\mathbf{X} = \mathbf{x}\mathbf{x}^*$ we can rewrite the map $\mathbf{M}_{\mathcal{F}}$ as

$$\mathbf{M}_{\mathcal{F}}(\mathbf{X}) = [\mathbf{x}^*\mathbf{A}_1\mathbf{x}, \ldots, \mathbf{x}^*\mathbf{A}_N\mathbf{x}]^T = [\mathrm{tr}(\mathbf{A}_1\mathbf{X}), \ldots, \mathrm{tr}(\mathbf{A}_N\mathbf{X})]^T \tag{1.2}$$

where $\mathbf{A}_n = \mathbf{f}_n\mathbf{f}_n^*$. Obviously $\mathbf{M}_{\mathcal{F}}$ is a linear map from $\mathbb{C}^{d\times d}$ to $\mathbb{R}^N$. Thus, the original problem is now a linear equation $\mathbf{M}_{\mathcal{F}}(\mathbf{X}) = \mathbf{b}$ subject to the constraints $\mathbf{X} \geq 0$ and has rank 1. This type of problems is not convex and cannot be solved efficiently in general. Nevertheless, when random frames are used, it was shown in [9] that with high probability if $N \geq O(d \log d)$ then phase retrieval is equivalent to solving the following convex optimization

$$\min_{\mathbf{X}} \mathrm{tr}(\mathbf{X}) \quad \text{subject to} \quad \mathbf{X} \geq 0, \quad \mathbf{M}_{\mathcal{F}}(\mathbf{X}) = \mathbf{b}. \tag{1.3}$$

The bound was later improved to $N \geq O(d)$ [7]. The robustness of the method was also proved. This convex relaxation method, called *PhaseLift*, solves (1.3) using semi-definite programming. It can be done with reasonable efficiency for small d (typically up to about $d = 1000$ on a PC). Several refinements and variations of PhaseLift have also being proposed, e.g. PhaseCut and MaxCut [6,23] for Fourier measurements with random masks. However, all of them employ semi-definite programming with unknown matrices $O(d) \times O(d)$, which for larger d becomes slow and impractical. To avoid large unknown matrices in 'lifting' and semidefinite programming, some convex relaxation methods operate in the natural domain of the signal (i.e., $\mathbb{R}^d$ or $\mathbb{C}^d$) to relax each measurement equation $|\mathbf{a}_n^*\mathbf{x}|^2 = b_n$ to an inequality $|\mathbf{a}_n^*\mathbf{x}|^2 \leq b_n$. Let $\mathbf{a}_0 \in \mathbb{C}\backslash\{0\}$ be a non-vanishing correlation with $\mathbf{x}$ in the sense that

$$|\mathbf{a}_0^*\mathbf{x}| \geq \delta\|\mathbf{a}_0\|_2\|\mathbf{x}\|_2, \ \delta \in (0, 1).$$

Solving the signal $\mathbf{x}$ can be obtained by solving the following

$$\max_{\mathbf{x}\in\mathbb{C}^d} \mathrm{Re}(\mathbf{a}_0^*\mathbf{x}), \ \text{subject to } |\langle \mathbf{a}_n, \mathbf{x}\rangle| \leq b_n, n = 1, \ldots, N, \tag{1.4}$$

where $\mathrm{Re}(\cdot)$ is the real part of complex value. Geometric conditions are characterized for the success of phase retrieval through (1.4), which holds true with high probability if random frames are used [2,13].

The third category of methods is non-convex optimization methods. We optimize non-convex functions that penalize the error of nonlinear equations $|\langle \mathbf{x}, \mathbf{f}_n\rangle|^2 = b_n, n = 1, \ldots, N$. In general, such optimization problems are very difficult to solve, because the non-convex objective may have numerous local minima. Surprisingly, when random frames are used, these non-convex optimization problems are not as difficult as they appear, and many results are available on their performance guarantee by specific numerical solvers. In [20,27], an alternating minimization algorithm is applied to solve

$$\min_{\mathbf{x},\mathbf{p}\in\mathbb{C}^d} \frac{1}{N}\sum_{n=1}^{N}\left(\langle \mathbf{x}, \mathbf{f}_n\rangle p_n - \sqrt{b_n}\right)^2 \quad \text{s.t.} \quad |p_n| = 1, \ n = 1, \ldots, N, \tag{1.5}$$

where $\mathbf{p}$ represents the missing phases. In other words, the algorithm alternates between estimating the missing phase information and the candidate solution. It was proved that the alternating minimization algorithm [20] converges linearly to the underlying true solution up

to a unimodular scalar. Some other methods consider the non-linear fittings to the measurements equation and solve

$$\min_{\mathbf{x}\in\mathbb{C}^d} \frac{1}{N}\sum_{n=1}^{N} \left||\langle \mathbf{x}, \mathbf{f}_n\rangle|^2 - b_n\right|^p. \tag{1.6}$$

In [12], they consider (1.6) with $p = 1$ as a composite optimization problem and linearize $|\langle \mathbf{x}, \mathbf{f}_n\rangle|^2 - b_n$ locally. In [8], a Wirtinger gradient flow algorithm is applied to solve (1.6) with $p = 2$, and it was proven that the algorithm is guaranteed to converge to the global minimizer of (1.6) when the algorithm is initialized by the so-called spectral initialization and the frame is random Gaussian or Fourier with random masks. To further improve the efficiency and robustness to noise, variants of the Wirtinger flow algorithm are proposed in, e.g., [11,29], and their convergence to the correct solution are provided. More recently, it is revealed in [21] that the objective in (1.6) with $p = 2$ actually has no spurious local minima if $\mathcal{F}$ is a random Gaussian frame with $N \geq O(d\log^3 d)$. Therefore, there are many other efficient algorithms that may be able to find the global minimizer of (1.6). Besides (1.5) and (1.6), we can also use the amplitude-based cost function and solve

$$\min_{\mathbf{x}\in\mathbb{C}^d} \frac{1}{N}\sum_{n=1}^{N} \left(|\langle \mathbf{x}, \mathbf{f}_n\rangle| - \sqrt{b_n}\right)^2. \tag{1.7}$$

Wang et al. [26] introduced an orthogonal-promoting initialization, which is followed by truncated generalized gradient descent iterations. Though non-convex algorithms have a better computation performance than convex ones, they still assume random frames for their performance guarantee, which may be impractical in real applications.

In the fourth category one strives to build special frames with far fewer elements but which still allow for fast and robust reconstruction. In [1] a deterministic graph-theoretic construction of a frame with Cd measurements was obtained. This is the few known deterministic construction that uses only $O(d)$ measurements and can robustly reconstruct *all* $\mathbf{x} \in \mathbb{C}^d$, at least in theory. Unfortunately the constant C is very large, so again computationally it would be impractical for large d. In [17] a highly efficient phase retrieval scheme using a small number of random linear combinations of Fourier transform measurements is developed. It uses $O(d\log d)$ measurements to guarantee robustness with high probability, and achieves the computational complexity of $O(d\log d)$. In numerical tests it easily performed robust phase retrieval for $d = 64{,}000$ in seconds on a laptop. A drawback is that it is robust only in the probabilistic sense; for a given $\mathbf{x}$ there is a small probability that the scheme will fail.

In this paper we develop an algorithm for phase retrieval that is both highly efficient and works for very general measurement matrices. Our algorithm is based on the ideas of convex relaxation in PhaseLift and the alternating minimization algorithm used for low rank matrix completion. By splitting the variables, our algorithm solves a bi-variant optimization problem whose objective is quadratic in one of the variables with the other fixed. We shall present both theoretical and numerical results, and discuss its efficient implementation.

2 Rank-One Minimization for Phase Retrieval

Let $\mathcal{X} = \{\mathbf{x}\mathbf{x}^* : \mathbf{x} \in \mathbb{H}^d\}$. As we noted in (1.2), $\mathcal{F} = \{\mathbf{f}_n\}_{n=1}^N$ in $\mathbb{H}^d$ is phase retrievable if and only if the map

$$\mathbf{M}_{\mathcal{F}}(\mathbf{X}) = [\mathbf{x}^*\mathbf{A}_1\mathbf{x}, \ldots, \mathbf{x}^*\mathbf{A}_N\mathbf{x}]^T = [\mathbf{f}_1 X\mathbf{f}_1^*, \ldots, \mathbf{f}_N X\mathbf{f}_N^*]^T \tag{2.1}$$

is an injective map from $\mathcal{X}$ to $\mathbb{R}^N$, where $\mathbf{A}_n = \mathbf{f}_n\mathbf{f}_n^*$. In the PhaseLift scheme the phase retrieval problem of solving for $\mathbf{M}_{\mathcal{F}}(\mathbf{X}) = \mathbf{b}$ subject to the constraints $\mathbf{X} \geq 0$ and has rank 1 (equivalent to $\mathbf{X} \in \mathcal{X}$) is being relaxed to solving the convex problem (1.3)

$$\min_{\mathbf{X}} \operatorname{tr}(\mathbf{X}) \quad \text{subject to} \quad \mathbf{X} \geq 0, \quad \mathbf{M}_{\mathcal{F}}(\mathbf{X}) = \mathbf{b}.$$

This relaxation yields the same solution to the original phase retrieval problem with high probability provided that the measurement frame matrix $\mathbf{F}$ is a Gaussian random $N \times d$ matrix with $N = O(d)$ for some unspecified constant, or the DFT matrix with random masks.

Still the drawback is that the measurement matrices are restricted to some specific types, which may or may not be practical for any given application. The optimization requires the use of semi-definite programming, which is slow in general and impractical for phase retrieval for large dimensions. Here we propose a new approach that resolves these difficulties.

The main idea is to relax the requirement $\mathbf{X} \in \mathcal{X}$ to simply $\operatorname{rank}(\mathbf{X}) = 1$. In other words we drop the requirement that $\mathbf{X}$ is Hermitian and positive semi-definite. Thus we consider solving the problem

$$\mathbf{M}_{\mathcal{F}}(\mathbf{X}) = [\mathbf{f}_1\mathbf{X}\mathbf{f}_1^*, \ldots, \mathbf{f}_N\mathbf{X}\mathbf{f}_N^*]^T = \mathbf{b} \quad \text{subject to} \quad \operatorname{rank}(\mathbf{X}) = 1, \tag{2.2}$$

or alternatively, given that noise might be present, solving the following problem:

$$\min_{\mathbf{X}} \|\mathbf{M}_{\mathcal{F}}(\mathbf{X}) - \mathbf{b}\| \quad \text{subject to} \quad \operatorname{rank}(\mathbf{X}) = 1. \tag{2.3}$$

Observe that in general the solution to (2.3) is not unique. If $\mathbf{X}$ is a solution then so is $\mathbf{X}^*$. To account for this ambiguity we shall use $\mathcal{R}_d(\mathbb{H})$ to denote the set of $d \times d$ rank-one matrices with the equivalence relation $\mathbf{X} \equiv \mathbf{X}^*$. We shall also let $\mathcal{S}_d(\mathbb{H})$ denote the set of $d \times d$ Hermitian rank-one matrices with entries in $\mathbb{H}$.

Theorem 2.1 *For $\mathcal{F} = \{\mathbf{f}_n\}_{n=1}^N \subset \mathbb{H}^d$ and $\mathbf{X} \in M_d(\mathbb{H})$ let*

$$\mathbf{M}_{\mathcal{F}}(\mathbf{X}) = [\mathbf{f}_1\mathbf{X}\mathbf{f}_1^*, \ldots, \mathbf{f}_N\mathbf{X}\mathbf{f}_N^*]^T.$$

(A) *For a generic $\mathcal{F} \subset \mathbb{H}^d$, $\mathbf{M}_{\mathcal{F}}$ is injective on $\mathcal{R}_d(\mathbb{H})$ if $N \geq 4d - 1$ for $\mathbb{H} = \mathbb{R}$, or if $N \geq 8d - 3$ for $\mathbb{H} = \mathbb{C}$.*

(B) *For a generic $\mathcal{F} \subset \mathbb{H}^d$, $\mathbf{M}_{\mathcal{F}}$ is injective on $\mathcal{S}_d(\mathbb{H})$ if $N \geq 2d + 1$ for $\mathbb{H} = \mathbb{R}$, or if $N \geq 4d - 1$ for $\mathbb{H} = \mathbb{C}$.*

Proof We shall identify $\mathcal{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N\}$ with its frame matrix $\mathbf{F}$ whose columns are $\{\mathbf{f}_n\}$. Consider the set of all 3-tuples

$$\mathcal{A} := \{(\mathbf{F}, \mathbf{X}, \mathbf{Y})\}$$

where $\mathbf{X}, \mathbf{Y} \in \mathcal{R}_d(\mathbb{H})$ or $\mathbf{X}, \mathbf{Y} \in \mathcal{S}_d(\mathbb{H})$ are distinct and satisfy $\mathbf{M}_{\mathcal{F}}(\mathbf{X}) = \mathbf{M}_{\mathcal{F}}(\mathbf{Y})$. We follow the technique in [4] of local dimension counting to prove our theorem.

Let $\mathbb{H}_+^d$ denote the set of vectors of $\mathbb{H}^d$ whose the first nonzero entry is real and positive. Note that any $d \times d$ rank-one matrix $\mathbf{Z}$ can be written uniquely as $\mathbf{Z} = a\mathbf{f}\mathbf{g}^*$ where $a \in \mathbb{H}$, $\mathbf{f}, \mathbf{g} \in \mathbb{H}_+^d$ and $\|\mathbf{f}\| = \|\mathbf{g}\| = 1$. Under this factorization $\mathbf{Z} \in \mathcal{S}_d(\mathbb{H})$ if and only if $\mathbf{f} = \mathbf{g}$ and $a \in \mathbb{R}$.

To prove the theorem there are 4 cases to be considered, with $\mathbb{H} = \mathbb{R}$ or $\mathbb{C}$ and $\mathbf{X}, \mathbf{Y} \in \mathcal{R}_d(\mathbb{H})$ or $\mathcal{S}_d(\mathbb{H})$. We deal with each case. Due to the similarity of the arguments we shall skip some redundant details.

Case 1 $\mathbb{H} = \mathbb{R}$ *and* $\mathbf{X}, \mathbf{Y} \in \mathcal{R}_d(\mathbb{R})$.

In this case, because $\mathbf{X}, \mathbf{Y}$ are distinct in $\mathcal{R}_d(\mathbb{R})$ each equality $\mathbf{f}_n \mathbf{X} \mathbf{f}_n^* = \mathbf{f}_n \mathbf{Y} \mathbf{f}_n^*$ yields a nontrivial constraint in the form of a quadratic polynomial equation for the (real) entries of $\mathbf{f}_n$. Furthermore, for different n the entries $\mathbf{f}_n$ are independent variables. Thus viewing the entries of $\mathbf{F}$ as points in $\mathbb{R}^{Nd}$, for any distinct $\mathbf{X}, \mathbf{Y} \in \mathcal{R}_d(\mathbb{R})$, those satisfying the constraint $\mathbf{M}_{\mathcal{F}}(\mathbf{X}) = \mathbf{M}_{\mathcal{F}}(\mathbf{Y})$ is a real algebraic variety of co-dimension $Nd - N$. By the unique factorization $\mathbf{X} = a\mathbf{f}\mathbf{g}^*$ discussed above each $\mathbf{X}$ has $2d - 1$ degrees of freedom. The same $2d - 1$ degree of freedom holds also for $\mathbf{Y}$. Thus the projection of $\mathcal{A} = \{(\mathbf{F}, \mathbf{X}, \mathbf{Y})\}$ to the first component has local dimension everywhere at most $Nd - N + 2(2d - 1)$. Suppose that $N \geq 4d - 1$. Then this local dimension has

$$Nd - N + 2(2d - 1) \leq Nd - 1 < Nd.$$

In other words, a generic $\mathbf{F} \in \mathbb{R}^{N \times d}$ is not a projection of an element in $\mathcal{A}$ to the first component. Thus for a generic $\mathcal{F}$ with $N \geq 4d - 1$ the map $\mathbf{M}_{\mathcal{F}}$ is injective on $\mathcal{R}_d(\mathbb{R})$.

Case 2 $\mathbb{H} = \mathbb{R}$ *and* $\mathbf{X}, \mathbf{Y} \in \mathcal{S}_d(\mathbb{R})$.

All arguments from Case 1 carry over to this case, except in the counting of degrees of freedom for $\mathbf{X}$ and $\mathbf{Y}$. Because now $\mathbf{X} = a\mathbf{f}\mathbf{f}^*$ there are exactly d degrees of freedom for $\mathbf{X}$. The same holds true for $\mathbf{Y}$. Thus the projection of $\mathcal{A} = \{(\mathbf{F}, \mathbf{X}, \mathbf{Y})\}$ to the first component has local dimension everywhere at most $Nd - N + 2d$. Suppose that $N \geq 2d + 1$. Then this local dimension has

$$Nd - N + 2d \leq Nd - 1 < Nd.$$

In other words, a generic $\mathbf{F} \in \mathbb{R}^{N \times d}$ is not a projection of an element in $\mathcal{A}$ to the first component. Thus for a generic $\mathcal{F}$ with $N \geq 2d + 1$ the map $\mathbf{M}_{\mathcal{F}}$ is injective on $\mathcal{S}_d(\mathbb{R})$.

Case 3 $\mathbb{H} = \mathbb{C}$ *and* $\mathbf{X}, \mathbf{Y} \in \mathcal{R}_d(\mathbb{C})$.

The main arguments from Case 1 carry to this case with slight modifications. A key difference is that we now view $\mathbf{F}$ as a point in $\mathbb{R}^{2Nd}$. Each constraint $\mathbf{f}_n \mathbf{X} \mathbf{f}_n^* = \mathbf{f}_n \mathbf{Y} \mathbf{f}_n^*$ where $\mathbf{X}, \mathbf{Y} \in \mathcal{R}_d(\mathbb{C})$ are distinct now yields an independent nontrivial *real* quadratic equation for the real variables $\mathrm{Re}(\mathbf{f}_n), \mathrm{Im}(\mathbf{f}_n)$. Each $\mathbf{X} = a\mathbf{f}\mathbf{g}^*$ with $a \in \mathbb{C}, \mathbf{f}, \mathbf{g} \in \mathbb{H}_+^d$ and $\|\mathbf{f}\| = \|\mathbf{g}\| = 1$ has $2 + (2d-2) + (2d-2) = 4d-2$ real degrees of freedom. The same holds for Y. Thus the projection of $\mathcal{A} = \{(\mathbf{F}, \mathbf{X}, \mathbf{Y})\}$ to the first component has real local dimension everywhere at most $2Nd - N + 2(4d - 2)$. Suppose that $N \geq 8d - 3$. Then this real local dimension has

$$2Nd - N + 2(4d - 2) \leq 2Nd - 1 < 2Nd.$$

In other words, a generic $F \in \mathbb{C}^{N \times d}$ is not a projection of an element in $\mathcal{A}$ to the first component. Thus for a generic $\mathcal{F}$ with $N \geq 8d - 3$ the map $\mathbf{M}_{\mathcal{F}}$ is injective on $\mathcal{R}_d(\mathbb{C})$.

Case 4 $\mathbb{H} = \mathbb{C}$ *and* $\mathbf{X}, \mathbf{Y} \in \mathcal{S}_d(\mathbb{C})$.

All arguments from Case 3 carry to this case, except in the counting of degrees of freedom for X and Y. Because now $\mathbf{X} = a\mathbf{f}\mathbf{f}^*$ where $a \in \mathbb{R}$, $\mathbf{f} \in \mathbb{H}_+^d$ and $\|\mathbf{f}\| = 1$ there are exactly $1 + 2d - 2 = 2d - 1$ real degrees of freedom for X. The same holds true for $\mathbf{Y}$. Thus the projection of $\mathcal{A} = \{(\mathbf{F}, \mathbf{X}, \mathbf{Y})\}$ to the first component has real local dimension everywhere at most $2Nd - N + 2(2d - 1)$. Suppose that $N \geq 4d - 1$. Then this local dimension has

$$Nd - N + 2(2d - 1) \leq Nd - 1 < Nd.$$

In other words, a generic $\mathbf{F} \in \mathbb{C}^{N \times d}$ is not a projection of an element in $\mathcal{A}$ to the first component. Thus for a generic $\mathcal{F}$ with $N \geq 4d - 1$ the map $\mathbf{M}_{\mathcal{F}}$ is injective on $\mathcal{S}_d(\mathbb{C})$. □

We can now reformulate the phase retrieval problem into two alternative optimization problems. Each rank-one matrix $\mathbf{X}$ can be written as $\mathbf{X} = \mathbf{x}\mathbf{y}^*$ for some $\mathbf{x}, \mathbf{y} \in \mathbb{H}^d$, although this representation is not unique. We have

Theorem 2.2 *Let $\mathcal{F} = \{\mathbf{f}_n\}_{n=1}^N$ be vectors in $\mathbb{H}^d$ such that $\mathbf{M}_\mathcal{F}$ is injective on $\mathcal{R}_d(\mathbb{H})$. Let $\mathbf{x}_0 \in \mathbb{H}^d(\mathbb{H})$ and $\mathbf{b} = \mathbf{M}_\mathcal{F}(\mathbf{x}_0\mathbf{x}_0^*) = [|\langle \mathbf{f}_1, \mathbf{x}_0\rangle|^2, \ldots, |\langle \mathbf{f}_N, \mathbf{x}_0\rangle|^2]^T$. Then any global minimizer*

$$(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = \operatorname{argmin}_{\mathbf{x},\mathbf{y}\in\mathbb{H}^d} \left\|\mathbf{M}_\mathcal{F}(\mathbf{x}\mathbf{y}^*) - \mathbf{b}\right\| \tag{2.4}$$

must satisfy $\hat{\mathbf{x}} = c\mathbf{x}_0$ and $\hat{\mathbf{y}} = \mathbf{x}_0/c$ with $c \neq 0$.

Proof The result follows trivially from the injectivity of $\mathbf{M}_\mathcal{F}(\mathbf{X})$ on $\mathcal{R}_d(\mathbb{H})$. Clearly if $\hat{\mathbf{x}} = c\mathbf{x}_0$ and $\hat{\mathbf{y}} = \mathbf{x}_0/c$ with $c \neq 0$, then $\hat{\mathbf{x}}\hat{\mathbf{y}}^* = \mathbf{x}_0\mathbf{x}_0^*$ which gives the global minimizer. Conversely, the global minimizer must have $\mathbf{M}_\mathcal{F}(\mathbf{x}\mathbf{y}^*) = \mathbf{b}$. The injectivity now implies that $\hat{\mathbf{x}}\hat{\mathbf{y}}^* = \mathbf{x}_0\mathbf{x}_0^*$. Thus $\hat{\mathbf{x}} = c\mathbf{x}_0$ and $\hat{\mathbf{y}} = \mathbf{x}_0/c$ for some $c \neq 0$. □

The above minimization problem is not convex so solving for the global minimum is very challenging. Such is the case for solving the phase retrieval problem in general. The advantage of the above formulation is that it allows us to use the popular alternating minimization technique used for many other applications such as low rank matrix completion, see e.g. [14,18,22,28] the references therein. In the alternating minimization algorithm, we first pick an initial $\mathbf{x}_1$ and minimize $\|\mathbf{M}_\mathcal{F}(\mathbf{x}_1\mathbf{y}^*) - \mathbf{b}\|$ with respect to $\mathbf{y}$ to obtain $\mathbf{y}_1$. This step is a standard ℓ_2-minimization and is linear problem. From $\mathbf{y}_1$ we then update $\mathbf{x}$ to $\mathbf{x}_2$ via minimizing $\|\mathbf{M}_\mathcal{F}(\mathbf{x}\mathbf{y}_1^*) - \mathbf{b}\|$. This process is iterated to yield a sequence $\mathbf{x}_k\mathbf{y}_k^*$. Often the sequence converges to the desired result.

The drawback of the above setup is that because there is no penalty for $\mathbf{x}$ and $\mathbf{y}$, when noise is added to the measurement vector $\mathbf{b}$, the stability and robustness is harder to analyze. It also requires, at least in theory, almost twice as many measurements as the minimally required number for phase retrieval. A better alternative is to replace $\mathbf{x} = \mathbf{y}$ by a least square quadratic penalty $\|\mathbf{x} - \mathbf{y}\|^2$ and add a regularization term to the previous minimization problem. Let $\lambda > 0$ and

$$E_{\lambda,\mathbf{b}}(\mathbf{x}, \mathbf{y}) = \left\|\mathbf{M}_\mathcal{F}(\mathbf{x}\mathbf{y}^*) - \mathbf{b}\right\|^2 + \lambda\|\mathbf{x} - \mathbf{y}\|^2. \tag{2.5}$$

Below we study the consequences of minimizing this function. In particular, we wish to establish certain robustness properties.

Lemma 2.3 *Let $\mathcal{F} = \{\mathbf{f}_n\}_{n=1}^N$ be a frame in $\mathbb{H}^d$. Let $\mathbf{X} \in M_d(\mathbb{H})$. Then $\|\mathbf{M}_\mathcal{F}(\mathbf{X})\|_1 \leq C\|\mathbf{X}\|_*$ where $\|\mathbf{X}\|_*$ denotes the nuclear norm of $\mathbf{X}$, and C is the upper frame bound of $\mathcal{F}$, i.e. C is the largest eigenvalue of $\mathbf{F}\mathbf{F}^*$ where $\mathbf{F} = [\mathbf{f}_1, \ldots, \mathbf{f}_N]$ is the frame matrix for $\mathcal{F}$. Furthermore, this C is optimal.*

Proof Assume that $\mathbf{X} = \mathbf{v}\mathbf{v}^*$ for some $\mathbf{v} \in \mathbb{H}^d$ and $\|\mathbf{v}\| = 1$. Then $\|\mathbf{X}\|_* = 1$

$$\|\mathbf{M}_\mathcal{F}(\mathbf{X})\|_1 = \sum_{n=1}^N |\langle \mathbf{f}_n, \mathbf{v}\rangle|^2 \leq C\|\mathbf{v}\|^2 = C.$$

Note that here this constant C is the best possible since it can be achieved by taking $\mathbf{v}$ to be an eigenvector of $\mathbf{F}\mathbf{F}^*$ corresponding to its largest eigenvalue. Now assume that $\mathbf{X}$ is Hermitian. Then we may write $\mathbf{X}$ as

$$\mathbf{X} = \sum_{j=1}^d \lambda_j \mathbf{v}_j \mathbf{v}_j^*$$

where $\{\mathbf{v}_j\}$ is an orthonormal basis for $\mathbb{H}^d$. Thus

$$\|\mathbf{M}_{\mathcal{F}}(\mathbf{X})\|_1 \le \sum_{j=1}^{d} |\lambda_j| \|\mathbf{M}_{\mathcal{F}}(\mathbf{v}_j\mathbf{v}_j^*)\|_1 \le C \sum_{j=1}^{d} |\lambda_j| = C\|\mathbf{X}\|_*.$$

For a non-Hermitian $\mathbf{X}$, let $\mathbf{Y} = \frac{1}{2}(\mathbf{X} + \mathbf{X}^*)$. Then $\|\mathbf{Y}\|_* \le \|\mathbf{X}\|_*$, and

$$\|\mathbf{M}_{\mathcal{F}}(\mathbf{X})\|_1 = \|\mathbf{M}_{\mathcal{F}}(\mathbf{Y})\|_1 \le C\|\mathbf{Y}\|_* \le C\|\mathbf{X}\|_*.$$

□

For a phase retrievable set $\mathcal{F} = \{\mathbf{f}_n\}_{n=1}^N$ in $\mathbb{H}^d$ we say $\mathbf{M}_{\mathcal{F}}$ satisfies the *c-stability condition* if for any $\mathbf{x}, \mathbf{y} \in \mathbb{H}^d$ we have

$$\left\|\mathbf{M}_{\mathcal{F}}(\mathbf{x}\mathbf{x}^*) - \mathbf{M}_{\mathcal{F}}(\mathbf{y}\mathbf{y}^*)\right\| \ge c\|\mathbf{x}\mathbf{x}^* - \mathbf{y}\mathbf{y}^*\|_*. \tag{2.6}$$

Theorem 2.4 *Let $\mathcal{F} = \{\mathbf{f}_n\}_{n=1}^N$ be phase retrievable in $\mathbb{H}^d$. Let $\mathbf{x}_0 \in \mathbb{H}^d(\mathbb{H})$ and $\mathbf{b} = \mathbf{M}_{\mathcal{F}}(\mathbf{x}_0\mathbf{x}_0^*) = [|\langle \mathbf{f}_1, \mathbf{x}_0\rangle|^2, \ldots, |\langle \mathbf{f}_N, \mathbf{x}_0\rangle|^2]^T$. Then*

(A) *Any global minimizer*

$$(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = \operatorname{argmin}_{\mathbf{x},\mathbf{y}\in\mathbb{H}^d} E_{\lambda,\mathbf{b}}(\mathbf{x}, \mathbf{y})$$

must satisfy $\hat{\mathbf{x}}\hat{\mathbf{y}}^* = \mathbf{x}_0\mathbf{x}_0^*$, or equivalently $\hat{\mathbf{x}} = \hat{\mathbf{y}} = c\mathbf{x}_0$ for some $|c| = 1$.

(B) *Let $\mathbf{b}' \in \mathbb{H}^d$ such that $\|\mathbf{b} - \mathbf{b}'\| \le \varepsilon$. Assume that $\mathbf{M}_{\mathcal{F}}$ satisfies the c-stability condition for some $c > 0$. Then any $\mathbf{x}, \mathbf{y} \in \mathbb{H}^d$ such that $E_{\lambda,\mathbf{b}'}(\mathbf{x}, \mathbf{y}) \le \delta^2$ must satisfy*

$$\left\|\mathbf{z}\mathbf{z}^* - \mathbf{x}_0\mathbf{x}_0^*\right\|_* \le \frac{1}{c}\Big(\frac{C}{4\lambda}\delta^2 + \delta + \varepsilon\Big), \tag{2.7}$$

where $\mathbf{z} = \frac{1}{2}(\mathbf{x} + \mathbf{y})$ and C is the upper frame bound of $\mathcal{F}$.

Proof Part (A) is rather straightforward. Note that $E_{\lambda,\mathbf{b}}(\mathbf{x}_0, \mathbf{x}_0) = 0$, so we must have $E_{\lambda,\mathbf{b}}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = 0$. It follows that $\hat{\mathbf{x}} = \hat{\mathbf{y}}$. Hence $\mathbf{M}_{\mathcal{F}}(\hat{\mathbf{x}}\hat{\mathbf{x}}^*) = \mathbf{b} = \mathbf{M}_{\mathcal{F}}(\mathbf{x}_0\mathbf{x}_0^*)$. The fact that $\mathcal{F}$ is phase retrievable now implies $\hat{\mathbf{x}}\hat{\mathbf{x}}^* = \mathbf{x}_0\mathbf{x}_0^*$.

To prove part (B), we have $\lambda\|\mathbf{x}-\mathbf{y}\|^2 \le \delta^2$. Thus $\|\mathbf{x}-\mathbf{y}\| \le \delta/\sqrt{\lambda}$. Let $\mathbf{Z} = \frac{1}{2}(\mathbf{x}\mathbf{y}^* + \mathbf{y}\mathbf{x}^*)$. Clearly $\mathbf{M}_{\mathcal{F}}(Z) = \mathbf{M}_{\mathcal{F}}(\mathbf{x}\mathbf{y}^*)$. Furthermore one checks easily that

$$\mathbf{z}\mathbf{z}^* - \mathbf{Z} = \frac{1}{4}(\mathbf{x} - \mathbf{y})(\mathbf{x} - \mathbf{y})^*.$$

Hence $\|\mathbf{z}\mathbf{z}^* - \mathbf{Z}\|_* = \frac{1}{4}\|\mathbf{x} - \mathbf{y}\|^2 \le \frac{\delta^2}{4\lambda}$. It follows that

$$\begin{aligned}
\|\mathbf{M}_{\mathcal{F}}(\mathbf{z}\mathbf{z}^*) - \mathbf{M}_{\mathcal{F}}(\mathbf{x}_0\mathbf{x}_0^*)\| &= \|\mathbf{M}_{\mathcal{F}}(\mathbf{x}\mathbf{x}^*) - \mathbf{b}\| \\
&\le \|\mathbf{M}_{\mathcal{F}}(\mathbf{z}\mathbf{z}^*) - \mathbf{M}_{\mathcal{F}}(\mathbf{Z})\| + \|\mathbf{M}_{\mathcal{F}}(\mathbf{Z}) - \mathbf{b}'\| + \|\mathbf{b}' - \mathbf{b}\| \\
&\le \|\mathbf{M}_{\mathcal{F}}(\mathbf{z}\mathbf{z}^*) - \mathbf{M}_{\mathcal{F}}(\mathbf{Z})\|_1 + \|\mathbf{M}_{\mathcal{F}}(\mathbf{Z}) - \mathbf{b}'\| + \|\mathbf{b}' - \mathbf{b}\| \\
&\le C\frac{\delta^2}{4\lambda} + \delta + \varepsilon.
\end{aligned}$$

The c-stability condition now implies (2.7) immediately. □

3 Alternating Minimization Algorithm

From the formulation in the previous section, the phase retrieval problem is solved robustly by finding a global minimizer of $E_{\lambda,\mathbf{b}}$ in (2.5). This section is devoted to fast algorithms for solving such a minimization problem. In Sect. 3.1, we introduce a fast alternating gradient descent algorithm for $\min_{\mathbf{x},\mathbf{y}} E_{\lambda,\mathbf{b}}(\mathbf{x},\mathbf{y})$. In Sect. 3.2, we prove the convergence of the proposed algorithm.

3.1 Alternating Gradient Descent Algorithm

Since λ and $\mathbf{b}$ are fixed during the minimization procedure, we drop the subscripts in $E_{\lambda,\mathbf{b}}$ for simplicity. That is, we solve

$$\arg\min_{\mathbf{x},\mathbf{y}} E(\mathbf{x},\mathbf{y}), \tag{3.1}$$

where

$$E(\mathbf{x},\mathbf{y}) = \frac{1}{N}\sum_{n=1}^{N} |\mathbf{x}^*\mathbf{f}_n\mathbf{f}_n^*\mathbf{y} - b_n|^2 + \lambda\|\mathbf{x}-\mathbf{y}\|^2. \tag{3.2}$$

Since the first term in $E(\mathbf{x},\mathbf{y})$ is quadratic in both $\mathbf{x}$ and $\mathbf{y}$, Eq. (3.1) is a non-convex optimization. However, when one of the variables $\mathbf{x}$ or $\mathbf{y}$ is fixed, $E(\mathbf{x},\mathbf{y})$ is quadratic with respect to the other variable. Therefore, it is natural to solve (3.1) by an alternating scheme.

We use the following alternating gradient descent algorithm: Fixing $\mathbf{x}$, we minimize $E(\mathbf{x},\mathbf{y})$ with respect to $\mathbf{y}$ by one step of gradient descent, and vice versa. More precisely, we define, for $k = 0, 1, 2, \ldots$,

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k), \\ \mathbf{y}_{k+1} = \mathbf{y}_k - \beta_k \nabla_{\mathbf{y}} E(\mathbf{x}_{k+1}, \mathbf{y}_k), \end{cases} \tag{3.3}$$

where α_k and β_k are step sizes. Since E is a real-valued function with complex variables, the gradients $\nabla_{\mathbf{x}} E$ and $\nabla_{\mathbf{y}} E$ in (3.3) are in the sense of Wirtinger gradient [8].

Since the gradient descent is applied to only one of the variables $\mathbf{x}$ and $\mathbf{y}$, the corresponding Hessian matrix has a much smaller norm than the Hessian of E with respect to $(\mathbf{x},\mathbf{y})$. Consequently, a much larger step size is allowed in the alternating gradient descent than the standard gradient descent for minimizing (3.2). This leads to a faster convergence. The alternating gradient descent algorithm is also faster than the Wirtinger flow (WF) [8] algorithm, where $G(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N}(|\mathbf{f}_n^*\mathbf{x}|^2 - b_n)^2$ is minimized via a gradient flow. As explained in Appendix A, in the real case, when the iterates $\mathbf{x}$ and $\mathbf{y}$ are sufficiently close, our proposed alternating gradient descent algorithm is 1.5 times faster than the WF algorithm in terms of the decreasing of the objective function value.

The initialization of our proposed algorithm is obtained via a spectral method, which is the same as that in the Wirtinger flow algorithm [8]. When $\mathbf{f}_n$, $n = 1, \ldots, N$, follow certain probability distributions (e.g., Gaussian), the expectation of $\mathbf{Y} = \frac{1}{N}\sum_{n=1}^{N} b_n\mathbf{f}_n\mathbf{f}_n^*$ has a leading principal eigenvector $\tilde{\mathbf{x}}$. Therefore, we choose $\mathbf{x}_0 = \mathbf{y}_0 = \mathbf{z}_0$, where $\mathbf{z}_0$ is the leading principal eigenvector of $\mathbf{Y}$. For completeness, Algorithm 1 lists how to calculate the initial guess for our proposed algorithm.

Algorithm 1: Initialization.

Input: Observations $b_n, n = 1, \cdots, N$.

Output: Initial guess $\mathbf{x}_0 = \mathbf{y}_0 = \mathbf{z}_0$.

1 Set

$$\theta^2 = d\frac{\sum_n b_n}{\sum_n \|\mathbf{f}_n\|^2},$$

where $\mathbf{f}_n \in \mathbb{C}^d, n = 1, \cdots, N$ is the sampling vectors;

2 $\mathbf{z}_0$ is the eigenvector corresponding to the largest eigenvalue of

$$\mathbf{Y} = \frac{1}{N}\sum_{n=1}^{N} b_n \mathbf{f}_n \mathbf{f}_n^*$$

and $\|\mathbf{z}_0\| = \theta$.

3.2 Convergence

In this section, we will show the convergence of the alternating gradient descent algorithm (3.3). More precisely, for any initial guess, we prove that algorithm (3.3) converges to a critical point of E.

We first present a lemma, which shows the coercivity of E.

Lemma 3.1 *If* $\mathbf{F}$ *is of full rank, i.e.,* $\operatorname{rank}(\mathbf{F}) = d$*, then the function* $E(\mathbf{x}, \mathbf{y})$ *is coercive, i.e.,* $E(\mathbf{x}, \mathbf{y}) \to \infty$ *as* $\|(\mathbf{x}, \mathbf{y})\| \to \infty$.

Proof Since $\mathbf{F}$ is of full row rank, there exists a constant C_1 such that, for any $\mathbf{x}$,

$$\||\mathbf{F}^*\mathbf{x}|^2\| = \|\mathbf{F}^*\mathbf{x}\|_4^2 \geq C_0\|\mathbf{F}^*\mathbf{x}\|^2 \geq C_1\|\mathbf{x}\|^2.$$

Also, there exists a constant C_2 such that, for any $\mathbf{x}$ and $\mathbf{z}$,

$$\|(\overline{\mathbf{F}^*\mathbf{x}}) \circ (\mathbf{F}^*\mathbf{z})\| \leq \|F^*\mathbf{x}\|\|\mathbf{F}^*\mathbf{z}\| \leq C_2\|\mathbf{x}\|\|\mathbf{z}\|,$$

where $\circ$ is the componentwise product.

Let $\|(\mathbf{x}, \mathbf{y})\| = M$. If $\|\mathbf{x}\| \leq \frac{M}{2}$, then $\|\mathbf{y}\| \geq \frac{\sqrt{3}M}{2}$ and

$$E(\mathbf{x}, \mathbf{y}) \geq \lambda\|\mathbf{x} - \mathbf{y}\|^2 \geq \frac{\lambda(\sqrt{3}-1)^2 M}{4}.$$

Similarly, if $\|\mathbf{y}\|_2 \leq \frac{M}{2}$, then $E(\mathbf{x}, \mathbf{y}) \geq \frac{\lambda(\sqrt{3}-1)^2 M}{4}$. Otherwise, both $\|\mathbf{x}\|_2 > \frac{M}{2}$ and $\|\mathbf{y}\|_2 > \frac{M}{2}$. Define $\mathbf{z} = \mathbf{y} - \mathbf{x}$. In this case, if $\|\mathbf{z}\| \leq \frac{C_1 M}{4C_2}$, then

$$\begin{aligned}
E(\mathbf{x}, \mathbf{y}) &\geq \frac{1}{N}\sum_{n=1}^{N} |\mathbf{x}^*\mathbf{f}_n\mathbf{f}_n^*\mathbf{y} - b_n|^2 = \frac{1}{N}\|(\overline{F^*\mathbf{x}}) \circ (F^*(\mathbf{x}+\mathbf{z})) - \mathbf{b}\|^2 \\
&\geq \frac{1}{N}(\||\mathbf{F}^*\mathbf{x}|^2\| - \|\mathbf{b}\| - \|(\overline{\mathbf{F}^*\mathbf{x}}) \circ (\mathbf{F}^*\mathbf{z})\|)^2 \\
&\geq \frac{1}{N}(C_1\|\mathbf{x}\|^2 - \|\mathbf{b}\| - C_2\|\mathbf{x}\|\|\mathbf{z}\|)^2 \\
&= \frac{1}{N}(\|\mathbf{x}\|(C_1\|\mathbf{x}\| - C_2\|\mathbf{z}\|) - \|\mathbf{b}\|)^2 \geq \frac{1}{N}(M^2/8 - \|\mathbf{b}\|)^2,
\end{aligned}$$

otherwise $E(\mathbf{x},\mathbf{y}) \geq \lambda\|\mathbf{z}\|^2 \geq \frac{\lambda C_1^2 M^2}{16C_2^2}$. In all the cases, as $M \to \infty$, the lower bounds approach infinity. □

Now we can prove the convergence of (3.3).

Theorem 3.2 *Assume* $\mathrm{rank}(\mathbf{F}) = d$. *Then, for any initial guess* $(\mathbf{x}_0,\mathbf{y}_0)$, *the sequence* $\{(\mathbf{x}_k,\mathbf{y}_k)\}_k$ *generated by* (3.3) *with a suitable step size converges to a critical point of* E.

Proof For simplicity, we assume $\alpha_k = \beta_k = \gamma$ for all k. The proof with variant step sizes can be done similarly. Since $E(\mathbf{x},\mathbf{y})$ is a quadratic function with respect to $\mathbf{x}$, Taylor's expansion gives

$$\begin{aligned}
&E(\mathbf{x}_{k+1},\mathbf{y}_k)\\
&= E(\mathbf{x}_k,\mathbf{y}_k) + \begin{bmatrix}\mathbf{x}_{k+1}-\mathbf{x}_k\\ \overline{\mathbf{x}}_{k+1}-\overline{\mathbf{x}}_k\end{bmatrix}^* \begin{bmatrix}\nabla_{\mathbf{x}}E(\mathbf{x}_k,\mathbf{y}_k)\\ \nabla_{\overline{\mathbf{x}}}E(\mathbf{x}_k,\mathbf{y}_k)\end{bmatrix}\\
&\quad + \frac{1}{2}\begin{bmatrix}\mathbf{x}_{k+1}-\mathbf{x}_k\\ \overline{\mathbf{x}}_{k+1}-\overline{\mathbf{x}}_k\end{bmatrix}^* \begin{bmatrix}\nabla^2_{\mathbf{xx}}E(\mathbf{x}_k,\mathbf{y}_k) & \nabla^2_{\mathbf{x}\overline{\mathbf{x}}}E(\mathbf{x}_k,\mathbf{y}_k)\\ \nabla^2_{\overline{\mathbf{x}}\mathbf{x}}E(\mathbf{x}_k,\mathbf{y}_k) & \nabla^2_{\overline{\mathbf{x}}\overline{\mathbf{x}}}E(\mathbf{x}_k,\mathbf{y}_k)\end{bmatrix}\begin{bmatrix}\mathbf{x}_{k+1}-\mathbf{x}_k\\ \overline{\mathbf{x}}_{k+1}-\overline{\mathbf{x}}_k\end{bmatrix} \qquad (3.4)\\
&= E(\mathbf{x}_k,\mathbf{y}_k) - 2\gamma\|\nabla_{\mathbf{x}}E(\mathbf{x}_k,\mathbf{y}_k)\|^2 + \gamma^2\nabla_{\mathbf{x}}E(\mathbf{x}_k,\mathbf{y}_k)^*\nabla^2_{\mathbf{xx}}E(\mathbf{x}_k,\mathbf{y}_k)\nabla_{\mathbf{x}}E(\mathbf{x}_k,\mathbf{y}_k)\\
&= E(\mathbf{x}_k,\mathbf{y}_k) - 2\gamma\left(\|\nabla_{\mathbf{x}}E(\mathbf{x}_k,\mathbf{y}_k)\|^2 - \frac{\gamma}{2}\nabla_{\mathbf{x}}E(\mathbf{x}_k,\mathbf{y}_k)^*\nabla^2_{\mathbf{xx}}E(\mathbf{x}_k,\mathbf{y}_k)\nabla_{\mathbf{x}}E(\mathbf{x}_k,\mathbf{y}_k)\right).
\end{aligned}$$

Similarly, because $E(\mathbf{x},\mathbf{y})$ is a quadratic function with respect to $\mathbf{y}$, by Taylor's expansion, we obtain

$$\begin{aligned}
E(\mathbf{x}_{k+1},\mathbf{y}_{k+1}) = E(\mathbf{x}_{k+1},\mathbf{y}_k) - 2\gamma\Big(&\|\nabla_{\mathbf{y}}E(\mathbf{x}_{k+1},\mathbf{y}_k)\|^2\\
&-\frac{\gamma}{2}\nabla_{\mathbf{y}}E(\mathbf{x}_{k+1},\mathbf{y}_k)^*\nabla^2_{\mathbf{yy}}E(\mathbf{x}_{k+1},\mathbf{y}_k)\nabla_{\mathbf{y}}E(\mathbf{x}_{k+1},\mathbf{y}_k)\Big). \qquad (3.5)
\end{aligned}$$

By Lemma 3.1, the level set $\mathcal{S} = \{(\mathbf{x},\mathbf{y}) : E(\mathbf{x},\mathbf{y}) \leq E(\mathbf{x}_0,\mathbf{y}_0)\}$ is a bounded closed set. Therefore, the continuous functions $\nabla^2_{\mathbf{xx}}E(\mathbf{x},\mathbf{y})$ and $\nabla^2_{\mathbf{yy}}E(\mathbf{x},\mathbf{y})$ are bounded on $\mathcal{S}$. Let $M > 0$ be the bound, i.e.,

$$\|\nabla^2_{\mathbf{xx}}E(\mathbf{x},\mathbf{y})\| \leq M, \quad \|\nabla^2_{\mathbf{yy}}E(\mathbf{x},\mathbf{y})\| \leq M, \quad \forall\,(\mathbf{x},\mathbf{y}) \in \mathcal{S}.$$

Suppose $(\mathbf{x}_k,\mathbf{y}_k) \in \mathcal{S}$. Choose $\gamma \in (0, 2/M)$, so that (3.4) and (3.5) implies that $E(\mathbf{x}_{k+1},\mathbf{y}_{k+1}) \leq E(\mathbf{x}_k,\mathbf{y}_k)$ and

$$\begin{aligned}
&E(\mathbf{x}_k,\mathbf{y}_k) - E(\mathbf{x}_{k+1},\mathbf{y}_{k+1})\\
&= E(\mathbf{x}_k,\mathbf{y}_k) - E(\mathbf{x}_{k+1},\mathbf{y}_k) + E(\mathbf{x}_{k+1},\mathbf{y}_k) - E(\mathbf{x}_{k+1},\mathbf{y}_{k+1})\\
&\geq \zeta(\|\nabla_{\mathbf{x}}E(\mathbf{x}_k,\mathbf{y}_k)\|^2 + \|\nabla_{\mathbf{y}}E(\mathbf{x}_{k+1},\mathbf{y}_k)\|^2)\\
&= \frac{\zeta}{\gamma}(\|\mathbf{x}_{k+1}-\mathbf{x}_k\|^2 + \|\mathbf{y}_{k+1}-\mathbf{y}_k\|^2) \qquad (3.6)
\end{aligned}$$

with $\zeta = 2\gamma\left(1 - \frac{\gamma M}{2}\right)$. Therefore, $(\mathbf{x}_{k+1},\mathbf{y}_{k+1}) \in \mathcal{S}$. Thus, by induction, $(\mathbf{x}_k,\mathbf{y}_k) \in \mathcal{S}$ and (3.6) hold for all k as long as $\gamma \in (0, 2/M)$.

Summing (3.6) over k from 0 to $+\infty$, we obtain

$$E(\mathbf{x}_0,\mathbf{y}_0) - \lim_{k\to+\infty} E(\mathbf{x}_k,\mathbf{y}_k) \geq \zeta\sum_{k=0}^{+\infty}(\|\nabla_{\mathbf{x}}E(\mathbf{x}_k,\mathbf{y}_k)\|^2 + \|\nabla_{\mathbf{y}}E(\mathbf{x}_{k+1},\mathbf{y}_k)\|^2).$$

Because $E(\mathbf{x}_k, \mathbf{y}_k) \geq 0$ is monotonically nonincreasing according to (3.6), its limit exists and is finite, which implies

$$\begin{aligned}\lim_{k\to+\infty} \|\nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k)\| &= \lim_{k\to+\infty} \|\mathbf{x}_{k+1} - \mathbf{x}_k\| = 0,\\ \lim_{k\to+\infty} \|\nabla_{\mathbf{y}} E(\mathbf{x}_{k+1}, \mathbf{y}_k)\| &= \lim_{k\to+\infty} \|\mathbf{y}_{k+1} - \mathbf{y}_k\| = 0.\end{aligned}$$

This, together with the continuity of $\nabla_{\mathbf{x}} E$, $\nabla_{\mathbf{y}} E$, and the norm function, means that any clustering point of $\{(\mathbf{x}_k, \mathbf{y}_k)\}_k$ is a critical point of $E(\mathbf{x}, \mathbf{y})$.

It remains to prove that $\{(\mathbf{x}_k, \mathbf{y}_k)\}_k$ is convergent, which is done by checking that $\{(\mathbf{x}_k, \mathbf{y}_k)\}_k$ is a Cauchy sequence. Since $E(\mathbf{x}, \mathbf{y})$ is a real-valued polynomial function, it belongs to a semi-algebraic set. By [5, Theorem 3], there exists a differentiable and concave function $\psi(t)$ such that

$$\psi'(E(\mathbf{x}_k, \mathbf{y}_k) - E(\hat{\mathbf{x}}, \hat{\mathbf{y}})) \cdot \left\| \begin{bmatrix} \nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k) \\ \nabla_{\mathbf{y}} E(\mathbf{x}_k, \mathbf{y}_k) \end{bmatrix} \right\| \geq 1 \tag{3.7}$$

for any k and for any critical point $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ of E. Since $\psi(t)$ is concave, by the inequalities (3.6) and (3.7), we have

$$\begin{aligned}&\psi(E(\mathbf{x}_k, \mathbf{y}_k) - E(\hat{\mathbf{x}}, \hat{\mathbf{y}})) - \psi(E(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}) - E(\hat{\mathbf{x}}, \hat{\mathbf{y}}))\\ &\quad\geq \psi'(E(\mathbf{x}_k, \mathbf{y}_k) - E(\hat{\mathbf{x}}, \hat{\mathbf{y}}))(E(\mathbf{x}_k, \mathbf{y}_k) - E(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}))\\ &\quad\geq \frac{\zeta}{\gamma} \frac{\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + \|\mathbf{y}_{k+1} - \mathbf{y}_k\|^2}{\sqrt{\|\nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k)\|^2 + \|\nabla_{\mathbf{y}} E(\mathbf{x}_k, \mathbf{y}_k)\|^2}}.\end{aligned} \tag{3.8}$$

Furthermore,

$$\|\nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k)\| = \frac{1}{\gamma} \|(\mathbf{x}_{k+1} - \mathbf{x}_k)\| \tag{3.9}$$

and

$$\begin{aligned}\|\nabla_{\mathbf{y}} E(\mathbf{x}_k, \mathbf{y}_k)\| &= \left\| \nabla_{\mathbf{y}} E(\mathbf{x}_k, \mathbf{y}_k) - \nabla_{\mathbf{y}} E(\mathbf{x}_{k+1}, \mathbf{y}_k) + \frac{1}{\gamma}(\mathbf{y}_{k+1} - \mathbf{y}_k) \right\|\\ &\leq \left\| \nabla_{\mathbf{y}} E(\mathbf{x}_k, \mathbf{y}_k) - \nabla_{\mathbf{y}} E(\mathbf{x}_{k+1}, \mathbf{y}_k) \right\| + \frac{1}{\gamma} \|\mathbf{y}_{k+1} - \mathbf{y}_k\|\\ &\leq M' \|\mathbf{x}_{k+1} - \mathbf{x}_k\| + \frac{1}{\gamma} \|\mathbf{y}_{k+1} - \mathbf{y}_k\|,\end{aligned} \tag{3.10}$$

where $M' = \sup_{(\mathbf{x},\mathbf{y})\in\mathcal{S}} \|\nabla^2_{\mathbf{xy}} E(\mathbf{x}, \mathbf{y})\|$ that is finite. Plugging (3.9) and (3.10) into (3.8) gives

$$\begin{aligned}&\psi(E(\mathbf{x}_k, \mathbf{y}_k) - E(\hat{\mathbf{x}}, \hat{\mathbf{y}})) - \psi(E(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}) - E(\hat{\mathbf{x}}, \hat{\mathbf{y}}))\\ &\quad\geq C \left(\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + \|\mathbf{y}_{k+1} - \mathbf{y}_k\|^2\right)^{1/2},\end{aligned}$$

where $C = \frac{\zeta}{\gamma M' + 1}$. Summing it over k, we get

$$\begin{aligned}&\sum_{k=0}^{+\infty} \left(\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + \|\mathbf{y}_{k+1} - \mathbf{y}_k\|^2\right)^{1/2}\\ &\quad\leq \frac{1}{C} \left(\psi(E(\mathbf{x}_0, \mathbf{y}_0) - E(\hat{\mathbf{x}}, \hat{\mathbf{y}})) - \lim_{k\to\infty} \psi(E(\mathbf{x}_k, \mathbf{y}_k) - E(\hat{\mathbf{x}}, \hat{\mathbf{y}})) \right).\end{aligned}$$

The right hand side is finite, as ψ is smooth and $\lim_{k\to\infty} E(\mathbf{x}_k, \mathbf{y}_k)$ is finite. This verifies that $\{(\mathbf{x}_k, \mathbf{y}_k)\}_k$ is a Cauchy sequence, and therefore it is convergent. □

4 Numerical Implementation

In this section, we present some numerical experiments to evaluate the proposed alternating gradient descent algorithm and compare it with some others [8,11–13,20,25–27]. As demonstrated in Sect. 4.2 on synthetic data and Sect. 4.3 on real image data, our proposed algorithm is more efficient in the sense that a smaller number of iterations are required to achieve the same recovery accuracy.

4.1 Experiment Setup

The initialization of our proposed algorithm is described in Algorithm 1, which is done by 50 iterations of the power method. In WF algorithm [8], the step size in τth iteration is chosen heuristically and experimentally as $\tilde{\mu}_\tau = \min(1 - e^{-\tau/\tilde{\tau}_0}, \tilde{\mu}_{max})$, with $\tilde{\tau}_0 = 330$ and $\tilde{\mu}_{max} = 0.2$ or 0.4. This choice of step size is the most efficient according to our test. Parameter λ is a trade-off between the fitting to the equations and the penalty of distance of $\mathbf{u}$ and $\mathbf{v}$. Theoretically, as long as $\lambda > 0$, the global minimizer of (3.2) is $\mathbf{x} = \mathbf{y} = \hat{\mathbf{x}}$, where $\hat{\mathbf{x}}$ is the underlying true solution. In other words, any choice of $\lambda > 0$ will give the true solution. However, the choice of λ will affect the speed of our algorithm. Larger λ will keep $\mathbf{x}$ and $\mathbf{y}$ closer, and smaller λ will make the fitting error smaller. In the early stage of our algorithm, we want to decrease the fitting error with $\mathbf{x}$ and $\mathbf{y}$ close. So we choose a relatively larger λ. As the iteration goes on, both $\mathbf{x}$ and $\mathbf{y}$ are close to the true solution, and we want the fitting error to decrease faster. Thus, a smaller λ is chosen. Following this principle, we choose an exponentially decay λ in our experiments. Thus, the step size of our method is also chosen in the form as $\mu_\tau = \min(1 - e^{-\tau/\tau_0}, \mu_{max})$ and the tuning parameter $\lambda_\tau = \lambda_0 e^{-\xi\tau}$. The parameters $\tilde{\tau}_0$, τ_0, $\tilde{\mu}_{max}$ μ_{max}, λ_0 and ξ will be specified later.

Throughout the test, we consider complex signals only and mainly focus on the Gaussian model and the coded diffraction (CDF) model. In the Gaussian model, we collect the data $b_n = |\mathbf{f}_n^*\mathbf{x}|^2$ with the sampling vectors distributed as Gaussian model, that is,

$$\mathbf{f}_n \overset{\text{i.i.d.}}{\sim} \begin{cases} \mathcal{N}(0, \mathbf{I}/2) + i\mathcal{N}(0, \mathbf{I}/2), & \text{if } \mathbf{f}_n \in \mathbb{C}^d, \\ \mathcal{N}(0, \mathbf{I}), & \text{if } \mathbf{f}_n \in \mathbb{R}^d, \end{cases}$$

where $\mathcal{N}(0, \mathbf{V})$ is the real mean-zero Gaussian distribution with covariance matrix $\mathbf{V}$. In the CDF model, we acquire the data via

$$b_{p,q} = |\mathbf{x}^*\mathbf{a}_{p,q}|^2, \quad \text{with } 0 \le q \le d-1,\ 1 \le p \le L,$$

with $\mathbf{a}_{p,q} = \mathbf{G}_p\mathbf{f}_q$, where $\mathbf{f}_q^*$ is the qth row of the $d \times d$ discrete Fourier Transform (DFT) matrix and $\mathbf{G}_p$ is a diagonal matrix with i.i.d. diagonal entries $g_p(0)$, $g_p(1)$, …, $g_p(d-1)$ randomly drawn from $\left\{\pm\frac{\sqrt{2}}{2},\ \pm\frac{\sqrt{2}}{2}i\right\}$ with probability $\frac{1}{5}$ for each element, and $\left\{\pm\sqrt{3},\ \pm\sqrt{3}i\right\}$ with probability $\frac{1}{20}$ for each element.

4.2 Synthetic Data

In this subsection, we test the algorithms on synthetic data. Following [8], we are interested in the two signals described below:

– *Random low-pass signals* The true signal $\tilde{\mathbf{x}} \in \mathbb{C}^d$ is generated by

$$\tilde{x}[t] = \sum_{k=-(M/2-1)}^{M/2} (r_k + i j_k) e^{2\pi i (k-1)(t-1)/d}$$

where $M = \frac{d}{8}$, and r_k and j_k are i.i.d. obeying the standard normal distribution.

– *Random Gaussian signals* The true signal $\tilde{\mathbf{x}} \in \mathbb{C}^d$ is a random complex Gaussian vector with i.i.d. entries of the form

$$\tilde{x}[t] = \sum_{k=-(d/2-1)}^{d/2} (r_k + i j_k) e^{2\pi i (k-1)(t-1)/d},$$

where r_k and j_k are i.i.d. normal distribution $\mathcal{N}(0, \frac{1}{8})$.

We first evaluate the effectiveness of our proposed algorithm in terms of the smallest N required for successful phase retrieval. We use 100 trials for both the Gaussian and CDF models. In each trial, we generate the random sampling vectors according to the Gaussian or CDF model and stop the alternating iteration after 2500 iterations (1250 iterations for $\mathbf{x}$ and 1250 iterations for $\mathbf{y}$ corresponding to our method). We declare it is successful if the relative error of the construction $\text{dist}(\tilde{\mathbf{x}}, \hat{\mathbf{x}})/\|\tilde{\mathbf{x}}\| < 10^{-5}$, where $\hat{\mathbf{x}}$ is the numerical solution by our alternating minimization algorithm. The empirical probability of success is defined as the average of success over 100 trials. We use $d = 1000$. In the Gaussian model, we choose $\tau_0 = \tilde{\tau}_0 = 330$, $\tilde{\mu}_{max} = 0.2$, $\mu_{max} = 0.4$, $\lambda_0 = 30$ and $\xi = 0.15/330$ for random Gaussian signal, and $\tau_0 = \tilde{\tau}_0 = 330$, $\tilde{\mu}_{max} = 0.2$, $\mu_{max} = 0.4$, $\lambda_0 = 5$ and $\xi = 0.05/300$ for the random low-pass signal. In the CDF model, we choose $\tau_0 = \tilde{\tau}_0 = 330$, $\tilde{\mu}_{max} = 0.2$, $\mu_{max} = 0.4$, $\lambda_0 = 0.2$ and $\xi = 0.0015/330$ for random Gaussian signal, and $\tau_0 = \tilde{\tau}_0 = 330$, $\tilde{\mu}_{max} = 0.2$, $\mu_{max} = 0.4$, $\lambda_0 = 0.05$ and $\xi = 1.5/330$ for the random low-pass signal. We plot the empirical probability of success against the over sampling ratio N/d in Fig. 1. We see that the minimum oversampling ratios for an almost 100% successful phase retrieval by our algorithm are around 4.3 for the Gaussian model and 6 for the CDF model, which is slightly better or the same as the requirement of the WF algorithm as reported in [8].

Next, we demonstrate the efficiency of our proposed algorithm in terms of computational time. We compare our proposed algorithm with WF [8], AltMin [20], Kacz [27] and composite optimization [12]. We use the complex Gaussian measurement model with $N = 4.5d$. Figure 2a gives the plots of relative errors versus running time. We see that our proposed algorithm is faster than all the other algorithms except AltMin. However, AltMin needs to solve a least squares problem of coefficient matrix $\mathbf{F}^T$ at each iteration. We implemented it by first computing the pseudo inverse of $\mathbf{F}^T$ and then multiplying directly the pseudo inverse to get a least square solution at each iteration. Though it works very well in our setting $d = 1000$, it is not scalable as the precomputing of the pseudo inverse is very time and memory consuming. Moreover, we observe that our proposed algorithm can give 10^{-15} relative error for $\mathbf{x}$ and $\mathbf{y}$ after about 135 s, but WF needs more than 200 s. This implies our algorithm needs less computational time than the Wirtinger flow algorithm to get the same relative error.

Our proposed splitting scheme can be easily adapted to other cost functions fitting to the nonlinear equations $|\mathbf{f}_n^* \mathbf{x}|^2 = b_n$, $n = 1, \ldots, N$. For example, we can split the variables in the Poisson log likelihood error as follows

$$E(\mathbf{x}, \mathbf{y}) = -\sum_{n=1}^{N} b_n \log(|\mathbf{f}_n^* \mathbf{x}||\mathbf{f}_n^* \mathbf{y}|) - |\mathbf{x}^* \mathbf{f}_n \mathbf{f}_n^* \mathbf{y}| + \lambda \|\mathbf{x} - \mathbf{y}\|^2, \tag{4.1}$$

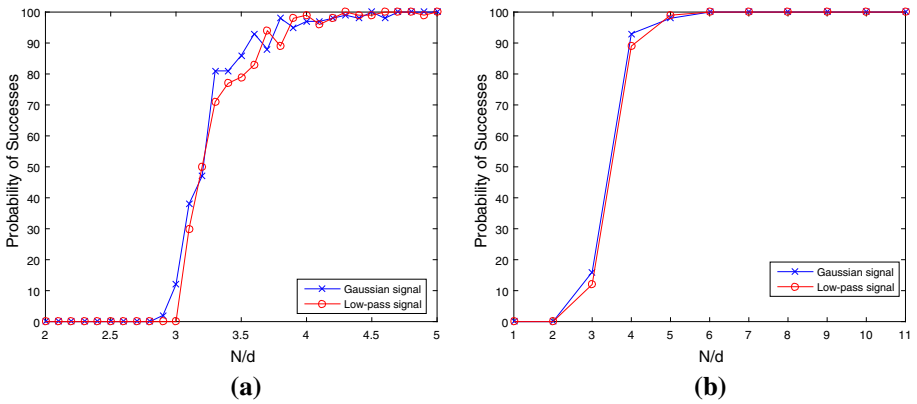

Fig. 1 The plot of the probability of success versus N/d. **a** Gaussian model and **b** coded diffraction model

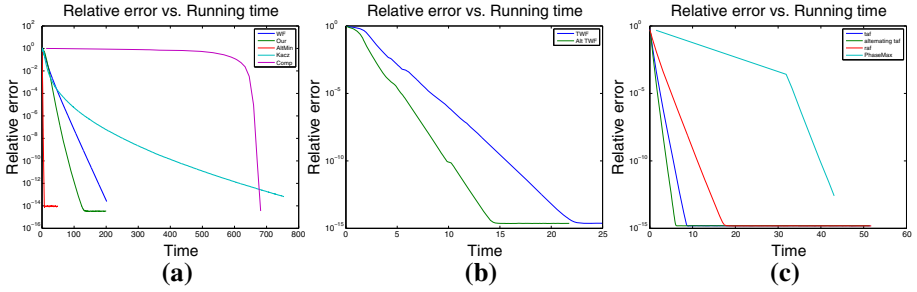

Fig. 2 The plot of the relative error versus the running time

and an alternating gradient descent algorithm can be applied for the numerical solution. We compare the resulting algorithm with truncated Wirtinger flow (TWF) [11], which minimizes directly the Poisson log likelihood error by a truncated gradient descent. Figure 2b illustrates the plots of relative errors versus running time of these two algorithms. By splitting variables, our method is faster than TWF for Poisson model. Moreover, our splitting scheme can be applied to the amplitude-based cost function (1.7) to get

$$E(\mathbf{x},\mathbf{y}) = \frac{1}{2N}\sum_{n=1}^{N}\left(\sqrt{|\mathbf{x}^*\mathbf{f}_n|}\sqrt{|\mathbf{f}_n^*\mathbf{y}|} - \sqrt{b_n}\right)^2 + \lambda\|\mathbf{x}-\mathbf{y}\|^2. \tag{4.2}$$

Again, an alternating gradient descent algorithm is applied to solve (4.2). We compare this algorithm with other algorithms that solve the amplitude equation, including the truncated amplitude flow (TAF) [26] and reweighted amplitude flow (RAF) [25]. We also compare it with FISTA algorithm to solve the PhaseMax [13] model, as it works on the relaxation of the amplitude equation. Since PhaseMax needs at least about $7d$ measurements, we choose $N = 7d$ in this comparison. Figure 2c gives the relative errors and running time for the four algorithms. We see our algorithm converges the fastest.

Table 1 The relative errors

Method	100	125	150
The Naqsh-e Jahan Square. ($L = 15$, $\tilde{\tau}_0 = 330$, $\tilde{\tau}_0 = 150$, $\tilde{\mu}_{max} = 0.4$, $\mu_{max} = 1$, $\lambda_0 = 8000$ and $\xi = 0.001$.)			
WF	3.7097×10^{-4}	6.1209×10^{-7}	1.2521×10^{-9}
Our	$\mathbf{1.4927 \times 10^{-4}}$	$\mathbf{5.9163 \times 10^{-8}}$	$\mathbf{1.2817 \times 10^{-11}}$
The Stanford main quad. ($L = 15$, $\tau_0 = 330$, $\tau_1 = 150$, $\tilde{\mu}_{max} = 0.4$, $\mu_{max} = 1$, $\lambda_0 = 8000$ and $\xi = 0.001$.)			
WF	0.5608	9.9557×10^{-4}	1.4987×10^{-6}
Our	**0.5310**	$\mathbf{1.4925 \times 10^{-4}}$	$\mathbf{3.6023 \times 10^{-8}}$
The van Gogh's painting $f458$. ($L = 15$, $\tilde{\tau}_0 = 330$, $\tau_0 = 100$, $\tilde{\mu}_{max} = 0.4$, $\mu_{max} = 0.5$, $\lambda_0 = 5000$, $\xi = 0.0015$.)			
WF	0.2178	0.0028	3.2730×10^{-6}
Our	$\mathbf{7.7887 \times 10^{-4}}$	$\mathbf{1.6263 \times 10^{-6}}$	$\mathbf{2.6466 \times 10^{-8}}$

The values bolded are the best results for the tests

Fig. 3 The recovered images for Naqsh-e Jahan Square, Esfahan

4.3 Real Image Data

We also test our algorithm for the CDF model on three real images in different sizes, namely, the Naqsh-e Jahan Square in the central Iranian city of Esfahan ($189 \times 768 \times 3$), the Stanford main quad ($320 \times 1280 \times 3$), and the van Gogh's painting $f458$ ($281 \times 369 \times 3$). Since those are all color images, we run our proposed algorithm and WF algorithm on each of the RGB channels. Let $\mathbf{x}$ denote the underlying true image and $\hat{\mathbf{x}}$ the solution by the algorithms. The relative error is defined as $\|\hat{\mathbf{x}} - \mathbf{x}\|/\|\mathbf{x}\|$ with $\|\mathbf{x}\|^2 = \sum_{i,j,k} |x_{ijk}|^2$. Table 1 lists the relative errors for WF with $2n$ iterations and our method with n iterations for $\mathbf{x}$ and n iteration for $\mathbf{y}$ with $n = 100, 125, 150$. From the results in the table, we see that our proposed algorithm use less iterations than WF method to achieve the same relative error. In Figs. 3, 4 and 5, the recoveries for the three real images are illustrated after 150 iterations for $\mathbf{x}$ and 150 iterations for $\mathbf{y}$.

5 Conclusion

In this paper, we introduce a fast rank-one alternating minimization method for phase retrieval. We split variables in (1.6) and solve a bi-variate optimization problem (3.1) which is quadratic

Fig. 4 The recovered images for the stanford image

Fig. 5 The recovered images for the van Gogh painting $f458$

in each of the variables. We use an alternating gradient descent algorithm as the numerical solver and give its convergence for any given initialization. Since a larger step size is allowed, the alternating gradient descent algorithm converges faster than WF method. Numerical results show that our method outperforms some existing methods, e.g., WF and AltMinPhase.

For future work, it is interesting to study the performance guarantee of our alternating gradient descent algorithm and the geometric landscape of the objective function in (3.1). Moreover, our strategy of splitting the variables can be applied equally to other non-convex optimizations such as (1.5) and (1.6). Preliminary numerical test suggests that this leads to more efficient phase retrieval algorithms. We will investigate thoroughly the empirical and theoretical performance of these splitting schemes in the future.

Acknowledgements The authors would like to thank Emmanuel Candès, Mo Mu and Aditya Viswanathan for very helpful discussions.

A Larger step size

In this appendix, we demonstrate that, when $\mathbf{x}$ and $\mathbf{y}$ are sufficiently close, our alternating gradient descent algorithm is roughly 1.5 times faster than the WF algorithm in the real case.

To this end, we let $E(\mathbf{x}, \mathbf{y})$ be the function defined in (3.2). Choose $\lambda = 0$ and assume $\mathbf{x}_k \approx \mathbf{y}_k$. Then, by Taylor's expansion, we obtain

$$\begin{aligned}
&E(\mathbf{x}_{k+1}, \mathbf{y}_k) \\
&\approx E(\mathbf{x}_k, \mathbf{y}_k) + \nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k)^* \begin{bmatrix} \mathbf{x}_{k+1} - \mathbf{x}_k \\ \mathbf{x}_{k+1} - \mathbf{x}_k \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \mathbf{x}_{k+1} - \mathbf{x}_k \\ \mathbf{x}_{k+1} - \mathbf{x}_k \end{bmatrix}^* \nabla_{\mathbf{x}}^2 E(\mathbf{x}_k, \mathbf{y}_k) \begin{bmatrix} \mathbf{x}_{k+1} - \mathbf{x}_k \\ \mathbf{x}_{k+1} - \mathbf{x}_k \end{bmatrix} \\
&= E(\mathbf{x}_k, \mathbf{x}_k) - \alpha_k \left\| \nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k) \right\|_2^2 + \frac{\alpha_k^2}{2} \Re \left(\nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k)^* \left(\sum_n \mathbf{f}_n \mathbf{f}_n^* \mathbf{y}_k \mathbf{y}_k^* \mathbf{f}_n \mathbf{f}_n^* \right) \nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k) \right) \\
&= E(\mathbf{x}_k, \mathbf{y}_k) - \alpha_k \left\| (\nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k) \right\|_2^2 + \frac{\alpha_k^2}{2} \nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k)^* \left(\sum_n \mathbf{f}_n \mathbf{f}_n^* \mathbf{y}_k \mathbf{y}_k^* \mathbf{f}_n \mathbf{f}_n^* \right) \nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k) \\
&= E(\mathbf{x}_k, \mathbf{y}_k) - \alpha_k \left(\left\| \nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k) \right\|_2^2 - \frac{\alpha_k}{2} \nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k)^* \left(\sum_n \mathbf{f}_n \mathbf{f}_n^* \mathbf{y}_k \mathbf{y}_k^* \mathbf{f}_n \mathbf{f}_n^* \right) \nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k) \right)
\end{aligned} \tag{A.1}$$

The last equality hold because $\sum_n \mathbf{f}_n \mathbf{f}_n^* \mathbf{y}_k \mathbf{y}_k^* \mathbf{f}_n \mathbf{f}_n^*$ is Hermitian. We choose $\alpha_k > 0$. Therefore, $E(\mathbf{x}_{k+1}, \mathbf{y}_k) - E(\mathbf{x}_k, \mathbf{y}_k) \le 0$ as long as

$$\frac{2}{\alpha_k} \ge \frac{\nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k)^* \left(\sum_n \mathbf{f}_n \mathbf{f}_n^* \mathbf{y}_k \mathbf{y}_k^* \mathbf{f}_n \mathbf{f}_n^* \right) \nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k)}{\left\| \nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k) \right\|_2^2},$$

which is guaranteed if

$$\alpha_k \le \frac{2}{\left\| \sum_n \mathbf{f}_n \mathbf{f}_n^* \mathbf{y}_k \mathbf{y}_k^* \mathbf{f}_n \mathbf{f}_n^* \right\|_2}.$$

To minimize $E(\mathbf{x}_{k+1}, \mathbf{y}_k) - E(\mathbf{x}_k, \mathbf{y}_k)$, it is easy seen from (A.1) that α_k is chosen as

$$\alpha_k = \frac{\left\| \nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k) \right\|_2^2}{\nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k)^* \left(\sum_n \mathbf{f}_n \mathbf{f}_n^* \mathbf{y}_k \mathbf{y}_k^* \mathbf{f}_n \mathbf{f}_n^* \right) \nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k)}.$$

In this case,

$$E(\mathbf{x}_{k+1}, \mathbf{y}_k) - E(\mathbf{x}_k, \mathbf{y}_k) \approx -\frac{\left\| \nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k) \right\|_2^4}{2 \nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k)^* \left(\sum_n \mathbf{f}_n \mathbf{f}_n^* \mathbf{y}_k \mathbf{y}_k^* \mathbf{f}_n \mathbf{f}_n^* \right) \nabla_{\mathbf{x}} E(\mathbf{x}_k, \mathbf{y}_k)}. \tag{A.2}$$

Now we consider the WF algorithm, which minimizes $G(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N (|\mathbf{f}_n^* \mathbf{x}|^2 - b_n)^2$. Assume we have the same $\mathbf{x}_k$ as in the alternating gradient descent algorithm, and the WF algorithm generates the new iterates by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \delta_k \nabla_{\mathbf{x}} G(\mathbf{x}_k).$$

With the optimal choice of δ_k, an analogous analysis leads to

$$\begin{aligned}
&G(\mathbf{x}_{k+1}) - G(\mathbf{x}_k) \\
&\approx -\frac{1}{2} \frac{\left\| \nabla_{\mathbf{x}} G(\mathbf{x}_k) \right\|_2^4}{\Re \left(\nabla_{\mathbf{x}} G(\mathbf{x}_k)^* H_{11}(\mathbf{x}_k) \nabla_{\mathbf{x}} G(\mathbf{x}_k) \right) + \Re \left(\nabla_{\mathbf{x}} G(\mathbf{x}_k)^T H_{21}(\mathbf{x}_k) \nabla_{\mathbf{x}} G(\mathbf{x}_k) \right)},
\end{aligned} \tag{A.3}$$

where $\Re$ denotes the real part, and

$$H_{11}(\mathbf{x}_k) = 4\sum_n \mathbf{f}_n\mathbf{f}_n^*\mathbf{x}_k\mathbf{x}_k^*\mathbf{f}_n\mathbf{f}_n^*,$$
$$H_{21}(\mathbf{x}_k) = \sum_n \left(\overline{\mathbf{f}_n\mathbf{f}_n^*\mathbf{x}_k}\mathbf{x}_k^*\mathbf{f}_n\mathbf{f}_n^* + \overline{\mathbf{f}_n\mathbf{f}_n^*\mathbf{x}_k}\mathbf{x}_k^*\mathbf{f}_n\mathbf{f}_n^*\right).$$

Since we assumed $\mathbf{x}_k \approx \mathbf{y}_k$ and $\lambda = 0$,

$$\nabla_{\mathbf{x}} G(\mathbf{x}_k) \approx 2\nabla_{\mathbf{y}} E(\mathbf{x}_k, \mathbf{y}_k), \tag{A.4}$$

which implies

$$\begin{aligned}
&\Re\left(\nabla_{\mathbf{x}} G(\mathbf{x}_k)^* H_{11}(\mathbf{x}_k)\nabla_{\mathbf{x}} G(\mathbf{x}_k)\right)\\
&= \nabla_{\mathbf{x}} G(\mathbf{x}_k)^* H_{11}(\mathbf{x}_k)\nabla_{\mathbf{x}} G(\mathbf{x}_k)\\
&\approx \left(2\nabla_{\mathbf{y}} E(\mathbf{x}_k, \mathbf{y}_k)\right)^* \left(4\sum_n \mathbf{f}_n\mathbf{f}_n^*\mathbf{x}_k\mathbf{x}_k^*\mathbf{f}_n\mathbf{f}_n^*\right)\left(2\nabla_{\mathbf{y}} E(\mathbf{x}_k, \mathbf{y}_k)\right)\\
&\approx 16\cdot\left(\nabla_{\mathbf{y}} E(\mathbf{x}_k, \mathbf{y}_k)\right)^* \left(\sum_n \mathbf{f}_n\mathbf{f}_n^*\mathbf{y}_k\mathbf{y}_k^*\mathbf{f}_n\mathbf{f}_n^*\right)\nabla_{\mathbf{y}} E(\mathbf{x}_k, \mathbf{y}_k).
\end{aligned} \tag{A.5}$$

If we further assume all vectors involved are real, then we have $H_{21}(\mathbf{x}_k) = 2\sum_n \mathbf{f}_n\mathbf{f}_n^*\mathbf{x}_k\mathbf{x}_k^*\mathbf{f}_n\mathbf{f}_n^*$ and

$$\begin{aligned}
\Re\left(\nabla_{\mathbf{x}} G(\mathbf{x}_k)^T H_{21}(\mathbf{x}_k)\nabla_{\mathbf{x}} G(\mathbf{x}_k)\right) &= \nabla_{\mathbf{x}} G(\mathbf{x}_k)^* H_{21}(\mathbf{x}_k)\nabla_{\mathbf{x}} G(\mathbf{x}_k)\\
&\approx 8\cdot\nabla_{\mathbf{y}} E(\mathbf{x}_k, \mathbf{y}_k)^*\left(\sum_n \mathbf{f}_n\mathbf{f}_n^*\mathbf{y}_k\mathbf{y}_k^*\mathbf{f}_n\mathbf{f}_n^*\right)\nabla_{\mathbf{y}} E(\mathbf{x}_k, \mathbf{y}_k).
\end{aligned} \tag{A.6}$$

Substituting (A.4), (A.5), and (A.6) into (A.3), we get

$$(G(\mathbf{x}_{k+1}) - G(\mathbf{x}_k)) \approx \frac{2}{3}\left(E(\mathbf{x}_{k+1}, \mathbf{y}_k) - E(\mathbf{x}_k, \mathbf{y}_k)\right)$$

This means that *the alternating gradient descent algorithm is* 1.5 *times faster than Wirtinger flow in terms of the decreasing of the objective.*

References

1. Alexeev, B., Bandeira, A.S., Fickus, M., Mixon, D.G.: Phase retrieval with polarization. SIAM J. Imaging Sci. **7**(1), 35–66 (2014)
2. Bahmani, S., Romberg, J.: Phase retrieval meets statistical learning theory: a flexible convex relaxation (2016). arXiv preprint arXiv:1610.04210
3. Balan, R., Bodmann, B.G., Casazza, P.G., Edidin, D.: Painless reconstruction from magnitudes of frame coefficients. J. Fourier Anal. Appl. **15**(4), 488–501 (2009)
4. Balan, R., Casazza, P., Edidin, D.: On signal reconstruction without phase. Appl. Comput. Harmon. Anal. **20**(3), 345–356 (2006)
5. Bolte, J., Sabach, S., Teboulle, M.: Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Math. Program. **146**(1–2), 459–494 (2014)
6. Candès, E.J., Eldar, Y.C., Strohmer, T., Voroninski, V.: Phase retrieval via matrix completion. SIAM Rev. **57**(2), 225–251 (2015)
7. Candès, E.J., Li, X.: Solving quadratic equations via phaselift when there are about as many equations as unknowns. Found. Comput. Math. **14**(5), 1017–1026 (2014)
8. Candès, E.J., Li, X., Soltanolkotabi, M.: Phase retrieval via Wirtinger flow: theory and algorithms. IEEE Trans. Inf. Theory **61**(4), 1985–2007 (2015)

9. Candès, E.J., Strohmer, T., Voroninski, V.: Phaselift: exact and stable signal recovery from magnitude measurements via convex programming. Commun. Pure Appl. Math. **66**(8), 1241–1274 (2013)
10. Chai, A., Moscoso, M., Papanicolaou, G.: Array imaging using intensity-only measurements. Inverse Probl. **27**(1), 015,005 (2010)
11. Chen, Y., Candès, E.: Solving random quadratic systems of equations is nearly as easy as solving linear systems. In: Advances in Neural Information Processing Systems, pp. 739–747 (2015)
12. Duchi, J.C., Ruan, F.: Solving (most) of a set of quadratic equalities: composite optimization for robust phase retrieval (2017). arXiv preprint arXiv:1705.02356
13. Goldstein, T., Studer, C.: Phasemax: convex phase retrieval via basis pursuit. IEEE Trans. Inf. Theory **64**(4), 2675–2689 (2018)
14. Hardt, M.: Understanding alternating minimization for matrix completion. In: 2014 IEEE 55th Annual Symposium on Foundations of Computer Science (FOCS), pp. 651–660. IEEE (2014)
15. Harrison, R.W.: Phase problem in crystallography. JOSA A **10**(5), 1046–1055 (1993)
16. Heinosaari, T., Mazzarella, L., Wolf, M.M.: Quantum tomography under prior information. Commun. Math. Phys. **318**(2), 355–374 (2013)
17. Iwen, M., Viswanathan, A., Wang, Y.: Robust sparse phase retrieval made easy. Appl. Comput. Harmon. Anal. **42**(1), 135–142 (2017)
18. Jain, P., Netrapalli, P., Sanghavi, S.: Low-rank matrix completion using alternating minimization. In: Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of computing, pp. 665–674. ACM (2013)
19. Millane, R.P.: Phase retrieval in crystallography and optics. JOSA A **7**(3), 394–411 (1990)
20. Netrapalli, P., Jain, P., Sanghavi, S.: Phase retrieval using alternating minimization. In: Advances in Neural Information Processing Systems, pp. 2796–2804 (2013)
21. Sun, J., Qu, Q., Wright, J.: A geometric analysis of phase retrieval. Found. Comput. Math. **18**(5), 1131–1198 (2018)
22. Tanner, J., Wei, K.: Low rank matrix completion by alternating steepest descent methods. Appl. Comput. Harmon. Anal. **40**(2), 417–429 (2016)
23. Waldspurger, I., dAspremont, A., Mallat, S.: Phase recovery, maxcut and complex semidefinite programming. Math. Program. **149**(1–2), 47–81 (2015)
24. Walther, A.: The question of phase retrieval in optics. J. Mod. Opt. **10**(1), 41–49 (1963)
25. Wang, G., Giannakis, G., Saad, Y., Chen, J.: Solving most systems of random quadratic equations. In: Advances in Neural Information Processing Systems, pp. 1865–1875 (2017)
26. Wang, G., Giannakis, G.B., Eldar, Y.C.: Solving systems of random quadratic equations via truncated amplitude flow. IEEE Trans. Inf. Theory **64**, 773–794 (2018)
27. Wei, K.: Solving systems of phaseless equations via kaczmarz methods: a proof of concept study. Inverse Probl. **31**(12), 125,008 (2015)
28. Wei, K., Cai, J.F., Chan, T.F., Leung, S.: Guarantees of riemannian optimization for low rank matrix completion (2016). arXiv preprint arXiv:1603.06610
29. Zhang, H., Chi, Y., Liang, Y.: Median-truncated nonconvex approach for phase retrieval with outliers (2017). arXiv preprint arXiv:1603.63805