Phase Retrieval via Smoothing Projected Gradient Method

Samuel Pinilla*, Jorge Bacca†‡, Jhon Angarita * and Henry Arguello†§
* Department of Electronic Engineering, Universidad Industrial de Santander
† Department of Computer Science, Universidad Industrial de Santander
Email: *samuel.pinilla@correo.uis.edu.co, ‡jorge.bacca1@correo.uis.edu.co, §henarfu@uis.edu.co

***Abstract*—Phase retrieval is a kind of ill-posed inverse problem, which is present in various applications, such as optics, astronomical imaging, and X-ray crystallography. Mathematically this inverse problem consists on recovering an unknown signal $\mathbf{x} \in \mathbb{R}^n/\mathbb{C}^n$ from a set of absolute square projections $y_k = |\langle \mathbf{a}_k, \mathbf{x}\rangle|^2, k = 1, \cdots, m$, where $\mathbf{a}_k$ are the sampling vectors. However, the square absolute function is in general non-convex and non-differentiable, which are desired properties in order to solve the problem, when traditional convex optimization algorithms are used. Therefore, this paper introduces a special differentiable function, known as smoothing function, in order to solve the phase retrieval problem by using the smoothing projected gradient (SPG) method. Moreover, to accelerate the convergence of this algorithm, this paper uses a nonlinear conjugate gradient method applied to the smoothing function as the search direction. Simulation results are provided to validate its efficiency on existing algorithms for phase retrieval. It is shown that compared with recently developed algorithms, the proposed method is able to accelerate the convergence.**

I. Introduction

Phase retrieval (PR) consists on recovering a signal from phaseless measurements, which is useful in many fields in science and engineering, such as optics [1], astronomical imaging [2], microscopy [3] and x-ray crystallography [4]. For example, in x-ray crystallography [5], the PR technique is used to determine the atomic position of a crystal in a three-dimensional (3D) space [6]. Moreover, a recent approach which modifies the traditional x-ray crystallography system has drawn attention, since it combines some actives fields such as x-ray imaging, coded diffractive imaging and phase retrieval techniques [7], [8], [9], [10].

The most traditional algorithms to solve the phase retrieval problem are based on the Error-Reduction method [11] which was proposed in 1970, however, the rate of convergence of these algorithms is considerably slow and does not have theoretical guarantees [8], [11]. Recently, a convex formulation was proposed in [12] via Phaselift, which consists on lifting up the original vector recovering problem from a quadratic system into recovery a rank-1 matrix. Further, theoretical guarantees of convergence and recovery have been developed for this convex approach, but its computational complexity becomes too high when the dimension of the signal is large. On the other hand, more recent methods described in [7], [13], [14] recover the phase by applying techniques such as non-convex formulations, and matrix completion. Specifically, one of the non-convex formulations, called Truncated Wirtinger Flow (TWF) algorithm proposed in [7], which optimizes the Poisson likelihood and keeps the convergence by designing truncation thresholds for calculating the step gradient. Additionally, in [15] and [16] the Truncated Amplitude Flow (TAF) and the Reshaped Wirtinger Flow (RWF) algorithms were developed, respectively, which are also gradient descend non-convex methods based on the Wirtinger derivate. Further, in terms of the sample complexity and speed of convergence, the TAF and RWF methods exhibit a superior performance over the actual state-of-the-art algorithms. It is important to highlight that the functions which optimize the TAF and RWF methods are also non-convex and non-differentiable.

In summary, the TWF, TAF and RWF algorithms optimize cost functions which are non-convex and non-differentiable. Moreover, the TWF and TAF algorithms require the extra truncation procedure in the step gradient, which in practice means to calculate more parameters to obtain a desired performance in recovering the phase. On the other hand, in [17] the Smoothing Projected Gradient (SPG) method was developed, for non-differentiable and non-convex optimization problems on a closed convex set. Moreover, the SPG algorithm solves the non-differentiability of the optimization problem by introducing a special differentiable function, called smoothing function, which approximates the original optimization function.

Given that the phase retrieval can be formulated as a non-convex and non-differentiable optimization problem, this paper proposes an algorithm based on the (SPG) method, by introducing a smoothing function, in order to solve the phase retrieval problem. Moreover, to accelerate the convergence of this algorithm, this paper uses a nonlinear conjugate gradient method applied to the smoothing function as the search direction. Numerical results validate its efficiency, showing that when compared with recently developed algorithms, such as TWF, TAF and RWF, the proposed method is able to accelerate the convergence. Specifically, the proposed method requires up to 66.7%, 58.3% and 41.1% less number of iterations with respect to the TWF, TAF and RWF algorithms, respectively. Further, the sample complexity of the proposed method is lower in comparison with the named state-of-the-art methods, in terms of the number of measurements. Also, it is important to highlight that the proposed method does not require the truncation procedure used in the TWF, and TAF.

II. Phase Retrieval Problem

Phase retrieval can be formulated by solving the system of m quadratic equations of the form

$$y_k = |\langle \mathbf{a}_k, \mathbf{x}\rangle|^2, k = 1, \cdots, m, \tag{1}$$

where the data vector $\mathbf{y} := [y_1, \cdots, y_m]^T \in \mathbb{R}^m$ represents the measurements, $\mathbf{a}_k \in \mathbb{R}^n/\mathbb{C}^n$ are the known sampling vectors and $\mathbf{x} \in \mathbb{R}^n/\mathbb{C}^n$ is the desired unknown signal. This work considers the sampling vectors are assumed to be independently and identically distributed (i.i.d.), that is, $\mathbf{a}_k \sim \mathcal{N}(0, \mathbf{I}_n)$ and $\mathbf{a}_k \sim \mathcal{N}(0, \frac{1}{2}\mathbf{I}_n) + j\mathcal{N}(0, \frac{1}{2}\mathbf{I}_n)$ for the real and complex cases respectively, where $j = \sqrt{-1}$. Then, adopting the least-squares criterion, the task of recovering a solution from the phaseless measurements in (1) reduces to minimize the amplitude-based loss function

$$\min_{\mathbf{x}\in\mathbb{R}^n/\mathbb{C}^n} f(\mathbf{x}) = \frac{1}{m}\sum_{k=1}^{m}\left(f_k(\mathbf{x}) - q_k\right)^2, \tag{2}$$

where $f_k(\mathbf{x}) = |\langle \mathbf{a}_k, \mathbf{x}\rangle|$ and $q_k = \sqrt{y_k}$. However, notice that the optimization problem in (2) is non-differentiable and non-convex [12]. This work proposes an algorithm based on the Smoothing Projected Gradient (SPG) method, which was proposed in [17] to solve non-differentiable and non-convex optimization problems. The SPG algorithm assumes that the objective function is locally Lipschitz continuous but not necessary differentiable. Also, the SPG method introduces an auxiliary differentiable function g to approximate the original objective function, in order to solve the non-differentiable and non-convex optimization problem. Moreover, the SPG requires some conditions over the auxiliary function g, but this will be discussed in Section III. According to the SPG assumptions, it is necessary to prove that the objective function $f(\mathbf{x})$ in (2) is Locally Lipschitz continuous.

Throughout the paper the following notations are considered. The set $\mathbb{R}_+ = \{x \in \mathbb{R} : x \geq 0\}$ and the set $\mathbb{R}_{++} = \{x \in \mathbb{R} : x > 0\}$. The distance of any two complex vectors $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{C}^n$ is defined as

$$d_r(\mathbf{x}_1, \mathbf{x}_2) = \min_{\theta\in[0,2\pi)} \|\mathbf{x}_1 e^{-j\theta} - \mathbf{x}_2\|_2, \tag{3}$$

where $j = \sqrt{-1}$, and $\|\cdot\|_2$ denotes the Euclidean norm. Notice that the distance $d_r(\cdot,\cdot)$ defined in (3) reduces to calculate $d_r(\mathbf{x}_1, \mathbf{x}_2) := \min\|\mathbf{x}_1 \pm \mathbf{x}_2\|_2$ for $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$. For the analysis we focus on the real-valued case, and the effectiveness of the method via numerical results is showed in Section IV.

Definition II.1. *Locally Lipschitz continuous under the distance $d_r(\cdot,\cdot)$*: Let $f : (\mathbb{R}^n, d_r(\cdot,\cdot)) \to \mathbb{R}$ be a function. The function f is called Lipschitz continuous if there exists a constant $L > 0$ such that, for $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$

$$|f(\mathbf{x}_1) - f(\mathbf{x}_2)| \leq L\ d_r(\mathbf{x}_1, \mathbf{x}_2). \tag{4}$$

The following lemma shows that $f(\mathbf{x})$ in (2) is locally Lipschitz according to Definition II.1.

Lemma II.1. The function $f(\mathbf{x})$ in (2) is locally Lipschitz continuous.

Proof: The proof of Lemma II.1 is deferred to Appendix A. ■

The next section introduces the concept of a smoothing function. Also, the smoothing function g which approximates the function $f(\mathbf{x})$ in (2) will be presented. Moreover, the conditions over the function g in order to guarantee the convergence of the proposed method are established.

III. Phase Retrieval algorithm

The concept of the smoothing function was presented in [17] as the following definition, which is an important notion in the SPG algorithm.

Definition III.1. *Smoothing function:* Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function. Then $g : \mathbb{R}^n \times \mathbb{R}_{++} \to \mathbb{R}$ is called a smoothing function of f, if $g(\cdot, \mu)$ is continuous differentiable in $\mathbb{R}^n$ for any fixed $\mu \in \mathbb{R}_{++}$ and

$$\lim_{\mu\downarrow 0} g(\mathbf{x}, \mu) = f(\mathbf{x}), \tag{5}$$

for any fixed $\mathbf{x} \in \mathbb{R}^n$.

According to the above definition, consider the function $\varphi_\mu : \mathbb{R} \to \mathbb{R}_{++}$ defined as

$$\varphi_\mu(x) = \sqrt{x^2 + \mu^2}, \tag{6}$$

where $\mu \in \mathbb{R}_{++}$. The following lemma shows that φ_μ has important smooth properties to approximate the functions f_k, given that $\varphi_0(\mathbf{a}_k^T\mathbf{x}) = f_k(\mathbf{x})$.

Lemma III.1. The function $\varphi_\mu(x)$, defined in (6), has the following properties.

1) $\varphi_\mu(x)$ and $\varphi_\mu(x)\varphi'_\mu(x)$ are Lipschitz continuous functions.
2) $\varphi_\mu(x)$ converges uniformly to $\varphi_0(x)$ on $\mathbb{R}$, that is, $|\varphi_\mu(x) - \varphi_0(x)| \leq \mu$.

Proof: The proof of Lemma III.1 is deferred to the Appendix B. ■

The first result in Lemma III.1 is used to guarantee the convergence of the proposed algorithm in Subsection III-B. Also, note that item 2) in Lemma III.1 establishes that the function $\varphi_\mu(\mathbf{a}_k^T\mathbf{x})$ approximates uniformly $f_k(\mathbf{x})$, which is a desirable approximation, since it only depends on the value of μ. Thus, a differentiable optimization problem to recover the unknown desired signal $\mathbf{x} \in \mathbb{R}^n$ from the measurements q_k in (2) can be formulated as

$$\min_{\mathbf{x}\in\mathbb{R}^n/\mathbb{C}^n} g(\mathbf{x}, \mu) = \frac{1}{m}\sum_{k=1}^{m}\left(\varphi_\mu(\mathbf{a}_k^T\mathbf{x}) - q_k\right)^2, \tag{7}$$

where $g(\mathbf{x}, \mu)$ is the smoothing function of $f(\mathbf{x})$ in (2). Then, in order to solve (7), this work presents the Phase Retrieval Smoothing Conjugate Gradient method (PR-SCG) which is summarized in Algorithm 1.

Algorithm 1 is a descend gradient method based on the SPG algorithm. Further, this paper uses a nonlinear conjugate gradient method developed in [18], in order to accelerate its convergence. Algorithm 1 calculates the conjugate direction in Line 16. Moreover, the smoothing parameter is updated as in the smoothing projected gradient method (SPG) in [17], to obtain a new point. That is, if $\|\nabla g(\mathbf{x}_{i+1}, \mu_i)\| \geq \gamma\mu_i$ in Line 10 is satisfied, then the smoothing parameter is updated using the new point in Line 13. Also, a backtracking line search strategy is used to choose a correct step size of the conjugate gradient update direction, which is calculated in Line 9. On the other hand, each vector $\tilde{\mathbf{g}}_i$ in Algorithm 1 is calculated as the partial derivative of the function $g(\mathbf{x}, \mu)$ respect to $\mathbf{x}$, *i.e*, $\tilde{\mathbf{g}}_i = \nabla_{\mathbf{x}} g(\mathbf{x}_i, \mu_i)$, which is given by

$$\nabla g(\mathbf{z}^t, \mu_t) = \frac{2}{m}\sum_{i=1}^{m}(\varphi_{\mu_t}(\mathbf{a}_i^T\mathbf{z}^t) - q_i)\nabla\varphi_{\mu_t}(\mathbf{a}_i^T\mathbf{z}^t), \tag{8}$$

where $\nabla\varphi_{\mu_t}(\mathbf{a}_i^T\mathbf{z}^t) = \frac{\mathbf{a}_i^T\mathbf{x}}{\varphi_{\mu_t}(\mathbf{a}_i^T\mathbf{z}^t)}\mathbf{a}_i$.

Algorithm 1 PR-SCG: Phase Retrieval Smoothing Conjugate Gradient Method

1: **Input:** Data $\{(\mathbf{a}_k; q_k)\}_{k=1}^m$ and choose constants $\varepsilon_0 > 0$, $r \geq 0$. Choose $\delta, \gamma_1 \in (0,1), \mu_0 > 0, \gamma > 0$ and T maximum number of iterations.
2: Initial point $\mathbf{x}_0 = \sqrt{\frac{\sum_{k=1}^m q_k^2}{m}}\tilde{\mathbf{z}}_0$. where $\tilde{\mathbf{z}}_0$ is the leading eigenvector of $\mathbf{Y}_0 := \frac{1}{|I_0|}\sum_{k \in I_0} \frac{\mathbf{a}_k \mathbf{a}_k^T}{\|\mathbf{a}_k\|^2}$.
3: Set $\mathbf{d}_0 = -\tilde{\mathbf{g}_0}$.
4: **for** $i = 0 : T-1$ **do**
5: Set $\rho = 1$.
6: **while** $g(\mathbf{x}_i + \rho \mathbf{d}_i, \mu_i) > (g(\mathbf{x}_i, \mu_i) + \delta \rho \tilde{\mathbf{g}}_i^T \mathbf{d}_i)$ **do**
7: $\rho = 0.4\rho$
8: **end while**
9: Set $\alpha_i = \rho$ and $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{d}_i$
10: **if** $\| \nabla g(\mathbf{x}_{i+1}, \mu_i) \| \geq \gamma \mu_i$ **then**
11: $\mu_{i+1} = \mu_i$
12: **else**
13: $\mu_{i+1} = \gamma \mu_i$
14: **end if**
15: $\mathbf{d}_{i+1} = -\tilde{\mathbf{g}}_{i+1} + \left(\frac{\tilde{\mathbf{g}}_{i+1}^T \tilde{\mathbf{z}}_i}{\mathbf{d}_i^T \tilde{\mathbf{z}}_i} - \frac{2\|\tilde{\mathbf{z}}_i\|^2 \tilde{\mathbf{g}}_{i+1}^T \mathbf{d}_i}{(\mathbf{d}_i^T \tilde{\mathbf{z}}_i)^2} \right) \mathbf{d}_i + \frac{\tilde{\mathbf{g}}_{i+1}^T \mathbf{d}_i}{\mathbf{d}_i^T \tilde{\mathbf{z}}_i} \tilde{\mathbf{z}}_i$.
16: where $\tilde{\mathbf{z}}_i = \tilde{\mathbf{p}}_i + \left(\epsilon_0 \| \tilde{\mathbf{g}}_{i+1} \|^r + \max\{0, -\frac{\mathbf{s}_i^T \tilde{\mathbf{p}}_i}{\|\mathbf{s}_i\|_2^2}\} \right) \mathbf{s}_i$.
17: $\tilde{\mathbf{p}}_i = \tilde{\mathbf{g}}_{i+1} - \tilde{\mathbf{g}}_i$, $\mathbf{s}_i = \mathbf{x}_{i+1} - \mathbf{x}_i$.
18: **end for**
19: **return:** $\mathbf{x}_T$

A. Initialization stage

This work uses the Orthogonality-promoting Initialization proposed in [15]. This initialization consists on calculating the vector $\mathbf{x}_0$, which is the leading eigenvector $\tilde{\mathbf{z}}_0$ of the matrix $\mathbf{Y}_0 := \frac{1}{|I_0|}\sum_{k \in I_0} \frac{\mathbf{a}_k \mathbf{a}_k^T}{\|\mathbf{a}_k\|^2}$ scaled by the quantity $\lambda_0 := \sqrt{\frac{\sum_{k=1}^m q_k^2}{m}}$, *i.e*, $\mathbf{x}_0 = \lambda_0 \tilde{\mathbf{z}}_0$. The set I_0 is the collection of indexes corresponding to the largest values of $\{q_k / \|\mathbf{a}_k\|\}$. Moreover, the notation $|I_0|$ is the cardinality of the set I_0. This procedure is calculated in Line 2 of Algorithm 1. Moreover, in [15] it was established that the distance between the initial guess $\mathbf{x}_0$ and the true signal $\mathbf{x}$ is given by

$$d_r(\mathbf{x}_0, \mathbf{x}) \leq \frac{1}{10}\|\mathbf{x}\|_2, \tag{9}$$

with probability exceeding $1-(m+5)e^{-n/2} - e^{-c_0 m} - 3/n^2$, providing that $m \geq c_1|I_0| \geq c_2 n$ for some constants $c_2, c_1, c_0 > 0$ and sufficiently large n.

B. Convergence conditions

The following assumption is used in the analysis of convergence for nonlinear conjugate gradient methods, in order to guarantee the convergence of Algorithm 1.

Assumption 1.

1) For any $(\tilde{\mathbf{x}}, \mu) \in \mathbb{R}^n \times \mathbb{R}_{++}$, the level set

$$S_\mu(\tilde{\mathbf{x}}) = \{\mathbf{x} \in \mathbb{R}^n | g(\mathbf{x}, \mu) \leq g(\tilde{\mathbf{x}}, \mu)\}, \tag{10}$$

is bounded.

2) The partial derivative $\nabla_{\mathbf{x}} g(\mathbf{x}, \mu)$ in (8) is continuously differentiable and there exists a constant $L_g > 0$ such that for any $\tilde{\mathbf{x}} \in \mathbb{R}^n$ and fixed $\mu \in \mathbb{R}_{++}$ is satisfied $\forall \mathbf{x}, \mathbf{y} \in S_\mu(\tilde{\mathbf{x}})$, that

$$d_r(\nabla_{\mathbf{x}} g(\mathbf{x}, \mu), \nabla_{\mathbf{x}} g(\mathbf{y}, \mu)) \leq L_g d_r(\mathbf{x}, \mathbf{y}). \tag{11}$$

Theorem III.3 shows that the objective function g defined in (7) satisfies the Assumption 1 by using the first result in Lemma III.1. However, before starting the proof of Theorem III.3, we need to introduce Lemma III.2, which is useful to prove item 2) in Theorem III.3.

Lemma III.2. Assume that f_1 and f_2 are Lipschitz continuous functions on a bounded set I with constants L_1 and L_2, respectively, and moreover assume there is a constant $v > 0$ such that $f_2(x) \geq v$ for all $x \in I$. Then f_1/f_2 is Lipschitz continuous on I. (The proof of Lemma III.2 can be found in [19]).

Theorem III.3. Functions $\varphi_\mu(x)$ and $g(\mathbf{x}, \mu)$, defined in Eqs. (6) and (7) respectively, satisfy the following properties:

1) Assumption 1 is satisfied.
2) The function $\varphi'_\mu(x)$ is Lipschitz continuous on any level set $S_\mu(\tilde{\mathbf{x}})$, for a fixed $\mu \in \mathbb{R}_{++}$.

Proof: The proof of Theorem III.3 can be found in Appendix C. ■

IV. NUMERICAL RESULTS

In this section, the performance of the proposed method relative to Truncated Wirtinger Flow (TWF) [7] , Reshaped Wirtinger Flow (RWF) [16] and Truncated Amplitude Flow (TAF) [15] is presented. All the parameters pertinent to the implementation of each algorithm were their suggested values. The performance metric was Relative error $:= d_r(\mathbf{z}, \mathbf{x})/\|\mathbf{x}\|$. For each trial, $1,000$ iterations for all algorithms were performed. A trial is declared to be successful when the returned estimate incurs a relative error less than 10^{-5}. All experiments were implemented in Matlab 2017a on an Intel Core i7 3.41Ghz CPU and 32 GB RAM. For reproducibility, the Matlab code of our PR-SCG algorithm is publicly available at http://diffraction.uis.edu.co/codes.html.

For all experiments, the signal is generated as $\mathbf{x} \sim \mathcal{N}(0, \mathbf{I}_{1,000})$ and the measurement $\mathbf{a}_k \sim \mathcal{N}(0, \mathbf{I}_{1,000})$ for $k = 1, ..., m$. The default values for the parameters in Algorithm 1 were established as $\varepsilon_0 = 10^{-10}, r = 2, \delta = 0.1, \gamma_1 = 0.5, \mu_0 = 20, \gamma = 0.08$ and $T = 1000$.

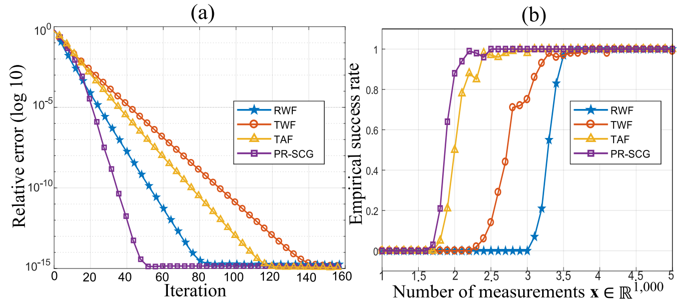

Fig. 1. Numerical results assuming $\mathbf{x} \sim \mathcal{N}(0, \mathbf{I}_n)$ and $\mathbf{a}_k \sim \mathcal{N}(0, \mathbf{I}_n)$ for a real Gaussian noiseless model. (a) Relative error versus iteration for $n = 1,000$, $m/n = 8$. (b) Empirical success rate versus number of measurements with $n = 1,000$, m/n varying 0.1 from 0 to 7 under the same Truncated spectral initialization.

The first experiment, shown in Fig. 1(a), compares the convergence speed of different schemes equipped with their own initialization in the original codes for real cases. Note that the proposed method overcomes its competing alternatives converging faster in terms of iterations. Also comparing the convergence rate in Fig. 1(b), the PR-SCG algorithm achieves a success rate of over 90% when $m/n = 2$ and guarantees perfect recovery from about $2.5n$ measurements for real-valued, which shows the effectiveness of the proposed method. It can be observed that the proposed algorithm needs less number of measurements to solve the phase retrieval problem in comparison with the TWF, TAF and RWF methods.

V. Conclusion

This paper presented the PR-CSG algorithm to solve the phase retrieval problem. Some numerical experiments were done to evaluate the performance of the proposed method in comparison with state-of-art methods. Specifically, simulations show an improvement of the PR-SCG method in the sample complexity, since it requires less number of measurements in contrast to the TWF, TAF and RWF algorithms. Moreover, the results also established that the convergence speed overcomes the recent methods in the literature, since it requires up to 66.7%, 58.3% and 41.1% less number of iterations with respect to the TWF, TAF and RWF algorithms.

Appendix A
Proof of Lemma II.1

Proof: To prove the lemma, we proceed showing that for all $k \in \{1, \cdots, m\}$ the functions $f_k(\mathbf{x})$ in (2) are locally Lipschitz continuous. Then, let $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$ be two any vectors, such that $|f_k(\mathbf{x}_1) - f_k(\mathbf{x}_2)| = ||\langle \mathbf{a}_k, \mathbf{x}_1\rangle| - |\langle \mathbf{a}_k, \mathbf{x}_2\rangle||$. Now, by using the triangle inequality on the right term, one can write

$$||\langle \mathbf{a}_k, \mathbf{x}_1\rangle| - |\langle \mathbf{a}_k, \mathbf{x}_2\rangle|| \leq |\langle \mathbf{a}_k, \mathbf{x}_1\rangle \pm \langle \mathbf{a}_k, \mathbf{x}_2\rangle|. \tag{12}$$

Using the fact that $\langle \mathbf{a}_k, \mathbf{x}_2\rangle = \mathbf{a}_k^T\mathbf{x}$, from (12) it can be obtained that $|f_k(\mathbf{x}_1) - f_k(\mathbf{x}_2)| \leq |\mathbf{a}_k^T(\mathbf{x}_1 \pm \mathbf{x}_2)|$. Since $\mathbf{a}_k^T\mathbf{x} = \sum_{i=1}^n (\mathbf{a}_k)_i (\mathbf{x})_i$, then by using the triangle inequality, one can write

$$\begin{aligned} |f_k(\mathbf{x}_1) - f_k(\mathbf{x}_2)| &\leq \textstyle\sum_{i=1}^n |(\mathbf{a}_k)_i| |(\mathbf{x}_1 \pm \mathbf{x}_2)_i| \\ &\leq a_{max}^k \|\mathbf{x}_1 \pm \mathbf{x}_2\|_1, \end{aligned} \tag{13}$$

where $a_{max}^k = \max\{|(\mathbf{a}_k)_i| : i = 1, \cdots, n\}$ of the fixed known vector $\mathbf{a}_k$ and $\|\cdot\|_1$ is the ℓ_1 norm. Finally, since ℓ_1 and ℓ_2 are equivalent norms, there exists a constant $\rho > 0$ such that $\|\mathbf{x}\|_1 \leq \rho\|\mathbf{x}\|_2$ for all $\mathbf{x} \in \mathbb{R}^n$ [20]. Thus, from (13) it can obtained that

$$|f_k(\mathbf{x}_1) - f_k(\mathbf{x}_2)| \leq \left(a_{max}^k \rho\right) d_r(\mathbf{x}_1, \mathbf{x}_2). \tag{14}$$

Thus, from (14) it can be concluded that f_k is a Lipschitz continuous function with constant $L_k = a_{max}^k \rho$. Remark that every Lipschitz continuous function is locally Lipschitz continuous [21]. Thus, since $f(\mathbf{x})$ is a sum of locally Lipschitz continuous functions, the result holds [19]. ∎

Appendix B
Proof of Lemma III.1

Proof: 1) Since $\mu > 0$ then $\varphi_\mu(x)$ is differentiable on $\mathbb{R}$, where $\varphi'_\mu(x)$ is given by $\varphi'_\mu(x) = \frac{x}{\sqrt{x^2+\mu^2}}$. Notice that $\sqrt{x^2+\mu^2} \geq x$ for all $x \in \mathbb{R}$, then $|\varphi'_\mu(x)| \leq 1$. Then, $\varphi_\mu(x)$ is a Lipschitz continuous function because its first derivative is bounded [19].

On the other hand, note that the function $\varphi_\mu(x)\varphi'_\mu(x)$ is given by $\varphi_\mu(x)\varphi'_\mu(x) = \sqrt{x^2+\mu^2}\left(\frac{x}{\sqrt{x^2+\mu^2}}\right) = x$. Now, taking into account Definition II.1, let $x_1, x_2 \in \mathbb{R}$ be two different real numbers such that

$$|\varphi_\mu(x_1)\varphi'_\mu(x_1) - \varphi_\mu(x_2)\varphi'_\mu(x_2)| = |x_1 - x_2| \leq |x_1 - x_2|. \tag{15}$$

Thus, from (15) it can be concluded that $\varphi_\mu(x)\varphi'_\mu(x)$ is a Lipschitz continuous function.

2) According to the definition of the function φ_μ in (6), it can be obtained that $|\varphi_\mu(x) - \varphi_0(x)| = |\sqrt{x^2+\mu^2} - \sqrt{x^2}|$. Note that by the Minkowski inequality [22], it can be concluded that $\sqrt{x^2+\mu^2} \leq \sqrt{x^2} + \mu$, therefore $|\varphi_\mu(x) - \varphi_0(x)| \leq |\sqrt{x^2} + \mu - \sqrt{x^2}| \leq \mu$. ∎

Appendix C
Proof of Theorem III.3

Proof: 1) Suppose that $S_\mu(\tilde{\mathbf{x}})$ is unbounded, then there exists a sequence $\{\mathbf{x}_\ell\} \subseteq S_\mu(\tilde{\mathbf{x}})$ such that $\|\mathbf{x}_\ell\|_2 \to \infty$. From the definition of the level set $S_\mu(\tilde{\mathbf{x}})$ in 10, it can be obtained that $g(\mathbf{x}_\ell, \mu) \leq g(\tilde{\mathbf{x}}, \mu) < \infty, \forall \ell \in \mathbb{N}$. However, $\|\mathbf{x}_\ell\|_2 \to \infty$ implies that $g(\mathbf{x}_\ell, \mu) \to \infty$ according to the definition of function g in (7). Then $g(\mathbf{x}_\ell, \mu) \to \infty$ is a contradiction, because $g(\mathbf{x}_\ell, \mu) < \infty$, $\forall \ell \in \mathbb{N}$. Thus, $S_\mu(\tilde{\mathbf{x}})$ is a bounded set.

To prove the second part of Assumption 1, we proceed to show that for each function $h_{i,\mu}(\mathbf{x}) = \left(\varphi_\mu\left(\mathbf{a}_i^T\mathbf{x}\right) - y_i\right)^2$ the condition in (11) is satisfied. Thus, since $g(\mathbf{x}, \mu)$ is the sum of the functions $h_{i,\mu}(\mathbf{x})$, then $g(\mathbf{x}, \mu)$ also satisfies (11) as it is proven in [21].

Consider, the gradient of the function $h_{i,\mu}$ which is given by $\nabla h_{i,\mu}(\mathbf{x}) = 2\left(\varphi_\mu\left(\mathbf{a}_i^T\mathbf{x}\right) - y_i\right)\varphi'_\mu\left(\mathbf{a}_i^T\mathbf{x}\right)\mathbf{a}_i$. Now, since $d_r(\cdot, \cdot)$ is a metric, it can be obtained that

$$d_r(\nabla h_i(\mathbf{x}_1), \nabla h_i(\mathbf{x}_2)) \leq 2\|\mathbf{a}_i\|_2(y_i p_1 + p_2), \tag{16}$$

where $p_1 = d_r(\varphi'_\mu(\mathbf{a}_i^T\mathbf{x}_1), \varphi'_\mu(\mathbf{a}_i^T\mathbf{x}_2))$ and $p_2 = d_r(\varphi_\mu(\mathbf{a}_i^T\mathbf{x}_1)\varphi'_\mu(\mathbf{a}_i^T\mathbf{x}_1), \varphi_\mu(\mathbf{a}_i^T\mathbf{x}_2)\varphi'_\mu(\mathbf{a}_i^T\mathbf{x}_2))$. Considering that the following item 2) shows that $\varphi'_\mu(x)$ is Lipschitz continuous on the bounded set $S_\mu(\tilde{\mathbf{x}})$ and from Lemma III.1 we have that $\varphi_\mu(x)\varphi'_\mu(x)$ is also Lipschitz continuous, then from Definition II.1 there exists constants $L_{\varphi'_\mu}$ and $L_{\varphi_\mu\varphi'_\mu}$ such that from (16) it can be obtained that

$$d_r(\nabla h_i(\mathbf{x}_1), \nabla h_i(\mathbf{x}_2)) \leq L_{h_{i,\mu}} d_r(\mathbf{x}_1, \mathbf{x}_2), \tag{17}$$

where $L_{h_{i,\mu}} = 2\|\mathbf{a}_i\|_2(y_i L_{\varphi'_\mu} + L_{\varphi_\mu\varphi'_\mu})$. Thus, the result holds.

2) Note that, from Lemma III.1 the function $\varphi'_\mu(x)$ can be expressed as $\varphi'_\mu(x) = \frac{f_1(x)}{f_2(x)}$, where $f_1(x) = x$ and $f_2(x) = \sqrt{x^2+\mu^2}$. Also, in Lemma III.1 was established that $f_1(x)$ and $f_2(x)$ are Lipschitz continuous functions, for all $x \in \mathbb{R}$. Notice that, $f_2(x) = \sqrt{x^2+\mu^2} \geq \mu > 0$, for any fixed μ. Now, considering the fact that $S_\mu(\tilde{\mathbf{x}})$ is a bounded set from item 1) and the previous conditions over functions $f_1(x)$ and $f_2(x)$, it can be obtained that $\varphi'_\mu(x)$ is a Lipschitz continuous function on $S_\mu(\tilde{\mathbf{x}})$, because the hypothesis in Lemma III.2 are satisfied. ∎

REFERENCES

[1] Y. Xu, Z. Ren, K. K. Wong, and K. Tsia, "Overcoming the limitation of phase retrieval using gerchberg–saxton-like algorithm in optical fiber time-stretch systems," *Optics letters*, vol. 40, no. 15, pp. 3595–3598, 2015.

[2] C. Fienup and J. Dainty, "Phase retrieval and image reconstruction for astronomy," *Image Recovery: Theory and Application*, pp. 231–275, 1987.

[3] S. Mayo, T. Davis, T. Gureyev, P. Miller, D. Paganin, A. Pogany, A. Stevenson, and S. Wilkins, "X-ray phase-contrast microscopy and microtomography," *Optics Express*, vol. 11, no. 19, pp. 2289–2302, 2003.

[4] S. Pinilla, J. Poveda, and H. Arguello, "Coded diffraction system in x-ray crystallography using a boolean phase coded aperture approximation," *Optics Communications*, vol. 410, pp. 707–716, 2018.

[5] R. P. Millane, "Phase retrieval in crystallography and optics," *JOSA A*, vol. 7, no. 3, pp. 394–411, 1990.

[6] M. Smyth and J. Martin, "x ray crystallography," *Journal of Clinical Pathology*, vol. 53, no. 1, p. 8, 2000.

[7] Y. Chen and E. Candes, "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.

[8] E. J. Candes, X. Li, and M. Soltanolkotabi, "Phase retrieval from coded diffraction patterns," *Applied and Computational Harmonic Analysis*, vol. 39, no. 2, pp. 277–299, 2015.

[9] S. E. Pinilla and H. Arguello, "Phase recovery from diffraction patterns using boolean coded apertures and the truncated wirtinger flow algorithm," in *3D Image Acquisition and Display: Technology, Perception and Applications*, pp. JT3A–28, Optical Society of America, 2016.

[10] S. Pinilla, C. Noriega, and H. Arguello, "Stochastic truncated wirtinger flow algorithm for phase retrieval using boolean coded apertures," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pp. 6050–6054, IEEE, 2017.

[11] J. R. Fienup, "Phase retrieval algorithms: a comparison," *Applied optics*, vol. 21, no. 15, pp. 2758–2769, 1982.

[12] E. J. Candès and X. Li, "Solving quadratic equations via phaselift when there are about as many equations as unknowns," *Foundations of Computational Mathematics*, vol. 14, no. 5, pp. 1017–1026, 2014.

[13] E. J. Candes, X. Li, and M. Soltanolkotabi, "Phase retrieval via wirtinger flow: Theory and algorithms," *IEEE Transactions on Information Theory*, vol. 61, no. 4, pp. 1985–2007, 2015.

[14] E. J. Candes, Y. C. Eldar, T. Strohmer, and V. Voroninski, "Phase retrieval via matrix completion," *SIAM review*, vol. 57, no. 2, pp. 225–251, 2015.

[15] G. Wang, G. B. Giannakis, and Y. C. Eldar, "Solving systems of random quadratic equations via truncated amplitude flow," *arXiv preprint arXiv:1605.08285*, 2016.

[16] H. Zhang and Y. Liang, "Reshaped wirtinger flow for solving quadratic system of equations," in *Advances in Neural Information Processing Systems*, pp. 2622–2630, 2016.

[17] C. Zhang and X. Chen, "Smoothing projected gradient method and its application to stochastic linear complementarity problems," *SIAM Journal on Optimization*, vol. 20, no. 2, pp. 627–649, 2009.

[18] X. Chen and W. Zhou, "Smoothing nonlinear conjugate gradient method for image restoration using nonsmooth nonconvex minimization," *SIAM Journal on Imaging Sciences*, vol. 3, no. 4, pp. 765–790, 2010.

[19] K. Eriksson, D. Estep, and C. Johnson, *Applied mathematics: Body and soul: Volume 1: Derivatives and geometry in IR3*. Springer Science & Business Media, 2013.

[20] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[21] A. Bagirov, N. Karmitsa, and M. M. Mäkelä, *Introduction to Nonsmooth Optimization: theory, practice and software*. Springer, 2014.

[22] E. Kreyszig, *Introductory functional analysis with applications*, vol. 1. wiley New York, 1989.