

Poisson Phase Retrieval in Very Low-Count Regimes

Zongyu Li , Graduate Student Member, IEEE, Kenneth Lange, and Jeffrey A. Fessler , Fellow, IEEE

Abstract—This paper proposes novel phase retrieval algorithms for maximum likelihood (ML) estimation from measurements following independent Poisson distributions in very low-count regimes, e.g., 0.25 photon per pixel. Specifically, we propose a modified Wirtinger flow (WF) algorithm using a step size based on the observed Fisher information. This approach eliminates all parameter tuning except the number of iterations. We also propose a novel curvature for majorize-minimize (MM) algorithms with a quadratic majorizer. We show theoretically that our proposed curvature is sharper than the curvature derived from the supremum of the second derivative of the Poisson ML cost function. We compare the proposed algorithms (WF, MM) with existing optimization methods, including WF using other step-size schemes, quasi-Newton methods and alternating direction method of multipliers (ADMM) algorithms, under a variety of experimental settings. Simulation experiments with a random Gaussian matrix, a canonical discrete Fourier transform (DFT) matrix, a masked DFT matrix and an empirical transmission matrix demonstrate the following. 1) As expected, algorithms based on the Poisson ML model consistently produce higher quality reconstructions than algorithms derived from Gaussian noise ML models when applied to low-count data. 2) For unregularized cases, our proposed WF algorithm with Fisher information for step size converges faster than other WF methods, e.g., WF with empirical step size, backtracking line search, and optimal step size for the Gaussian noise model; it also converges faster than the quasi-Newton method. 3) In regularized cases, our proposed WF algorithm converges faster than WF with backtracking line search, quasi-Newton, MM and ADMM.

Index Terms—Low-count image reconstruction, non-convex optimization, poisson phase retrieval.

I. INTRODUCTION

PHASE retrieval is an inverse problem with many applications in engineering and applied physics [1], [2], including radar [3], X-ray crystallography [4], astronomical imaging [5], Fourier ptychography [6], [7], [8], [9] and coherent diffractive

Manuscript received 14 April 2022; revised 11 August 2022; accepted 23 September 2022. Date of current version 5 October 2022. This work was supported in part by USPHS under Grants GM53275 and HG006139, in part by NSF under Grant IIS 1838179, and in part by NIH under Grant R01 EB022075. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Stanley H Chan. (Corresponding author: Zongyu Li.)

Zongyu Li and Jeffrey A. Fessler are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109-2122 USA (e-mail: zonyul@umich.edu; fessler@umich.edu).

Kenneth Lange is with the Departments of Computational Medicine, Human Genetics, and Statistics, University of California, Los Angeles, CA 90095 USA (e-mail: klange@ucla.edu).

Code for reproducing the results is available at <https://github.com/ZongyuLi-umich/PPR-low-count>.

This article has supplementary downloadable material available at <https://doi.org/10.1109/TCI.2022.3209936>, provided by the authors.

Digital Object Identifier 10.1109/TCI.2022.3209936

imaging (CDI) [10]. In these applications, the sensing systems can only measure the magnitude (or the square of the magnitude) of the signal, for example, optical detection devices (e.g., CCD cameras) cannot measure the phase of a light wave. The problem of recovering the original signal from only the magnitude of such linear measurements is called phase retrieval. Mathematically, the goal is to recover the unknown signal $\mathbf{x} \in \mathbb{F}^N$ from measurements $\{y_i\}$ that follow some statistical distribution

$$y_i \sim p(|\mathbf{a}'_i \mathbf{x}|^2 + b_i), \quad (1)$$

where $p(\cdot)$ is a probability density function. Here, $\mathbf{a}'_i \in \mathbb{C}^N$ denotes the i th row of the system matrix $\mathbf{A} \in \mathbb{C}^{M \times N}$, where $i = 1, \dots, M$, and $b_i \in \mathbb{R}_+$ denotes a known mean background signal for the i th measurement, e.g., as arising from dark current [11]. Here the field $\mathbb{F} = \mathbb{R}$ or $\mathbb{F} = \mathbb{C}$ depending on whether \mathbf{x} is known to be real or complex.

The sensing vectors $\{\mathbf{a}'_i\}$ are often assumed to follow some structure, e.g., i.i.d. random Gaussian, or the coefficients of discrete Fourier transform (DFT). For the random Gaussian case, Candès et al. [12] showed that $M \sim \mathcal{O}(N \log N)$ samples are sufficient to recover the signal; Bandeira et al. [13] posed a conjecture that $M = 4N - 4$ is necessary and sufficient to uniquely recover the original signal from noiseless measurements. However, under very low-count regimes with noise, a much larger M is often needed to successfully reconstruct the signal. Additionally, when \mathbf{A} corresponds to a Fourier transform, the measurements describe only the magnitudes of a signal's Fourier coefficients, and one usually does not have enough information to recover the signal; while the Fourier transform is injective, its point-wise absolute value is not [14]. So a common approach is to create redundancy in the measurement process by additional illuminations of the object using different masks [15]. Banderia et al. [14] showed that by using a set of $\mathcal{O}(\log M)$ random masks can increase the probability of recovering the signal.

A. Background for Gaussian Phase Retrieval

In many previous works, the measurement vector $\mathbf{y} \in \mathbb{R}^M$ was assumed to have statistically independent elements following Gaussian distributions with variance σ^2 :

$$y_i \sim \mathcal{N}(|\mathbf{a}'_i \mathbf{x}|^2 + b_i, \sigma^2). \quad (2)$$

For this Gaussian noise model, the ML estimate of \mathbf{x} corresponds to the following non-convex optimization problem

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathbb{F}^N} g(\mathbf{x}), \quad g(\mathbf{x}) \triangleq \sum_i \left| y_i - b_i - |\mathbf{a}'_i \mathbf{x}|^2 \right|^2. \quad (3)$$

To solve (3), numerous algorithms have been proposed. One approach reformulates (3) by “matrix lifting” [12], [15], [16], where a rank-one matrix is introduced and if the rank constraint is relaxed, then the transformed problem is convex and can be solved by semi-definite programming (SDP). The SDP based algorithms can yield robust solutions but can be computational expensive, especially on large-scale data. Another approach is Wirtinger Flow (WF) [17] and its variants [18], [19], [20] that descend the cost function with a (projected/thresholded/truncated) Wirtinger gradient using an appropriate step size. In the classic WF algorithm [17], the gradient ¹ for the Gaussian cost function (3) is

$$\nabla g(\mathbf{x}) = 4\mathbf{A}' \text{diag}\{|\mathbf{A}\mathbf{x}|^2 - \mathbf{y} + \mathbf{b}\}\mathbf{A}\mathbf{x}. \quad (4)$$

To descend the cost function, reference [17] used a heuristic where the step size μ is rather small for the first few iterations and gradually increases as the iterations proceed. The intuition is that the gradient is noisy at the early iterations so a small step size is preferred. A drawback of this approach is that one needs to select hyper-parameters that control the growth of μ . An alternative approach is to perform backtracking for μ at each iteration [21], i.e., by reducing μ until the cost function decreases sufficiently. This approach guarantees decreasing the cost function monotonically but can increase the compute time of the algorithm due to the variable number of inner iterations. Jiang et al. [18] derived the optimal step size for the Gaussian ML cost function (3) and showed faster convergence than the heuristic step size when measurements are noiseless or follow i.i.d. Gaussian distribution. Cai et al. [19] proposed thresholded WF and showed it can achieve the minimax optimal rates of convergence, but that scheme requires an appropriate selection of tuning parameters. Soltanolkotabi et al. [20] reformulated the phase retrieval problem as a nonconvex optimization problem and proved that projected Wirtinger gradient descent, when initialized in a neighborhood of the desired signal, has a linear convergence rate. However, it can be difficult to find an initial estimate satisfying the conditions mentioned in [20].

An alternative to cost function (3) (aka intensity model) is the magnitude model that works with the square root of \mathbf{y} . In particular, [22] proposed an algorithm known as Gerchberg Saxton (GS) that introduced a new variable $\boldsymbol{\theta}$ to represent the phase, leading to the following joint optimization problem

$$\begin{aligned} \hat{\mathbf{x}}, \hat{\boldsymbol{\theta}} &= \arg \min_{\mathbf{x} \in \mathbb{F}^N, \boldsymbol{\theta} \in \mathbb{C}^N} \|\mathbf{A}\mathbf{x} - \text{diag}\{\sqrt{\max(\mathbf{y} - \mathbf{b}, \mathbf{0})}\}\boldsymbol{\theta}\|_2^2, \\ &\text{subject to } |\theta_i| = 1, \quad i = 1, \dots, N. \end{aligned} \quad (5)$$

The square root in (5) is reminiscent of the Anscombe transform that converts a Poisson random variable into another random variable that approximately has a standard Gaussian distribution. However, that approximation is accurate when the Poisson mean is sufficiently large (e.g., above 5), whereas this paper focuses on the lower-count regime. The convergence and recovery guarantees of GS were studied in [23], [24].

¹If $\mathbf{x} \in \mathbb{R}^N$, then all gradients w.r.t. \mathbf{x} in this paper should be real and hence use only the real part of expressions like (4).

In addition to matrix-lifting, WF, GS and their variants, several other algorithms have been proposed to solve phase retrieval problems under the assumption of the Gaussian measurement noise, including Gauss-Newton methods [25], Limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGRS) updates to approximate the Hessian in the Newton’s method [26], majorize-minimize (MM) methods [21], alternating direction method of multipliers (ADMM) [27], and an iterative soft-thresholding with exact line search algorithm (STELA) [28]. It seems unlikely that any of the many existing methods for the Gaussian noise case are optimal for low-count Poisson noise.

B. Background for Poisson Phase Retrieval

In many low-photon count applications [8], [29], [30], [31], [32], [33], [34], especially in [34], where 0.25 photon per pixel on average is considered, a Poisson noise model is more appropriate:

$$y_i \sim \text{Poisson}(|\mathbf{a}'_i \mathbf{x}|^2 + b_i). \quad (6)$$

ML estimation of \mathbf{x} for the model (6) corresponds to the following optimization problem

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathbb{F}^N} f(\mathbf{x}), \quad f(\mathbf{x}) \triangleq \sum_i \psi(\mathbf{a}'_i \mathbf{x}; y_i, b_i),$$

$$\psi(v; y, b) \triangleq (|v|^2 + b) - y \log(|v|^2 + b). \quad (7)$$

Here, $f(\mathbf{x})$ denotes the negative log-likelihood corresponding to (6), ignoring irrelevant constants independent of \mathbf{x} , and the function $\psi(\cdot; y, b)$ denotes the marginal negative log-likelihood for a single measurement, where $v \in \mathbb{C}$. Because $|v|$ is real, it is helpful to re-write ψ in the form $\psi(v; y, b) = \phi(|v|; y, b)$, where

$$\phi(r; y, b) \triangleq (r^2 + b) - y \log(r^2 + b), \quad r \in \mathbb{R}_+. \quad (8)$$

One can verify that the function $\phi(r; y, b)$ is non-convex over $r \in \mathbb{R}_+$ when $0 < b < y$. That property, combined with the modulus within the logarithm in (7), makes (7) a challenging optimization problem. Similar problems for $b = 0$ have been considered previously [6], [7], [15], [35], [36], [37], [38], but many optical sensors also have Gaussian readout noise [7], [38] so that the mean background signal is unlikely to be zero. To accommodate the Gaussian readout noise, a more precise model would consider a sum of Gaussian and Poisson noise. However, the log likelihood for a Poisson plus Gaussian distribution is complicated, so a common approximation is to use a shifted Poisson model [39] that also leads to the cost function in (7). An alternative to the shifted Poisson model could be to work with an unbiased inverse transformation of a generalized Anscombe transform approximation [9], [40] or use a surrogate function that tightly upper bounds the challenging Poisson plus Gaussian ML objective function and apply a majorize-minimize algorithm [41]. Algorithms for the Poisson plus Gaussian noise model are interesting topics for future work.

Existing algorithms for the Poisson phase retrieval are limited in the literature. Chen et al. [36] proposed to solve the Poisson phase retrieval problem by minimizing a nonconvex functional as in the Wirtinger flow (WF) approach; Bian et al. [6] used Poisson ML estimation and truncated Wirtinger flow in Fourier

ptychographic (FP) reconstruction. Zhang et al. [42] consider a scale square root of (6) for the common case with $b_i = 0$. Chang et al. [43] derived a (TV) regularized ADMM algorithm for Poisson phase retrieval and established its convergence. Recently, Fatima et al. [44] proposed a double looped primal-dual majorize-minimize (PDMM) algorithm.

In this paper, we propose novel algorithms for the Poisson phase retrieval problem and report empirical comparisons of the convergence speed and reconstruction quality of algorithms under a variety of experimental settings. We presented a preliminary version of this work at the 2021 IEEE international conference on image processing (ICIP) [45]. We significantly extended this work by testing our proposed method under more practical experimental settings. We also added comparisons to related works such as [18], [26].

The main contributions of this paper can be summarized as follows:

- 1) We propose a novel method for computing the step size for the WF algorithm that can lead to faster convergence compared to empirical step size [17], backtracking line search [21], optimal step size derived for the Gaussian noise model [18], and LBFGS updates to approximate the Hessian in Newton's method [26]. Moreover, our proposed method can be computed efficiently without any tuning parameter.
- 2) We derive a majorize-minimize (MM) algorithm with quadratic majorizer using a novel curvature. We show theoretically that our proposed curvature is sharper than the curvature derived from the upper bound of the second derivative of the Poisson ML cost function.
- 3) We present numerical simulation results under random Gaussian, canonical DFT, masked DFT and empirical transmission system matrix settings for very low-count data, e.g., 0.25 photon per pixel. We show that under such experimental settings, algorithms derived from the Poisson ML model produce consistently higher reconstruction quality than algorithms derived from Gaussian ML model, as expected. Furthermore, the reconstruction quality is further improved by incorporating regularizers that exploit assumed properties of the signal.
- 4) We compare the convergence speed (in terms of cost function and peak signal-to-noise ratio (PSNR) vs. time) of WF with Fisher information with other methods for step size (backtracking line search, optimal Gaussian) and LBFGS quasi-Newton method. We also compare the convergence speed of regularized WF with MM and ADMM [46], using smooth regularizers such as corner-rounded anisotropic total variation (TV). For both cases, our proposed WF Fisher algorithm converges the fastest under all system matrix settings.

The rest of this paper is organized as follows. Section II introduces the proposed modified WF method with Fisher information for step size; and derives the improved curvature for the MM algorithm. Section III illustrates implementation details of algorithms discussed in Section II. Section IV provides numerical results using simulated data under different experimental

settings. Section V and Section VI discuss and conclude this paper and provide future directions.

Notation: Bold upper/lower case letters (e.g., \mathbf{A} , \mathbf{x} , \mathbf{y} , \mathbf{b}) denote matrices and column vectors, respectively. Italics (e.g., μ , y , b) denote scalars. y_i and b_i denote the i th element in vector \mathbf{y} and \mathbf{b} , respectively. \mathbb{R}^N and \mathbb{C}^N denote N -dimensional real/complex normed vector space, respectively. $(\cdot)^*$ denotes the complex conjugate and $(\cdot)'$ denotes Hermitian transpose. $\text{diag}\{\cdot\}$ is a diagonal matrix constructed from a column vector. Unless otherwise defined, a subscript denotes outer iterations and superscript denotes the inner iterations, respectively. For example, \mathbf{x}_k denotes the estimate of \mathbf{x} at the k th iteration of an algorithm. \oslash denotes element-wise division. The first and second derivatives of a scalar function ψ are denoted $\dot{\psi}$ and $\ddot{\psi}$, respectively. For gradients associated with complex numbers/vectors, the notation $\dot{\psi}(\cdot)$ and $\nabla(\cdot)$, should be considered as an ascent direction, not as a derivative.

II. METHODS

A. Wirtinger Flow (WF)

This section describes the modified WF algorithm with proposed step-size approach based on Fisher information. To generalize the Wirtinger flow algorithm to the Poisson cost function (7), the most direct approach simply replaces the gradient (4) by (9) in the WF framework [42] and performs backtracking to find the step-size μ , as in [21]. We propose a faster alternative next. We treat $0 \log 0$ as 0 in (7) because a Poisson random variable with zero mean can only take the value 0. With this assumption, one can verify that ψ has the following well-defined ascent direction (negative of descent direction [47]) and a second derivative:

$$\begin{aligned}\dot{\psi}(v; y, b) &= 2v \left(1 - \frac{y}{|v|^2 + b} \right), \quad v \in \mathbb{C}. \\ \ddot{\psi}(v; y, b) &= \text{sign}(v) \left(2 + 2y \frac{|v|^2 - b}{(|v|^2 + b)^2} \right), \\ |\dot{\psi}(v; y, b)| &\leq 2 + \frac{y}{4b}.\end{aligned}\tag{9}$$

1) *Fisher Information for Poisson Model:* We first make a quadratic approximation along the gradient direction of the cost function at each iteration, and then apply one step of Newton's method to minimize that 1D quadratic. Because computing the Hessian can be computationally expensive in large-scale problems, we follow the statistics literature by replacing the Hessian by the observed Fisher information when applying Newton's method [48], [49]. Our Fisher approach is based on the fact that the observed Fisher information is the negative Hessian matrix of the incomplete data log-likelihood functions evaluated at the observed data, and hence can provide a good approximation to the Hessian with enough data [50]. Moreover, the Fisher information matrix is always positive semi-definite, and avoids calculation of second derivatives. Using Fisher information in gradient-based algorithms has a long history in statistics and is central to Fisher's method of scoring [48], [49], [51], [52].

Specifically, we first approximate the 1D line search problem associated with (7) by the following Taylor series

$$\begin{aligned} \mu_k &= \arg \min_{\mu \in \mathbb{R}} f_k(\mu), \\ f_k(\mu) &\triangleq f(\mathbf{x}_k - \mu \nabla f(\mathbf{x}_k)) \approx f(\mathbf{x}_k) - \|\nabla f(\mathbf{x}_k)\|_2^2 \mu \\ &\quad + \frac{1}{2} \nabla f(\mathbf{x}_k)' \nabla^2 f(\mathbf{x}_k) \nabla f(\mathbf{x}_k) \mu^2, \end{aligned} \quad (10)$$

where one can verify that the minimizer is

$$\mu_k = \frac{\|\nabla f(\mathbf{x}_k)\|_2^2}{\text{real}\{\nabla f(\mathbf{x}_k)' \nabla^2 f(\mathbf{x}_k) \nabla f(\mathbf{x}_k)\}}. \quad (11)$$

We next approximate the Hessian matrix $\nabla^2 f(\mathbf{x})$ using the observed Fisher information matrix:

$$\begin{aligned} \nabla^2 f(\mathbf{x}) &\approx \mathbf{I}(\mathbf{x}, \mathbf{b}) \\ &\triangleq \mathbb{E}_{\mathbf{y}} \left[\nabla^2 f(\mathbf{x}; \mathbf{y}, \mathbf{b}) \middle| \mathbf{x}, \mathbf{b} \right] \\ &= \mathbb{E}_{\mathbf{y}} \left[(\nabla f(\mathbf{x}; \mathbf{y}, \mathbf{b})) (\nabla f(\mathbf{x}; \mathbf{y}, \mathbf{b}))' \middle| \mathbf{x}, \mathbf{b} \right] \\ &= \mathbf{A}' \mathbb{E}_{\mathbf{y}} \left[\left(\dot{\psi}(\mathbf{v}; \mathbf{y}, \mathbf{b}) \right) \left(\dot{\psi}(\mathbf{v}; \mathbf{y}, \mathbf{b}) \right)' \middle| \mathbf{v}, \mathbf{b} \right] \mathbf{A}. \end{aligned} \quad (12)$$

Here the dot subscript notation $\dot{\psi}(\mathbf{v}; \mathbf{y}, \mathbf{b})$ denotes element-wise application of the function $\dot{\psi}$ to its arguments (as in the Julia language), so the gradient $\dot{\psi}(\mathbf{v}; \mathbf{y}, \mathbf{b})$ is a vector in \mathbb{C}^M . One can verify that the marginal Fisher information for a single term $\psi(v; y, b)$ is

$$\begin{aligned} \bar{I}(v, b) &= \mathbb{E}_{\mathbf{y}} \left[|\dot{\psi}(v; y, b)|^2 \middle| v, b \right] \\ &= \frac{4|v|^2}{|v|^2 + b}, \quad v \in \mathbb{C}, b > 0. \end{aligned} \quad (13)$$

Substituting (13) into (12) using the statistical independence of the elements of the gradient vector, and then substituting (12) into (11) yields the simplified step-size expression

$$\mu_k \triangleq \frac{\|\nabla f(\mathbf{x}_k)\|_2^2}{\mathbf{d}'_k \mathbf{D}_1 \mathbf{d}_k} \in \mathbb{R}_+, \quad (14)$$

where $\mathbf{d}_k \triangleq \mathbf{A} \nabla f(\mathbf{x}_k)$ and $\mathbf{D}_1 \triangleq \text{diag}\{\bar{I}(\mathbf{A}\mathbf{x}_k, \mathbf{b})\}$. (Careful implementation avoids redundant matrix-vector products.)

This approach removes all tuning parameters other than number of iterations. In addition, using the observed Fisher information leads to a larger step size than using the best Lipschitz constant of (7), i.e., $\max_i \{2 + y_i / (4b_i)\}$ when $b_i > 0$, hence accelerating convergence.

To facilitate fair comparisons in subsequent sections, we also derive a Fisher information step size for the Gaussian noise model here. The marginal Fisher information for the scalar case of the Gaussian cost function (3) is

$$\begin{aligned} \bar{I}(v, b) &= \mathbb{E}_{\mathbf{y}} \left[|4|v|(|v|^2 - b - y)|^2 \middle| v, b \right] \\ &= 16|v|^2(|v|^2 + b). \end{aligned} \quad (15)$$

Substituting (15) into (14), one can also derive a convenient step size μ_k for the WF algorithm for the Gaussian model (3) using its observed Fisher information to approximate the exact Hessian. We used such step size in our experiment as will be discussed in Section IV.

2) *WF With Regularization*: To potentially improve the reconstruction quality, one often adds a regularizer or penalty to the Poisson log-likelihood cost function, leading to a cost function of the form

$$\Psi(\mathbf{x}) = f(\mathbf{x}) + \beta R(\mathbf{x}), \quad (16)$$

where $R: \mathbb{F}^N \mapsto \mathbb{R}_+$ is a regularizer and $\beta \geq 0$ denotes the regularization strength. The general methods in the paper are adaptable to many regularizers, but for simplicity we focus on regularizers that are based on the assumption that $\mathbf{T}\mathbf{x}$ is approximately sparse, for a $K \times N$ matrix \mathbf{T} . In particular, we used the corner-rounded anisotropic finite-difference matrix (aka total variation (TV)) for regularization. Because the WF algorithm requires a well-defined gradient, we replaced the ℓ_1 norm term with a Huber function regularizer of the form

$$\begin{aligned} R(\mathbf{x}) &= \mathbf{1}' h(\mathbf{T}\mathbf{x}; \alpha) = \min_{\mathbf{z}} \frac{1}{2} \|\mathbf{T}\mathbf{x} - \mathbf{z}\|_2^2 + \alpha \|\mathbf{z}\|_1, \\ h(t; \alpha) &\triangleq \begin{cases} \frac{1}{2}|t|^2, & |t| < \alpha, \\ \alpha|t| - \frac{1}{2}\alpha^2, & \text{otherwise,} \end{cases} \end{aligned} \quad (17)$$

which involves solving for \mathbf{z} analytically in terms of \mathbf{x} . This smooth regularizer is suitable for gradient-based methods like WF and for quasi-newton methods like LBFSGS, as well as for versions of MM and ADMM. We refer to (17) as ‘‘TV regularization’’ even though it is technically (anisotropic) ‘‘corner rounded’’ TV.

For the smooth regularizer (17), we majorize the Huber function $h(t)$ using quadratic polynomials with the optimal curvature using the ratio $\dot{h}(z)/z$ [52, p. 184], so that the step size μ_k becomes

$$\begin{aligned} \mu_k &\triangleq \frac{\|\nabla \tilde{f}(\mathbf{x}_k)\|_2^2}{\nabla \tilde{f}(\mathbf{x}_k)' (\mathbf{A}' \mathbf{D}_1 \mathbf{A} + \beta \mathbf{T}' \mathbf{D}_2 \mathbf{T}) \nabla \tilde{f}(\mathbf{x}_k)}, \\ \nabla \tilde{f}(\mathbf{x}_k) &\triangleq \nabla f(\mathbf{x}_k) + \beta \mathbf{T}' \dot{h}(\mathbf{T}\mathbf{x}_k; \alpha), \\ \mathbf{D}_2 &\triangleq \text{diag}\{\min(\alpha \oslash |\mathbf{T}\mathbf{x}_k|, 1)\}, \end{aligned} \quad (18)$$

where \oslash denotes element-wise division.

3) *Truncated Wirtinger Flow*: To potentially reduce the error in gradient estimation due to noisy measurements, Chen et al. [36] proposed a truncated Wirtinger flow (TWF) approach that uses only those measurements satisfying a threshold criterion to calculate the Wirtinger flow gradient. In particular, the threshold criterion [6] is defined as

$$|y_i - |\mathbf{a}'_i \mathbf{x}|| \leq a^h \frac{\|\mathbf{y} - |\mathbf{A}\mathbf{x}||_1}{M} \cdot \frac{|\mathbf{a}'_i \mathbf{x}|^2}{\|\mathbf{x}\|_2}, \quad (19)$$

where a^h is a user-defined parameter that controls the threshold value. When a^h is chosen appropriately, y_i values that do not satisfy (19) will be truncated when calculating the gradient, to try to reduce noise. However, we did not use gradient truncation in our experiments (Section IV) because we did not observe

Algorithm 1: Wirtinger Flow for the Poisson Model.**Input:** \mathbf{A} , \mathbf{y} , \mathbf{b} , \mathbf{x}_0 , \mathbf{T} , β and n (number of iterations)**for** $k = 0, \dots, n - 1$ **do** **if** gradient is truncated **then**

$$\begin{aligned} & \left| \nabla \tilde{f}(\mathbf{x}_k) = \mathbf{A}'_{\mathcal{T}} \dot{\psi}([\mathbf{A}\mathbf{x}_k]_{\mathcal{T}}; \mathbf{y}_{\mathcal{T}}, \mathbf{b}_{\mathcal{T}}) \right. \\ & \left. + \beta \mathbf{T}' \dot{h}(\mathbf{T}\mathbf{x}_k) \right. \end{aligned}$$

else

$$\left| \nabla \tilde{f}(\mathbf{x}_k) = \mathbf{A}' \dot{\psi}(\mathbf{A}\mathbf{x}_k; \mathbf{y}, \mathbf{b}) + \beta \mathbf{T}' \dot{h}(\mathbf{T}\mathbf{x}_k) \right.$$

end **if** cost function is regularized **then**

$$\left| \mu_k \leftarrow \text{Computed by (18)} \right.$$

else

$$\left| \mu_k \leftarrow \text{Computed by (14)} \right.$$

end

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mu_k \nabla f(\mathbf{x}_k)$$

end**Output:** \mathbf{x}_n

any improvement on the cost function value at convergence for various setting of a^h compared to WF (shown in the supplement), which is consistent with results in [7]. Furthermore, we found that the TWF can instead be computationally inefficient because it requires computing the truncated indices in each iteration, especially when both the iteration number and M are large.

4) *Summary:* Algorithm 1 summarizes the Wirtinger flow algorithm for the Poisson model that uses the observed Fisher information for the step size and the optional gradient truncation for noise reduction.

B. Majorize-Minimize (MM)

This section introduces our proposed MM algorithm with a quadratic majorizer using a novel curvature formula for the Poisson phase retrieval problem.

A majorize-minimize (MM) algorithm [54] is a generalization of the expectation-maximization (EM) algorithm that solves an optimization problem by iteratively constructing and solving simpler surrogate optimization problems. Quadratic majorizers are very common in MM algorithms because they have closed-form solutions and are well-suited to conjugate gradient methods.

The bounded curvature property derived in (9) enables us to derive an MM algorithm [55] with a quadratic majorizer for (7). Fig. 1 illustrates that one can construct a quadratic majorizer on \mathbb{R} for (8). With a bit more work to generalize to \mathbb{C}^N , a quadratic majorizer for the Poisson ML cost function (7) has the form

$$\begin{aligned} q(\mathbf{x}; \mathbf{x}_k) &\triangleq f(\mathbf{x}_k) + \text{real} \left\{ (\mathbf{x} - \mathbf{x}_k)' \mathbf{A}' \dot{\psi}(\mathbf{A}\mathbf{x}_k; \mathbf{y}, \mathbf{b}) \right\} \\ &+ \frac{1}{2} (\mathbf{x} - \mathbf{x}_k)' \mathbf{A}' \mathbf{W} \mathbf{A} (\mathbf{x} - \mathbf{x}_k), \end{aligned} \quad (20)$$

where \mathbf{W} denotes a diagonal curvature matrix. From (9), one choice of \mathbf{W} uses the maximum of $\ddot{\psi}$:

$$\mathbf{W}_{\max} \triangleq \text{diag}\{2 + \mathbf{y}/(4\mathbf{b})\} \in \mathbb{R}^{M \times M}. \quad (21)$$

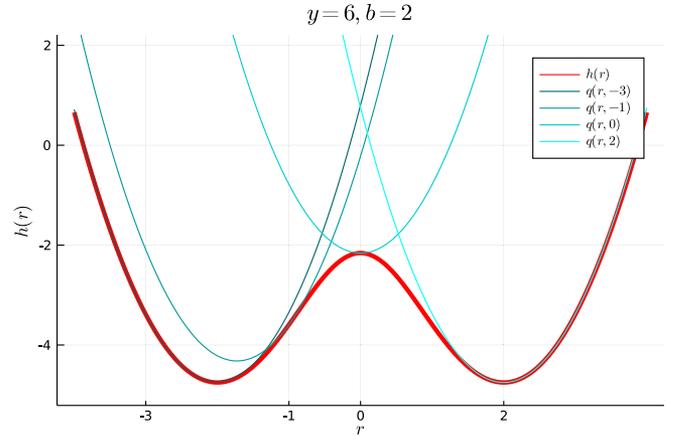


Fig. 1. Quadratic majorizers for the non-convex Poisson log-likelihood function $\phi(r; y, b)$ when $y = 6$ and $b = 2$.

However, \mathbf{W}_{\max} is suboptimal because the curvature of a quadratic majorizer of $\psi(v; \cdot)$ varies with $v = [\mathbf{A}\mathbf{x}_k]_i$. For example, when $|v| \rightarrow \infty$, then (7) is dominated by the quadratic term having curvature = 2; so if y is large and b is small, then \mathbf{W}_{\max} can be much greater than the optimal curvature 2. Thus, instead of using \mathbf{W}_{\max} to build majorizers, we propose to use the following improved curvature:

$$\begin{aligned} \mathbf{W}_{\text{imp}} &\triangleq \text{diag}\{c(\mathbf{A}\mathbf{x}_k; \mathbf{y}, \mathbf{b})\} \in \mathbb{R}^{M \times M}, \\ c(s; y, b) &\triangleq \begin{cases} \ddot{\psi}\left(\frac{b + \sqrt{b^2 + b|s|^2}}{|s|}; y, b\right), & s \neq 0, \\ 2, & s = 0. \end{cases} \end{aligned} \quad (22)$$

One can verify $\lim_{s \rightarrow 0} c(s; y, b) = 2$ so (22) is continuous over $s \in \mathbb{C}$. The next subsection proves that (22) provides a majorizer in (20) and is an improved curvature compared to \mathbf{W}_{\max} , though it is not necessarily the sharpest possible [56]; the sharpest (optimal) curvature $c_{\text{opt}}(s)$ in real case can be expressed as

$$c_{\text{opt}}(s) = \sup_{r \neq s} \frac{2 \left(\phi(r) - \phi(s) - \dot{\phi}(s)(r - s) \right)}{(r - s)^2}, \quad (23)$$

where $\phi(\cdot)$ is the marginal Poisson cost function defined in (8). However, (23) usually does not have a closed-form solution due to its transcendental derivative; while our \mathbf{W}_{imp} has a simpler form and is more efficient to compute. Fig. 2 visualizes the quadratic majorizer with different curvatures and the original Poisson cost function (7). We find the optimal curvature numerically by first discretizing r and then finding the supremum over all discrete segments.

For the ML case where constraints or regularizers are absent, the quadratic majorizer (20) associated with (21) or (22) leads to the following MM update:

$$\begin{aligned} \mathbf{x}_{k+1} &= \arg \min_{\mathbf{x} \in \mathbb{F}^N} q(\mathbf{x}; \mathbf{x}_k) \\ &= \mathbf{x}_k - (\mathbf{A}' \mathbf{W} \mathbf{A})^{-1} \mathbf{A}' \dot{\psi}(\mathbf{A}\mathbf{x}_k; \mathbf{y}, \mathbf{b}). \end{aligned} \quad (24)$$

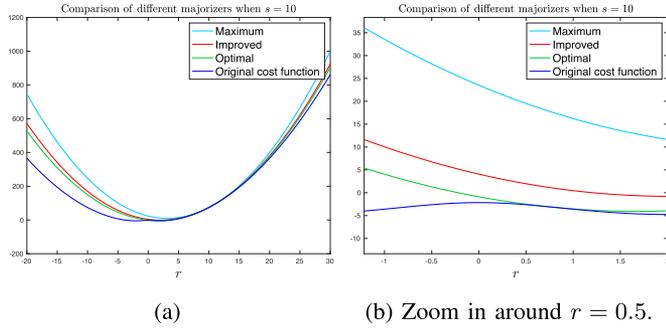


Fig. 2. Comparison of quadratic majorizers with maximum, improved and the optimal curvatures, for $y = 6$ and $b = 2$, visualized around $r = 0.5$. All three curves touch at the point $r = s = 10$ by construction.

If $\mathbf{x} \in \mathbb{R}^N$, then the MM update for \mathbf{x}_{k+1} is

$$\mathbf{x}_k - (\text{real}\{\mathbf{A}'\mathbf{W}\mathbf{A}\})^{-1} \text{real}\{\mathbf{A}'\dot{\psi}(\mathbf{A}\mathbf{x}_k; \mathbf{y}, \mathbf{b})\}.$$

When N is large, the matrix inverse operation in (24) is impractical, so we run a few inner iterations of conjugate gradient (CG) to descend the quadratic majorizer and hence descend the original cost function.

1) *Proof of the Proposed Curvature:* For simplicity, we drop the subscript i and irrelevant constants and focus on the negative log-likelihood for real case for simplicity as in (8).

One can generalize the majorizer derived here for (8) to the complex case by taking the magnitude and some other minor modifications.

First, we consider some simple cases:

- 1) If $y = 0$, then (8) is a quadratic function, so no quadratic majorizer is needed.
- 2) If $b = 0$ and $y > 0$ then (8) has unbounded 2nd derivative so no quadratic majorizer exists.
- 3) If $b = 0$ and $r = 0$, then y must be zero because a Poisson random variable with zero mean can only take the value 0. Thus again quadratic majorizer is not needed.

So hereafter we assume that $y > 0$, $b > 0$. Under these assumptions, the derivatives of (8) are:

$$\dot{\phi}(r) = 2r \left(1 - \frac{y}{r^2 + b} \right), \quad (25)$$

$$\ddot{\phi}(r) = 2 + 2y \frac{r^2 - b}{(r^2 + b)^2}, \quad (26)$$

$$\phi^{(3)}(r) = \frac{2yr(3b - r^2)}{(r^2 + b)^3}, \quad (27)$$

where $\phi^{(3)}(r)$ denotes the third derivative. Clearly, $\dot{\phi}(r)$ is convex on $(-\infty, -\sqrt{3b}]$ and $[0, \sqrt{3b}]$, and concave on $[-\sqrt{3b}, 0]$ and $[\sqrt{3b}, +\infty)$, based on the sign of $\phi^{(3)}(r)$.

A quadratic majorizer of $\phi(\cdot)$ at point s has the form:

$$\Phi(r; s) = \phi(s) + \dot{\phi}(s)(r - s) + \frac{1}{2}c(s)(r - s)^2. \quad (28)$$

The derivative of this function (w.r.t. r) is:

$$\dot{\Phi}(r; s) = c(s)(r - s) + \dot{\phi}(s). \quad (29)$$

By design, this kind of quadratic majorizer satisfies $\Phi(s; s) = \phi(s)$ and $\dot{\Phi}(s; s) = \dot{\phi}(s)$. From (27), we note that $r^2 = 3b$ is a maximizer of $\ddot{\phi}$ so the maximum curvature is:

$$\ddot{\phi}(r) \leq 2y \frac{2b}{(4b)^2} + 2 = 2 + \frac{y}{4b}. \quad (30)$$

Proposition: $\Phi(r; s)$ defined in (28) is a majorizer of $\phi(r)$ when $c(s) = c_{\text{imp}}(s)$, where:

$$c_{\text{imp}}(s) \triangleq \begin{cases} \ddot{\phi}(u(s)), & s \neq 0, \\ \lim_{s \rightarrow 0} \ddot{\phi}(u(s)), & s = 0, \end{cases} \quad (31)$$

where

$$u(s) \triangleq \frac{b + \sqrt{b^2 + bs^2}}{s}. \quad (32)$$

By construction, the proposed curvature $c(s)$ is at most the maximum curvature given in (30).

Proof: Because of the symmetry of $\ddot{\phi}(r)$, it suffices to prove the proposition for $s \geq 0$ without loss of generality. First we consider some trivial cases:

- 1) If $s = 0$, one can verify $\lim_{s \rightarrow 0} \ddot{\phi}(u(s)) = 2$. In this case, $\Phi(r; s)$ is simply

$$\begin{aligned} \Phi(r; 0) &= \phi(0) + \frac{1}{2}c(0)r^2 \\ &= r^2 + b - y \log(b) \\ &\geq r^2 + b - y \log(r^2 + b) = \phi(r). \end{aligned} \quad (33)$$

- 2) If $s = \sqrt{3b}$, one can verify

$$\ddot{\phi}(g(\sqrt{3b})) = 2 + \frac{y}{4b}, \quad (34)$$

which equals the maximum curvature.

Hereafter, we consider only $s > 0$ and $s \neq \sqrt{3b}$.

To proceed, it suffices to prove

$$\begin{aligned} \forall r \in (-\infty, s], \quad \dot{\phi}(r) &\geq \dot{\Phi}(r; s), \\ \forall r \in [s, +\infty), \quad \dot{\phi}(r) &\leq \dot{\Phi}(r; s), \end{aligned} \quad (35)$$

because if (35) holds, then $\forall \tilde{r} < s$:

$$\begin{aligned} \Phi(s; s) - \Phi(\tilde{r}; s) &= \int_{\tilde{r}}^s \dot{\Phi}(r; s) dr \\ &\leq \int_{\tilde{r}}^s \dot{\phi}(r) dr = \phi(s) - \phi(\tilde{r}), \end{aligned} \quad (36)$$

and $\forall \tilde{r} > s$:

$$\begin{aligned} \Phi(\tilde{r}; s) - \Phi(s; s) &= \int_s^{\tilde{r}} \dot{\Phi}(r; s) dr \\ &\geq \int_s^{\tilde{r}} \dot{\phi}(r) dr = \phi(\tilde{r}) - \phi(s). \end{aligned} \quad (37)$$

Together with $\Phi(s; s) = \phi(s)$, we have shown that (35) implies $\Phi(r; s) \geq \phi(r)$, $\forall r \in \mathbb{R}$.

Substituting $\dot{\Phi}(r; s) = c(s)(r - s) + \dot{\phi}(s)$ into (35), one can verify that showing (35) becomes showing

$$c_{\text{imp}}(s) \geq \frac{\dot{\phi}(r) - \dot{\phi}(s)}{r - s}, \quad \forall r \in \mathbb{R}, r \neq s. \quad (38)$$

Furthermore, when $s > 0$, the parabola $\Phi(\cdot; s)$ is symmetric about its minimizer:

$$\begin{aligned} \delta &= \delta(s) \triangleq \arg \min_r \Phi(r; s) = s - \frac{\dot{\phi}(s)}{c_{\text{imp}}(s)} \\ &= \frac{s \ddot{\phi}(u(s)) - \dot{\phi}(s)}{\ddot{\phi}(u(s))} \geq 0. \end{aligned} \quad (39)$$

This minimizer is nonnegative because $\dot{\phi}(s) \leq 2s$ and

$$\begin{aligned} c_{\text{imp}}(s) &= \ddot{\phi}(u(s)) = 2 + \frac{ys^2(b + \sqrt{b^2 + bs^2})}{b(b + s^2 + \sqrt{b^2 + bs^2})^2} \\ &\geq 2. \end{aligned} \quad (40)$$

Thus, if $\phi(r) \leq \Phi(r; s)$ when $r \geq 0$, we have $\phi(-r) = \phi(r) \leq \Phi(r; s) \leq \Phi(-r; s) = \Phi(r + 2\delta; s)$, so it suffices to prove (38) only for $r \geq 0$, which simplifies (38) to showing

$$c_{\text{imp}}(s) \geq \frac{\dot{\phi}(r) - \dot{\phi}(s)}{r - s}, \quad \forall r \geq 0, r \neq s. \quad (41)$$

In short, if (41) holds, then $\Phi(r; s) \geq \phi(r)$, $\forall r \in \mathbb{R}$.

To prove (41), we exploit a useful property of $c_{\text{imp}}(s)$. Under geometric view, $c_{\text{imp}}(s)$ defines (the ratio of) an affine function connecting points $(u(s), \dot{\phi}(u(s)))$ and $(s, \dot{\phi}(s))$ is tangent to $\dot{\phi}(r)$ at point $r = u(s)$, so that one can verify

$$\ddot{\phi}(u(s)) = c_{\text{imp}}(s) = \frac{\dot{\phi}(u(s)) - \dot{\phi}(s)}{u(s) - s}, \quad u(s) \neq s. \quad (42)$$

The reason why $u(s) \neq s$ is that one can verify $u(s) = s$ implies $s = \sqrt{3b}$ for $s > 0$ that has already been proved above.

Let $\xi(r) = (\dot{\phi}(r) - \dot{\phi}(s))/(r - s)$, where $r \geq 0$ and $r \neq s$, plugging in $\dot{\phi}(r)$ and $\dot{\phi}(s)$ yields:

$$\xi(r) = 2 + \frac{2y(sr - b)}{(s^2 + b)(r^2 + b)}. \quad (43)$$

Differentiating w.r.t. r leads to:

$$\dot{\xi}(r) = \frac{2y}{s^2 + b} \cdot \frac{-sr^2 + 2br + bs}{(r^2 + b)^2}, \quad (44)$$

where one can verify the positive root of $-sr^2 + 2br + bs = 0$ is $u(s)$ that is given by (32).

Together with $\dot{\xi}(r) > 0$ when $r \in (0, u(s))$ and $\dot{\xi}(r) < 0$ when $r \in (u(s), \infty)$, we have (41) holds because $\xi(r)$ achieves its maximum at $\xi(u(s))$:

$$\xi(r) \leq \xi(u(s)) = c_{\text{imp}}(s). \quad \blacksquare \quad (45)$$

2) *Regularized MM*: For the regularized cost function (16), one can use the quadratic majorizer (20) as a starting point. If the regularizer is prox-friendly, then the minimization step of an MM algorithm for the regularized optimization problem is

$$\mathbf{x}_{k+1} = \arg \min_{\mathbf{x} \in \mathbb{F}^N} q(\mathbf{x}; \mathbf{x}_k) + \beta \|\mathbf{T}\mathbf{x}\|_1. \quad (46)$$

To solve (46), one can apply proximal gradient methods [57], [58], [59]. We can use the proximal optimized gradient method (POGM) with adaptive restart [59] that provides faster worst-case convergence bound than the fast iterative shrinkage-thresholding algorithm (FISTA) [58].

Algorithm 2: MM Algorithm for the Poisson model.

Input: \mathbf{A} , \mathbf{y} , \mathbf{b} , \mathbf{x}_0 and n (number of iterations)

for $k = 0, \dots, n - 1$ **do**

 Build $q(\mathbf{x}; \mathbf{x}_k)$ (20) using \mathbf{W}_{imp} (22)

if cost function is regularized **then**

if \mathbf{T} is prox-friendly **then**

 | Update \mathbf{x}_k by (46) using POGM

else

 | Update \mathbf{x}_k by (47) using CG

end

else

 | Update \mathbf{x}_k by (24) or CG

end

end

Output: \mathbf{x}_n

For non-proximal friendly regularizers, we can “smooth” it using the Huber function (17), leading to the optimization problem of the form

$$\mathbf{x}_{k+1} = \arg \min_{\mathbf{x} \in \mathbb{F}^N} q(\mathbf{x}; \mathbf{x}_k) + \beta 1' h(\mathbf{T}\mathbf{x}; \alpha), \quad (47)$$

and we use nonlinear CG for this minimization, with step sizes based on Huber’s quadratic majorizer.

Algorithm 2 summarizes our MM algorithm with quadratic majorizer using the improved curvature (22).

III. IMPLEMENTATION DETAILS

This section introduces the implementation details of algorithms discussed in the previous section and our experimental setup for the numerical simulation (Section IV). We ran all algorithms on a server with Ubuntu 16.04 LTS operating system having Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20 GHz and 187 GB memory. All elements in the measurement vector \mathbf{y} were simulated to follow independent Poisson distributions per (6). All algorithms were implemented in Julia v1.7.3. All the timing results presented in Section IV were averaged across 10 independent test runs.

A. Initialization

Luo et al. [60] proposed the optimal initialization strategy under random Gaussian system matrix setting with Poisson noise. Since this paper focuses on very low-count regimes, the scale factor κ in [60] is a very small number so that $\mathbf{y} - \kappa \approx \mathbf{y}$. Therefore, we used $\tilde{\mathbf{x}}_0$, the leading eigenvector of $\mathbf{A}' \text{diag}\{\mathbf{y} \odot (\mathbf{y} + 1)\} \mathbf{A}$ (instead of $\mathbf{A}' \text{diag}\{(\mathbf{y} - \kappa) \odot (\mathbf{y} + 1)\} \mathbf{A}$) as an initial estimate of \mathbf{x} .

To accommodate signals of arbitrary scale, we scaled that leading eigenvector using a nonlinear least-square (LS) fit:

$$\hat{\alpha} = \arg \min_{\alpha \in \mathbb{R}} \|\mathbf{y} - \mathbf{b} - |\alpha \mathbf{A} \tilde{\mathbf{x}}_0|^2\|_2^2 = \frac{\sqrt{(\mathbf{y} - \mathbf{b})' |\mathbf{A} \tilde{\mathbf{x}}_0|^2}}{\|\mathbf{A} \tilde{\mathbf{x}}_0\|_4^2}. \quad (48)$$

Finally, our initial estimate is the element-wise absolute value of $\hat{\alpha} \mathbf{x}_0$ if \mathbf{x} is known to be real and nonnegative; and is $\hat{\alpha} \mathbf{x}_0$ otherwise.

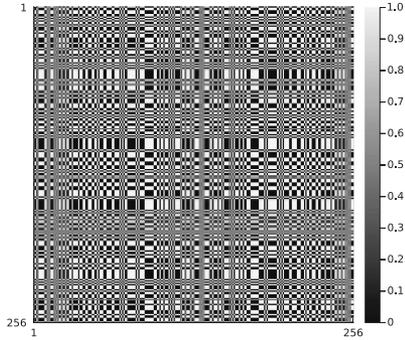


Fig. 3. Reference image from [31] used in HCDI and our canonical DFT experiments.

B. Ambiguities

To handle the global phase ambiguity, i.e., all the algorithms can recover the signal only to within a constant phase shift due to the loss of global phase information, before quantitative comparison, we corrected the phase of $\hat{\mathbf{x}}$ by

$$\hat{\mathbf{x}}_{\text{corrected}} \triangleq \text{sign}(\langle \hat{\mathbf{x}}, \mathbf{x} \rangle) \hat{\mathbf{x}}. \quad (49)$$

C. System Matrix and True Signals

1) *System Matrix*: We investigated 4 different choices for the system matrix \mathbf{A} : complex random Gaussian matrix (having 80000 rows), canonical DFT (with reference image), masked DFT matrix (with 20 masks) and a transmission matrix (ETM) that is acquired empirically through physical experiments [61], [62].

For the canonical DFT, we used a reference image as used in holographic coherent diffraction imaging (HCDI) [31], specifically, the measurements follow

$$\mathbf{y} \sim \text{Poisson}(|\mathcal{F}\{\mathbf{x}, \mathbf{0}, \mathcal{R}\}|^2 + \mathbf{b}), \quad (50)$$

where \mathcal{F} denotes discrete Fourier transform (DFT) and \mathcal{R} denotes a known reference image. This paper uses the reference image shown in Fig. 3, taken by screen shot from [31].

For the masked DFT case, the measurement vector \mathbf{y} in the Fourier phase retrieval problem has elements with means given by

$$\mathbb{E}[y[\tilde{n}]] = \left| \sum_{n=0}^{N-1} x[n] e^{-i2\pi n \tilde{n} / \tilde{N}} \right|^2 + b[\tilde{n}], \quad (51)$$

where $\tilde{N} = 2N - 1$ (here we consider the over-sampled case), and $\tilde{n} = 0, \dots, \tilde{N} - 1$. After introducing redundant masks, the measurement model becomes

$$\mathbb{E}[y_l[\tilde{n}]] = \left| \sum_{n=0}^{N-1} x[n] D_l[n] e^{-i2\pi n \tilde{n} / \tilde{N}} \right|^2 + b_l[\tilde{n}], \quad (52)$$

where $\mathbb{E}[\mathbf{y}_l] \in \mathbb{R}^{\tilde{N}}$ for $l = 1, \dots, L$ and D_l denotes the l th of L masks. Our experiment used $L = 21$ masks to define the overall system matrix $\mathbf{A} \in \mathbb{C}^{L\tilde{N} \times N}$, where the first mask has full sampling and the remaining 20 have sampling rate 0.5 with random sampling patterns.

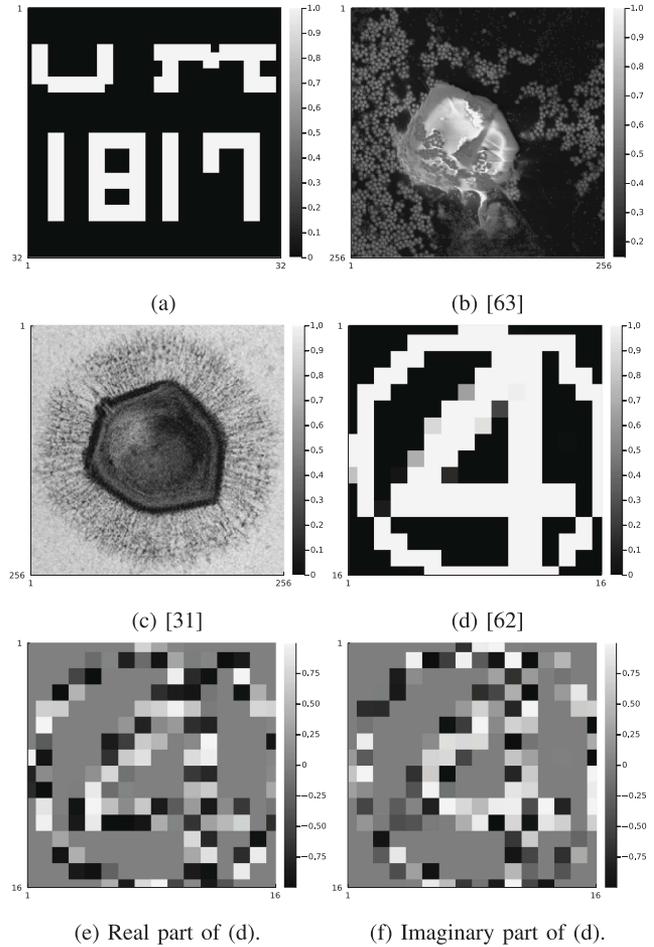


Fig. 4. True images used in the simulations. Subfigure (d) shows the magnitude of a complex image.

We scaled each system matrix by a constant factor such that the average count of measurement vector \mathbf{y} is 0.25, and the background count b is set to be 0.1.

2) *True Images*: We considered 4 images as the true images in our experiments Fig. 4 shows such images; (b) is from [63], (c) is from [31], (d)–(f) are from [62]. We used subfigure (a) for experiments with random Gaussian system matrix, (b) for masked DFT matrix, (c) for canonical DFT matrix and (d) for empirical transmission matrix, respectively.

IV. NUMERICAL SIMULATIONS RESULTS

A. Convergence Speed of WF With Fisher Information

This section compares convergence speeds, in terms of cost function vs. time and PSNR vs. time, between WF using our proposed Fisher information for step size, and empirical step size [15], backtracking line search [21], the optimal step size for the Gaussian ML cost function [18], and LBFQS quasi-Newton to approximate the Hessian in Newton's method [26]. The LBFQS algorithm was from the "Optim.jl" Julia package [64]. Fig. 5 shows that, for all system matrix choices, WF with Fisher information converged faster (in terms of decreasing the cost

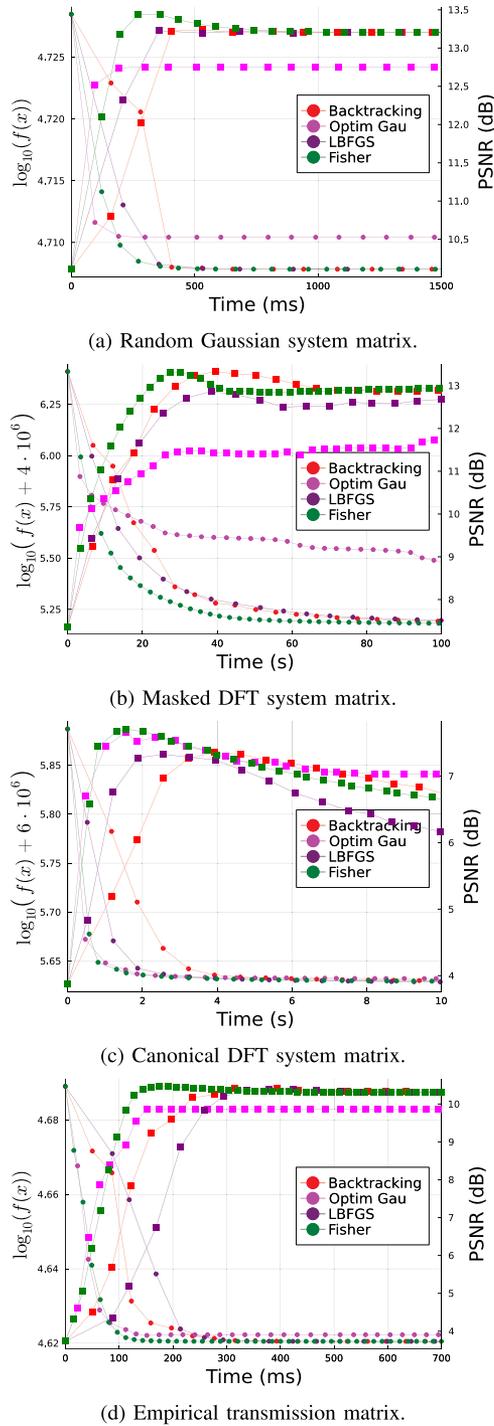


Fig. 5. Comparison of convergence speed for various WF methods and LBFGS under different system matrix settings. The “Optim Gau” curve is WF using the curvature from [18] that is optimal for Gaussian noise. The circle marker corresponds to the cost function and the square marker corresponds to PSNR.

function) than all other methods; the LBFGS algorithm had comparable convergence speed as WF with backtracking line search. We found that WF with the empirical step size did not converge using hyper-parameters in [15] so we excluded those results in Fig. 5. The backtracking approach, although slower than Fisher approach per wall-time, is faster per-iteration. However, the step

size found by backtracking line search could be sensitive to hyper-parameter choices. For the WF algorithm with optimal step size (derived based on Gaussian noise model [18]), we conjectured that it reached a non-stationary point that has larger cost function value than those of other methods, as expected.

In terms of PSNR, we found that in random Gaussian, masked DFT and empirical transmission cases, WF with Fisher information increased the PSNR faster than all other methods; WF with optimal Gaussian step size led to lower PSNR, perhaps again due to reaching a sub-optimal minimizer. However, for the canonical Fourier case, we found that all methods started decreasing PSNR after several iterations. The algorithms may be more sensitive to noise in the canonical Fourier matrix setting, especially in the very low-count regime considered here. Apparently WF with optimal Gaussian step size overfits the noise more slowly due to its sub-optimal step size under Poisson noise.

B. Comparison of Poisson and Gaussian Algorithms

This section compares the reconstruction quality, i.e., the NRMSE to the true signal, between WF derived from the Gaussian ML cost function (3), and WF derived from the Poisson ML cost function (7) as well as regularized WF under different system matrix settings. We used corner-rounded TV regularizer with $\beta = 32$ and $\alpha = 0.1$ in the regularized WF algorithm.

Fig. 6 shows that algorithms derived from the Poisson model yielded consistently better reconstruction quality (lower NRMSE) than algorithms derived from the Gaussian model, as expected. Furthermore, by incorporating regularizer that exploits the assumed property of the true signal, the NRMSE was further decreased. Spectral initialization worked well in random Gaussian matrix setting, but not for other system matrices, as expected from its theory. The WF Gaussian approach failed to reconstruct in masked and canonical DFT system matrix setting. Since incorporating appropriate regularizers helps algorithms yield higher quality reconstructions, a question is naturally raised about which regularized algorithm converges the fastest. The next subsection presents such comparisons.

C. Convergence Speed of Regularized Poisson Algorithms

As discussed in Section II, many algorithms can be modified to accommodate regularizers. We compared the convergence speeds of regularized Poisson algorithms (WF Fisher, WF backtracking, LBFGS, MM and ADMM [46]), with a smooth regularizer (corner-rounded TV), under different system matrix settings. Based on Fig. 5, we did not run simulations of regularized WF with empirical step size and with Gaussian optimal step size, respectively. For all other algorithms, we set the regularization parameters to be $\beta = 32$ and $\alpha = 0.1$ (defined in (16) and (17)).

Fig. 7 shows that the regularized WF with our proposed Fisher information for step size converged the fastest compared to other methods under all different system matrices. The LBFGS again had a comparable convergence speed as WF using backtracking line search. The MM algorithm with improved curvature, was slower in wall-time due to extra computation per iteration, but was faster per iteration due to its sharper curvature. In masked

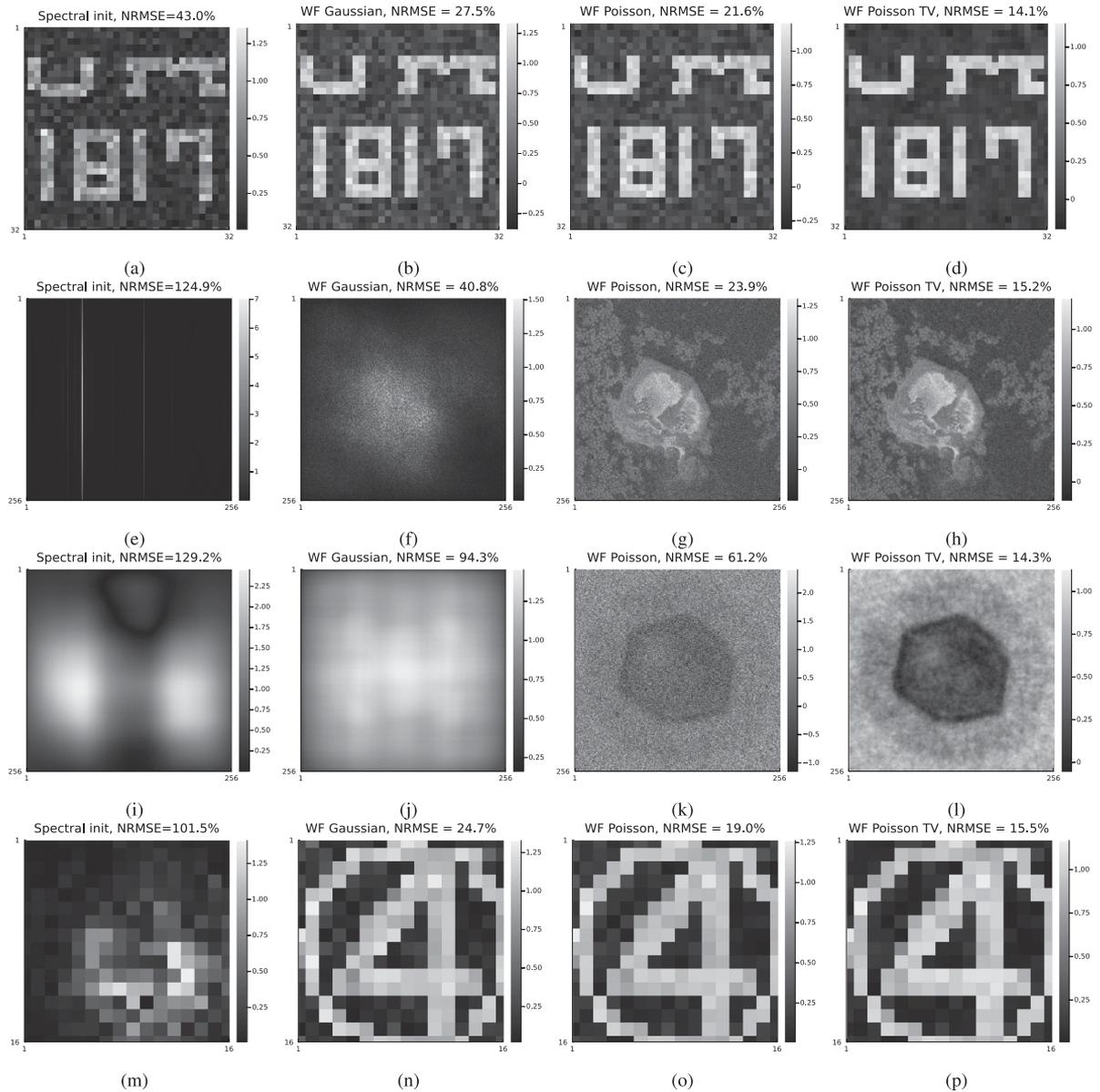


Fig. 6. Reconstruction quality comparison between four methods (left to right): the optimal Poisson spectral initialization [60], the WF Gaussian method, the WF Poisson method, and WF Poisson with TV regularization. System matrices: (a)–(d) random Gaussian; (e)–(h) masked DFT; (i)–(l) canonical DFT with reference image; (m)–(p) ETM. Magnitude of complex images shown. All WF algorithms used the proposed Fisher information for step size.

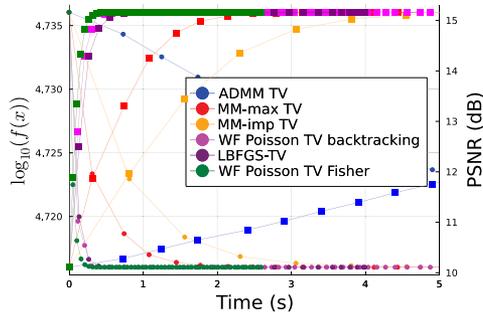
and canonical Fourier case, however, MM with improved curvature was faster than the maximum curvature in wall-time comparison, which can be attributed to large magnitude low frequency components in the coefficients of the Fourier transform.

V. DISCUSSION

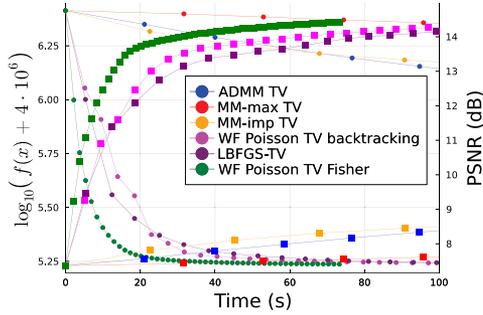
Current methods for phase retrieval mostly focus on ML estimation for Gaussian noise; fewer algorithms were derived for Poisson noise [6], [36], [43]. Here we proposed a novel WF algorithm and an MM algorithm and then did an empirical study on the convergence speed as well as reconstruction quality of several Poisson phase retrieval algorithms. In our proposed WF

algorithm, we first replaced the gradient term in Gaussian WF (4) with its Poisson counterpart (9). Then we did a quadratic approximation of the cost function and applied one iteration of Newton’s method to define an “optimal” step size. We then proposed to use the observed Fisher information to approximate the Hessian when computing the step size, which is a common method in computational statistics. Moreover, the Fisher information matrix is guaranteed to be positive semi-definite and is more computationally efficient compared to the Hessian. To further illustrate our proposed method of using Fisher information to approximate the Hessian, Fig. 8 visualizes these two matrices (in marginal forms).

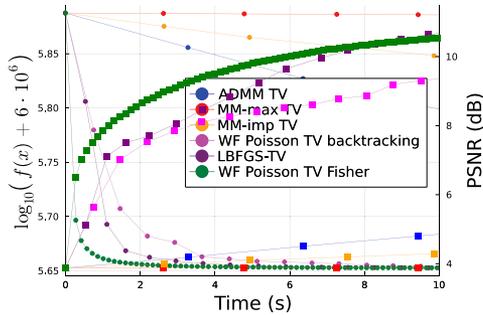
As shown in Fig. 8, the Hessian is noisy and can have some negative elements. Such undesirable features can lead to



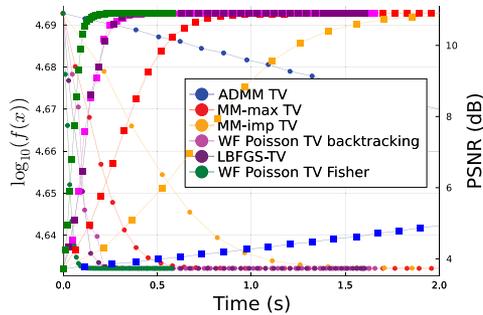
(a) Random Gaussian system matrix.



(b) Masked DFT system matrix.



(c) Canonical DFT system matrix.



(d) Empirical transmission matrix.

Fig. 7. Comparison of convergence speed of variant algorithms with corner-rounded TV regularizer. The circle marker corresponds to cost function and the square marker corresponds to PSNR.

unstable step size calculations. In contrast, the elements in Fisher information matrix are non-negative and less noisy. We ran some experiments and found that when the background counts b_i are large, using the noisy Hessian to calculate the step size can lead to divergence of the cost function, due to the negative values in the marginal second derivative. Setting such negative values in the second derivative to zero is a possible solution, but we found

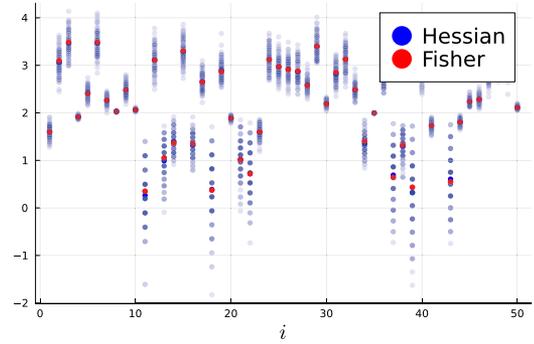


Fig. 8. Visualization of the marginal Hessian (9) and the marginal observed Fisher information (13). The horizontal axis denotes the i th element in the marginal Hessian/Fisher. Data were simulated with a random Gaussian matrix and 100 independent realizations.

that approach led to slower convergence than using the Fisher information. One potential alternative to our approach is to use the empirical Fisher information, but that may be suboptimal since the empirical Fisher information does not generally capture second-order information [65].

To accommodate our WF algorithm with non-smooth regularizers, e.g., $\|T\mathbf{x}\|_1$, we used a Huber function to approximate the ℓ_1 norm with a quadratic function around zero, so that the Wirtinger gradient is well-defined everywhere. A limitation of this paper is that we did not consider other regularizers in our experiments, though our algorithms can be generalized to handle other smooth regularizers with minor modifications. One drawback of TV regularization is that it assumes piece-wise uniform latent images so it lacks generalizability to other kinds of images. One way to address this is to train deep neural networks [66], [67] with a variety of images, potentially leading to better generalizability.

VI. CONCLUSION

This paper proposed and compared algorithms based on ML estimation and regularized ML estimation for phase retrieval from Poisson measurements, in very low-photon count regimes, e.g., 0.25 photon per pixel. We proposed a novel method that used the Fisher information to compute the step size in the WF algorithm; this approach eliminates all parameter tuning except the number of iterations. We also proposed a novel MM algorithm with improved curvature compared to the one derived from the upper bound of the second derivative of the cost function.

Simulation results experimented on random Gaussian matrix, masked DFT matrix, canonical DFT matrix and an empirical transmission matrix showed that: 1) For unregularized algorithms, the WF algorithm using our proposed Fisher information for step size converged faster than using empirical step size, backtracking line search, optimal step size for Gaussian noise model and LBFGS. Moreover, our proposed Fisher step size can be computed efficiently without any tuning parameter. 2) As expected, algorithms derived from the Poisson noise model produce consistently better reconstruction quality than algorithms derived from the Gaussian noise model for low-count

data. Furthermore, by incorporating regularizers that exploit the assumed properties of the true signal, the reconstruction quality can be further improved. 3) For regularized algorithms with smooth corner-rounded TV regularizer, WF with Fisher information converges faster than WF with backtracking line search, LBFSGS, MM and ADMM.

Future work includes precomputing and tabulating the optimal curvature for the quadratic majorizer, establishing sufficient conditions for global convergence, investigating algorithms with other kind of regularizers (e.g., deep learning methods [66], [67]), investigating sketching methods for large problem sizes [68], and testing Poisson phase retrieval algorithms under a wider variety of experimental settings.

REFERENCES

- [1] K. Jaganathan, Y. C. Eldar, and B. Hassibi, *Phase Retrieval: An Overview of Recent Developments*. Boca Raton, FL, USA: CRC Press, 2017. pp. 279–312.
- [2] P. Grohs, S. Koppensteiner, and M. Rathmair, “Phase retrieval: Uniqueness and stability,” *SIAM Rev.*, vol. 62, no. 2, pp. 301–350, 2020.
- [3] P. Jaming, “Phase retrieval techniques for radar ambiguity problems,” *J. Fourier Anal. Appl.*, vol. 5, no. 4, pp. 309–329, 1999.
- [4] R. P. Millane, “Phase retrieval in crystallography and optics,” *J. Opt. Soc. Amer. A*, vol. 7, no. 3, pp. 394–411, Mar. 1990.
- [5] J. Dainty and J. Fienup, “Phase retrieval and image reconstruction for astronomy,” *Imag. Recovery Theory Appl.*, vol. 13, pp. 231–275, 1987.
- [6] L. Bian et al., “Fourier ptychographic reconstruction using Poisson maximum likelihood and truncated Wirtinger gradient,” *Nature Sci. Rep.*, vol. 6, no. 1, 2016, pp. 1–10.
- [7] Y. Zhang, P. Song, and Q. Dai, “Fourier ptychographic microscopy using a generalized Anscombe transform approximation of the mixed Poisson-Gaussian likelihood,” *Opt. Exp.*, vol. 25, no. 1, pp. 168–179, Jan. 2017.
- [8] R. Xu, M. Soltanolkotabi, J. P. Haldar, W. Unglaub, J. Zusman, A. F. J. Levi, and R. M. Leahy, “Accelerated Wirtinger Flow: A fast algorithm for ptychography,” *arXiv: 1806.05546*, 2018, *arXiv:eess.IV*. [Online]. Available: <http://arxiv.org/abs/1806.05546>
- [9] X. Tian, “Fourier ptychographic reconstruction using mixed Gaussian-poisson likelihood with total variation regularisation,” *Electron. Lett.*, vol. 55, no. 19, pp. 1041–1043, Sep. 2019.
- [10] T. Lатычевскаиа, “Iterative phase retrieval in coherent diffractive imaging: Practical issues,” *Appl. Opt.*, vol. 57, no. 25, pp. 7187–7197, 2018.
- [11] D. L. Snyder, C. W. Helstrom, A. D. Lanterman, M. Faisal, and R. L. White, “Compensation for readout noise in CCD images,” *J. Opt. Soc. Amer. A*, vol. 12, no. 2, pp. 272–283, Feb. 1995.
- [12] E. J. Candès, T. Strohmer, and V. Voroninski, “Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming,” *Commun. Pure Appl. Math.*, vol. 66, no. 8, pp. 1241–1274, 2013.
- [13] A. S. Bandeira, J. Cahill, D. G. Mixon, and A. A. Nelson, “Saving phase: Injectivity and stability for phase retrieval,” *Appl. Comput. Harmon. Anal.*, vol. 37, no. 1, pp. 106–125, 2014.
- [14] A. S. Bandeira, Y. Chen, and D. G. Mixon, “Phase retrieval from power spectra of masked signals,” *Inf. Inference: A J. IMA*, vol. 3, no. 2, pp. 83–102, 2014.
- [15] E. J. Candès, Y. C. Eldar, T. Strohmer, and V. Voroninski, “Phase retrieval via matrix completion,” *SIAM J. Imag. Sci.*, vol. 6, no. 1, pp. 199–225, 2013.
- [16] Y. Shechtman, Y. C. Eldar, O. Cohen, H. N. Chapman, J. Miao, and M. Segev, “Phase retrieval with application to optical imaging: A contemporary overview,” *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 87–109, May 2015.
- [17] E. Candès, X. Li, and M. Soltanolkotabi, “Phase retrieval via Wirtinger flow: Theory and algorithms,” *IEEE Trans. Inf. Theory*, vol. 61, no. 4, pp. 1985–2007, Apr. 2015.
- [18] X. Jiang, S. Rajan, and X. Liu, “Wirtinger flow method with optimal stepsize for phase retrieval,” *IEEE Signal Process. Lett.*, vol. 23, no. 11, pp. 1627–1631, Nov. 2016.
- [19] T. T. Cai, X. Li, and Z. Ma, “Optimal rates of convergence for noisy sparse phase retrieval via thresholded Wirtinger flow,” *Ann. Stat.*, vol. 44, no. 5, pp. 2221–2251, 2016.
- [20] M. Soltanolkotabi, “Structured signal recovery from quadratic measurements: Breaking sample complexity barriers via nonconvex optimization,” *IEEE Trans. Inf. Theory*, vol. 65, no. 4, pp. 2374–2400, Apr. 2019.
- [21] T. Qiu, P. Babu, and D. P. Palomar, “PRIME: Phase retrieval via majorization-minimization,” *IEEE Trans. Signal Process.*, vol. 64, no. 19, pp. 5174–5186, Oct. 2016.
- [22] R. W. Gerchberg and W. O. Saxton, “Practical algorithm for determination of phase from image and diffraction plane pictures,” *OPTIK*, vol. 35, no. 2, pp. 237–246, 1972.
- [23] P. Netrapalli, P. Jain, and S. Sanghavi, “Phase retrieval using alternating minimization,” *IEEE Trans. Signal Process.*, vol. 63, no. 18, pp. 4814–4826, Sep. 2015.
- [24] I. Waldspurger, “Phase retrieval with random Gaussian sensing vectors by alternating projections,” *IEEE Trans. Inf. Theory*, vol. 64, no. 5, pp. 3301–3312, May 2018.
- [25] B. Gao and Z. Xu, “Phaseless recovery using the Gauss-Newton method,” *IEEE Trans. Signal Process.*, vol. 65, no. 22, pp. 5885–5896, Nov. 2017.
- [26] L. Ji and Z. Tie, “On gradient descent algorithm for generalized phase retrieval problem,” in *Proc. IEEE 13th Int. Conf. Signal Process.*, 2016, pp. 320–325.
- [27] J. Liang, P. Stoica, Y. Jing, and J. Li, “Phase retrieval via the alternating direction method of multipliers,” *IEEE Signal Process. Lett.*, vol. 25, no. 1, pp. 5–9, Jan. 2018.
- [28] Y. Yang, M. Pesavento, Y. C. Eldar, and B. Ottersten, “Parallel coordinate descent algorithms for sparse phase retrieval,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 7670–7674.
- [29] P. Thibault and M. Guizar-Sicairos, “Maximum-likelihood refinement for coherent diffractive imaging,” *New J. Phys.*, vol. 14, no. 6, Jun. 2012, Art. no. 063004.
- [30] A. Goy, K. Arthur, S. Li, and G. Barbastathis, “Low photon count phase retrieval using deep learning,” *Phys. Rev. Lett.*, vol. 121, no. 24, Dec. 2018, Art. no. 243902.
- [31] D. A. Barmherzig and J. Sun, “Low-photon holographic phase retrieval,” in *Imaging and Applied Optics Congress*, OSA Technical Digest (Optica Publishing Group, 2020), 2020, Paper JT4A.6.
- [32] I. Vazquez, I. E. Harmon, J. C. R. Luna, and M. Das, “Quantitative phase retrieval with low photon counts using an energy resolving quantum detector,” *J. Opt. Soc. Amer. A*, vol. 38, no. 1, pp. 71–79, Jan. 2021.
- [33] H. Lawrence, D. A. Barmherzig, H. Li, M. Eickenberg, and M. Gabriele, “Phase retrieval with holography and untrained priors: Tackling the challenges of low-photon nanoscale imaging,” 2020. [Online]. Available: <http://arxiv.org/abs/2012.07386>
- [34] A. Gnanasambandam and S. H. Chan, “Image classification in the dark using quanta image sensors,” in *Proc. Comput. Vis. 16th Eur. Conf.*, Glasgow, U.K., Springer-Verlag, Berlin, Heidelberg, 2020, pp. 484–501. [Online]. Available: https://doi.org/10.1007/978-3-030-58598-3_29
- [35] K. Choi and A. D. Lanterman, “Phase retrieval from noisy data based on minimization of penalized l_1 -divergence,” *J. Opt. Soc. Amer. A*, vol. 24, no. 1, pp. 34–49, Jan. 2007.
- [36] Y. Chen and E. J. Candès, “Solving random quadratic systems of equations is nearly as easy as solving linear systems,” *Commun. Pure Appl. Math.*, vol. 70, no. 5, pp. 822–883, May 2017.
- [37] H. Chang and S. Marchesini, “Denoising poisson phaseless measurements via orthogonal dictionary learning,” *Opt. Exp.*, vol. 26, no. 16, pp. 19773–19796, Aug. 2018.
- [38] I. Kang, F. Zhang, and G. Barbastathis, “Phase extraction neural network (PhENN) with coherent modulation imaging (CMI) for phase retrieval at low photon counts,” *Opt. Exp.*, vol. 28, no. 15, pp. 21578–21600, Jul. 2020.
- [39] D. L. Snyder, A. M. Hammoud, and R. L. White, “Image recovery from data acquired with a charge-coupled-device camera,” *J. Opt. Soc. Amer. A*, vol. 10, no. 5, pp. 1014–1023, May 1993.
- [40] M. Makitalo and A. Foi, “Optimal inversion of the generalized anscombe transformation for poisson-Gaussian noise,” *IEEE Trans. Imag. Process.*, vol. 22, no. 1, pp. 91–103, Jan. 2013.
- [41] G. Fatima and P. Babu, “PGPAL: A monotonic iterative algorithm for phase-retrieval under the presence of poisson-gaussian noise,” *IEEE Signal Process. Lett.*, vol. 29, pp. 533–537, 2022.
- [42] H. Zhang, Y. Liang, and Y. Chi, “A nonconvex approach for phase retrieval: Reshaped Wirtinger flow and incremental algorithms,” *J. Mach. Learn. Res.*, vol. 18, no. 141, pp. 1–35, 2017. [Online]. Available: <http://jmlr.org/papers/v18/16-572.html>
- [43] H. Chang, Y. Lou, Y. Duan, and S. Marchesini, “Total variation-based phase retrieval for poisson noise removal,” *SIAM J. Imag. Sci.*, vol. 11, no. 1, pp. 24–55, 2018.

- [44] G. Fatima, Z. Li, A. Arora, and P. Babu, "PDMM: A novel primal-dual Majorization-minimization algorithm for poisson phase-retrieval problem," *IEEE Trans. Signal Process.*, vol. 70, pp. 1241–1255, 2022.
- [45] Z. Li, K. Lange, and J. A. Fessler, "Poisson phase retrieval with Wirtinger flow," in *Proc. IEEE Int. Conf. Image Process.*, 2021, pp. 2828–2832.
- [46] Z. Li, K. Lange, and J. A. Fessler, "Algorithms for poisson phase retrieval," 2021. [Online]. Available: <https://arxiv.org/abs/2104.00861>
- [47] H. Zhang and D. P. Mandic, "Is a complex-valued stepsize advantageous in complex-valued gradient learning algorithms?," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2730–2735, Dec. 2016.
- [48] D. M. Titterton, "Recursive parameter estimation using incomplete data," *J. Roy. Statist. Soc. Ser. B*, vol. 46, no. 2, pp. 257–267, 1984. [Online]. Available: <http://www.jstor.org/stable/2345509>
- [49] K. Lange, "A gradient algorithm locally equivalent to the EM algorithm," *J. Roy. Statist. Soc. Ser. B*, vol. 57, no. 2, pp. 425–437, 1995. [Online]. Available: <http://www.jstor.org/stable/2345971>
- [50] L. Meng, "Method for computation of the fisher information matrix in the expectation-maximization algorithm," in *Proc. 51st Annu. Conf. Inf. Sci. Syst.*, 2017, pp. 1–6, doi: [10.1109/CISS.2017.7926126](https://doi.org/10.1109/CISS.2017.7926126).
- [51] M. R. Osborne, "Fisher's method of scoring," *Int. Stat. Rev.*, vol. 60, no. 1, pp. 99–117, 1992. [Online]. Available: <http://www.jstor.org/stable/1403504>
- [52] H. Hudson and J. Ma, "Fisher's method of scoring in statistical image reconstruction: Comparison of Jacobi and Gauss-Seidel iterative schemes," *Stat. Methods Med. Res.*, vol. 3, no. 1, pp. 41–61, 1994.
- [53] P. J. Huber, *Robust Statistics*. New York, NY, USA: Wiley, 1981.
- [54] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *Amer. Statist.*, vol. 58, no. 1, pp. 30–37, Feb. 2004.
- [55] D. Böhning and B. G. Lindsay, "Monotonicity of quadratic approximation algorithms," *Ann. Inst. Statist. Math.*, vol. 40, no. 4, pp. 641–663, Dec. 1988.
- [56] J. de Leeuw and K. Lange, "Sharp quadratic majorization in one dimension," *Comput. Statist. Data Anal.*, vol. 53, no. 7, pp. 2471–2484, May 2009.
- [57] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, Nov. 2004.
- [58] A. Beck and M. Teboulle, "A fast iterative Shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.
- [59] D. Kim and J. A. Fessler, "Adaptive restart of the optimized gradient method for convex optimization," *J. Optim. Theory Appl.*, vol. 178, no. 1, pp. 240–263, Jul. 2018.
- [60] W. Luo, W. Alghamdi, and Y. M. Lu, "Optimal spectral initialization for signal recovery with applications to phase retrieval," *IEEE Trans. Signal Process.*, vol. 67, no. 9, pp. 2347–2356, May 2019.
- [61] R. Chandra, Z. Zhong, J. Hontz, V. McCulloch, C. Studer, and T. Goldstein, "Phasepack: A phase retrieval library," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, 2017, pp. 1617–1621.
- [62] C. A. Metzler, M. K. Sharma, S. Nagesh, R. G. Baraniuk, O. Cossairt, and A. Veeraraghavan, "Coherent inverse scattering via transmission matrices: Efficient phase retrieval algorithms and a public dataset," in *Proc. Int. Conf. Comput. Photogr.*, 2017, pp. 1–16.
- [63] S. Marchesini, "Ptychography gold ball example dataset," 2017, doi: [10.11577/1454414](https://doi.org/10.11577/1454414).
- [64] P. K. Mogensen and A. N. Riseth, "Optim: A mathematical optimization package for julia," *J. Open Source Softw.*, vol. 3, no. 24, 2018, Art. no. 615.
- [65] F. Kunstner, P. Hennig, and L. Balles, "Limitations of the empirical fisher approximation for natural gradient descent," in *Proc. Adv. Neural Info. Proc. Sys.*, 2019, vol. 32, pp. 4156–4167.
- [66] T. Remez, O. Litany, R. Giryes, and A. M. Bronstein, "Class-aware fully convolutional Gaussian and poisson denoising," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5707–5722, Nov. 2018.
- [67] Z. Zha, B. Wen, X. Yuan, J. Zhou, and C. Zhu, "Simultaneous nonlocal low-rank and deep priors for poisson denoising," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 2320–2324.
- [68] Y. Luo, W. Huang, X. Li, and A. R. Zhang, "Recursive importance sketching for rank constrained least squares: Algorithms and high-order convergence," 2022. [Online]. Available: <http://arxiv.org/abs/2011.08360>

Supplement of “Poisson Phase Retrieval in Very Low-count Regimes”

Zongyu Li, *Student Member, IEEE*, Kenneth Lange and Jeffrey A. Fessler, *Fellow, IEEE*

A. WF with gradient truncation

Reference [1] shows that under Poisson noisy measurement setting, the relative error (RE) is almost monotonically decreasing as the tuning parameter a^h increases. Here we found a similar trend. Fig. 1 shows that under all three different system matrix settings, the Poisson ML cost function values of TWF is also monotonically decreasing as a^h increases; and are consistently higher than those of non-truncated WF. Moreover, we found that TWF can significantly increase the computational time compared to the non-truncated WF for random Gaussian and ETM matrix settings; for masked DFT matrix setting, both versions of WF showed comparable running time, presumably due to the efficient implementation of fast Fourier transform (FFT). Based on the results presented in Fig. 1, we did not use TWF for the rest simulations in this paper. And we will focus on incorporating regularization for potential reconstruction quality improvement.

B. Derivation of an ADMM algorithm for the Poisson phase retrieval problem

1) *ADMM for Poisson ML problem:* The ML estimation of \mathbf{x} for the Poisson noise model corresponds to the following optimization problem

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathbb{F}^N} f(\mathbf{x}), \quad f(\mathbf{x}) \triangleq \sum_i \psi(\mathbf{a}'_i \mathbf{x}; y_i, b_i),$$

$$\psi(v; y, b) \triangleq (|v|^2 + b) - y \log(|v|^2 + b). \quad (1)$$

With variable splitting $v_i = \mathbf{a}'_i \mathbf{x}$, an augmented linearized Lagrangian of (1) when $b_i = 0$ is given by

$$L_\rho(\mathbf{v}, \mathbf{x}; \boldsymbol{\eta}, \rho) = \sum_i \left[(|v_i|^2) - y_i \log(|v_i|^2) \right] + \frac{\rho}{2} \sum_i \left[|v_i - \mathbf{a}'_i \mathbf{x} + \eta_i|^2 - |\eta_i|^2 \right], \quad (2)$$

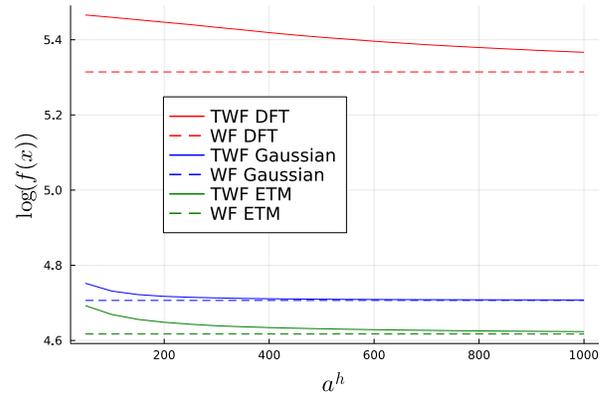
where $\boldsymbol{\eta}$ is the dual variable and $\rho > 0$ denotes the AL penalty parameter.

Zongyu Li and Jeffrey A. Fessler are with Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109-2122 (e-mail: zonyul@umich.edu, fessler@umich.edu).

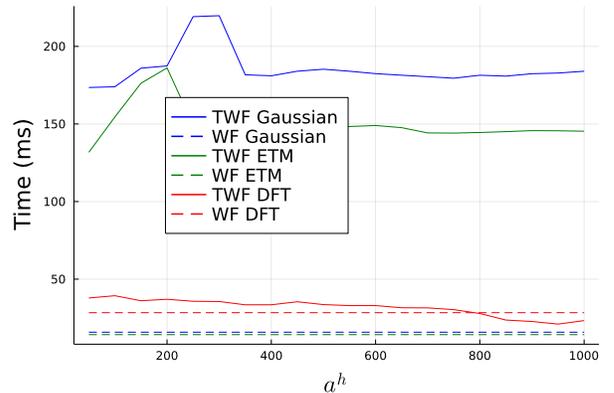
Kenneth Lange is with Departments of Computational Medicine, Human Genetics, and Statistics, University of California, Los Angeles, CA 90095 (e-mail: klange@ucla.edu).

Research supported in part by USPHS grants GM53275 and HG006139, and by NSF Grant IIS 1838179 and NIH R01 EB022075.

For clarification of notations, please refer to the main text.



(a) Cost function value (logarithm) of WF vs. TWF.



(b) Running time of WF vs. TWF.

Fig. 1: Comparison of TWF and WF under random Gaussian, masked DFT and empirical transmission matrix settings. The logarithm of the Poisson ML cost function was taken with base 10. All algorithms were ran 200 iterations.

We followed the order of first updating \mathbf{v} , then \mathbf{x} and finally $\boldsymbol{\eta}$. Because the updates for the phase and magnitude of \mathbf{v} are separable; in particular, the update for the phase of \mathbf{v} is

$$\text{sign}(\mathbf{v}_{k+1}) = \text{sign}(\mathbf{A}\mathbf{x}_k - \boldsymbol{\eta}_k). \quad (3)$$

To update the magnitude of \mathbf{v} , similar to [2], set $t_i = |\mathbf{a}'_i \mathbf{x}_k - \eta_i|$; then the update of a single component $|v_i|$ is given by the following positive solution of a quadratic polynomial:

$$|v_i| = \frac{\rho t_i + \sqrt{\rho^2 t_i^2 + 8y_i(2 + \rho)}}{2(2 + \rho)}. \quad (4)$$

Again, in the unregularized case, the \mathbf{x} update is a simple least square (LS) problem that can be optimized by CG or the

following operation (5) involving matrix inverse.

$$\mathbf{x}_{k+1} = (\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'(\mathbf{v}_{k+1} + \boldsymbol{\eta}_k), \quad (5)$$

Again, if $\mathbf{x} \in \mathbb{R}^N$, then the \mathbf{x} update is

$$\mathbf{x}_{k+1} = (\text{real}\{\mathbf{A}'\mathbf{A}\})^{-1}\text{real}\{\mathbf{A}'(\mathbf{v}_{k+1} + \boldsymbol{\eta}_k)\}.$$

2) *Regularized ADMM*: For the regularized case, the \mathbf{x} update becomes

$$\mathbf{x}_{k+1} = \arg \min_{\mathbf{x} \in \mathbb{F}^N} \frac{\rho}{2} \|\mathbf{A}\mathbf{x} - \mathbf{v}_{k+1} - \boldsymbol{\eta}_k\|_2^2 + \beta R(\mathbf{x}). \quad (6)$$

We solve this using CG or POGM, depending on whether the regularizer R is smooth or not.

The dual variable $\boldsymbol{\eta}$ ascent update [3] is simply

$$\boldsymbol{\eta}_{k+1} = \boldsymbol{\eta}_k + (\mathbf{v}_{k+1} - \mathbf{A}\mathbf{x}_{k+1}). \quad (7)$$

For the case $b_i > 0$, everything is the same as the case $b_i = 0$ except the update for $|v_i|$, for which one can verify the $|v_i|$ is instead a positive root of the following cubic polynomial

$$0 = (2 + \rho)|v_i|^3 - \rho t_i |v_i|^2 + (2b_i - 2y_i + \rho b_i)|v_i| - \rho b_i t_i. \quad (8)$$

Owing to the intermediate value theorem and the non-negativity of ρ , b_i , t_i , this cubic (8) can only possess one or three positive real roots. If the cubic has one positive root, then the update of $|v_i|$ is simply to assign the single positive root. If the cubic has three positive roots, we choose the root that minimizes the following Lagrangian term based on (9):

$$(|v_i|^2 + b_i) - y_i \log(|v_i|^2 + b_i) + \frac{\rho}{2} (|v_i| - t_i)^2. \quad (9)$$

A natural extension is to vary AL penalty parameter along with the variable update every few iterations. Reference [3] considered the magnitude of primal residual $\mathbf{r}_{k+1} = \mathbf{A}\mathbf{x}_{k+1} - \mathbf{v}_{k+1}$ and dual residual $\mathbf{s}_{k+1} = \rho\mathbf{A}'(\mathbf{v}_{k+1} - \mathbf{v}_k)$, as a principle to select penalty parameter to potentially improve convergence for convex optimization problems. However, for non-convex problems, the penalty parameter ρ is preferred to be sufficiently large to enable the convergence of ADMM algorithm [4]. Thus, we used the following heuristic to update ρ every 10 iterations:

$$\rho_{k+1} = \begin{cases} 2\rho_k, & \|r_k\| > 10\|s_k\| \\ \rho_k/2, & \|s_k\| > 100\rho_k\|r_k\| \\ \rho_k, & \text{otherwise.} \end{cases} \quad (10)$$

Algorithm 1 summarizes the ADMM algorithm described above.

C. Computation Time Comparison Between CG and BS

Fig. 2 compares the convergence rates of GS, MM and ADMM that involve solving least squares problems. In Fig. 2, $M = 3000$ and $N = 100$ are both small enough so that both backsubstitution (BS) (backslash in Julia) and CG are viable options for solving the inner quadratic optimization problems. In every case the CG version of 250 iterations decreased NRMSE faster than the BS version.

Algorithm 1: ADMM algorithm for the Poisson model

Input: \mathbf{A} , \mathbf{y} , \mathbf{b} , \mathbf{x}_0 and n (number of iterations)

Initialize:

$\mathbf{v}_0 \leftarrow \mathbf{A}\mathbf{x}_0$

$\boldsymbol{\eta}_0 \leftarrow \mathbf{v}_0 - \mathbf{A}\mathbf{x}_0$

for $k = 0, \dots, n - 1$ **do**

 Update $\text{sign}(\mathbf{v}_k)$ by (3)

if $b_i = 0$ **then**

 | Update $|v_k|$ by (4)

else

 | Update $|v_k|$ by selecting root based on (9)

end

if cost function is regularized **then**

 | Update \mathbf{x}_k by (6) using CG or POGM

else

 | Update \mathbf{x}_k by (5) or CG

end

 Update $\boldsymbol{\eta}_k$ by (7)

end

Output: \mathbf{x}_n

However, when \mathbf{A} is a DFT matrix, one can verify that $\mathbf{A}'\mathbf{A}$ is a diagonal matrix that is trivial to invert. In this case we used matrix inverse rather than CG to solve the least squares problems.

D. Huber Function vs. Alternating Minimization

For non-prox-friendly regularizers, other than modifying by the Huber function, an alternative can be introducing another variable and applying alternating minimization. In particular, The update of \mathbf{x} in LSMM becomes

$$\mathbf{x}_{k+1}, \mathbf{z}_{k+1} = \arg \min_{\mathbf{x} \in \mathbb{F}^N, \mathbf{z} \in \mathbb{C}^K} Q_k(\mathbf{x}, \mathbf{z}),$$

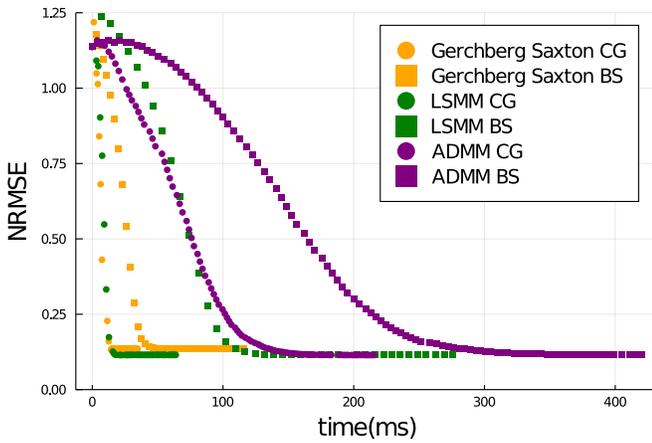
$$Q_k(\mathbf{x}, \mathbf{z}) \triangleq q(\mathbf{x}; \mathbf{x}_k) + \beta \left(\frac{1}{2} \|\mathbf{T}\mathbf{x} - \mathbf{z}\|_2^2 + \alpha \|\mathbf{z}\|_1 \right),$$

where one can alternatively update \mathbf{x} and \mathbf{z} . The \mathbf{x} update uses the closed-form solution that involves matrix inverse or conjugate gradient. The \mathbf{z} update is simply a soft-thresholding operation.

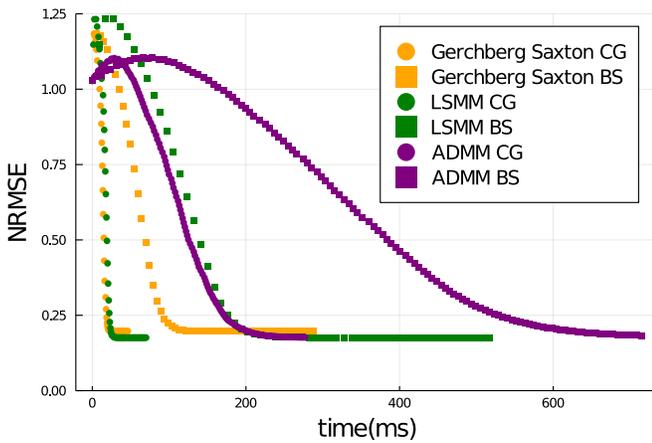
Similarly, the update of \mathbf{x} in ADMM is

$$\mathbf{x}_{k+1}, \mathbf{z}_{k+1} = \arg \min_{\mathbf{x} \in \mathbb{F}^N, \mathbf{z} \in \mathbb{C}^K} \frac{\rho}{2} \|\mathbf{A}\mathbf{x} - \mathbf{v}_{k+1} - \boldsymbol{\eta}_k\|_2^2 + \beta \left(\frac{1}{2} \|\mathbf{T}\mathbf{x} - \mathbf{z}\|_2^2 + \alpha \|\mathbf{z}\|_1 \right).$$

We compared these two approaches (Huber function vs. alternating minimization) and found that using Huber function was more efficient than alternating minimization, as evident in Fig. 3.



(a)



(b)

Fig. 2: NRMSE vs. time (ms) using different update strategy. Here \mathbf{A} is random Gaussian with the average of $\mathbf{a}_i^T \mathbf{x}$ equals to 2, $M = 3000$ and $b_i = 0.5$. Subfigure (a) and (b) correspond to experiments on a real signal and a complex signal, respectively.

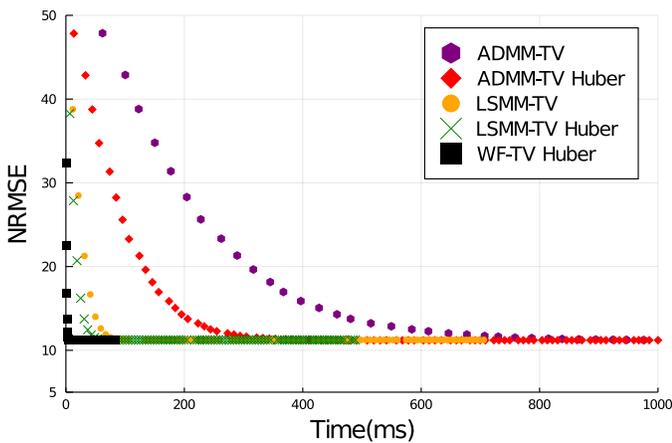


Fig. 3: Speed comparison of WF, LSMM and ADMM using Huber function and alternating minimization for regularization. LSMM-TV and ADMM-TV denote the alternating minimization approach.

REFERENCES

- [1] L. Bian, J. Suo, J. Chung, X. Ou, C. Yang, F. Chen, and Q. Dai. “Fourier ptychographic reconstruction using Poisson maximum likelihood and truncated Wirtinger gradient”. In: *Nature Sci. Rep.* 6.1 (2016).
- [2] H. Chang, Y. Lou, Y. Duan, and S. Marchesini. “Total variation-based phase retrieval for Poisson noise removal”. In: *SIAM J. Imag. Sci.* 11.1 (2018), pp. 24–55.
- [3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers”. In: *Found. Trends Mach. Learn.* 3.1 (Jan. 2011), pp. 1–122.
- [4] Y. Wang, W. Yin, and J. Zeng. “Global convergence of ADMM in nonconvex nonsmooth optimization”. In: *J. Sci. Comp.* 78.1 (2019), 29–63.