

# PhasePack: A Phase Retrieval Library

<https://www.cs.umd.edu/~tomg/projects/phasepack/>

Rohan Chandra  
Dept. of Computer Science  
University of Maryland  
College Park, USA  
rohan@cs.umd.edu

Tom Goldstein  
Dept. of Computer Science  
University of Maryland  
College Park, USA  
tomg@cs.umd.edu

Christoph Studer  
Dept. of ECE  
Cornell University  
Ithaca, NY  
studer@cornell.edu

**Abstract**—Phase retrieval deals with the estimation of complex-valued signals solely from the magnitudes of linear measurements. While there has been a recent explosion in the development of phase retrieval algorithms, the lack of a common interface has made it difficult to compare new methods against the state-of-the-art. The purpose of PhasePack is to create a common software interface for a wide range of phase retrieval algorithms and to provide a common testbed using both synthetic data and empirical imaging datasets. PhasePack is able to benchmark a large number of recent phase retrieval methods against one another to generate comparisons using a range of different performance metrics. The software package handles single method testing as well as multiple method comparisons.

The algorithm implementations in PhasePack differ slightly from their original descriptions in the literature in order to achieve faster speed and improved robustness. In particular, PhasePack uses adaptive stepsizes, line-search methods, and fast eigensolvers to speed up and automate convergence.

## I. INTRODUCTION

Classical linear inverse problems recover an unknown signal  $\mathbf{x} \in \mathbb{R}^n$  from  $m$  linear measurements of the form  $\mathbf{Ax} = \mathbf{b}$ , where  $\mathbf{A} \in \mathbb{C}^{m \times n}$  is a (known) linear measurement operator and  $\mathbf{b} \in \mathbb{C}^m$  is a (possibly noisy) vector of measurements. In contrast, phase retrieval [12], [20], [30] is the recovery of signals from the *magnitudes* of linear measurements  $|\mathbf{Ax}|$ , where  $|\cdot|$  denotes the entry-wise absolute value of the (possibly complex-valued) measurements. Phase retrieval is prevalent in the physical sciences where detectors can measure only the magnitude (brightness or intensity) of an electromagnetic wave, but not its phase. Such modalities include optical imaging [4], [25], quantum state tomography, electron microscopy [26], crystallography, astronomy [12], and diffraction imaging [30].

Phase Retrieval is an example of a non-convex quadratic program with quadratic constraints. In the real-valued case, it is a combinatorial problem of determining the missing signs of  $\mathbf{Ax}$ , which is known to be NP-hard [29]. Despite this observation, recent years have seen the development of new algorithms that solve phase retrieval problems effectively. Unfortunately, because of the lack of publicly available real-world data, the lack of a common software interface for different algorithms, and a knowledge gap between practitioners and theoreticians, only little work has been devoted to compare and evaluate newer phase retrieval methods.

## II. INTRODUCING PHASEPACK

PhasePack is a software package that contains implementations of many different phase retrieval methods, and tools for easily applying them to real and synthetic datasets. The purpose of PhasePack is to create a common interface for a wide range of phase retrieval schemes, and to provide a common testbed using both synthetic data and empirical imaging datasets [24]. PhasePack can also benchmark different algorithms against one another, and generate performance comparisons with varying numbers of measurements, signal-to-noise ratio, iterations, and running time. The package handles single method testing as well as multiple method comparisons.

The methods in PhasePack differ in numerous ways from their original descriptions in the literature in order to achieve improved robustness to different measurement models and to enable faster convergence. After reviewing the kinds of methods available in PhasePack in Section III, we will discuss practical implementation consideration in Section IV.

## III. METHODS AVAILABLE IN PHASEPACK

The methods implemented in PhasePack can be divided into three categories. First, alternating minimization methods work by iteratively estimating the missing phases of the measurements, and then solving the linear system  $\mathbf{Ax} = \tilde{\mathbf{b}}$  where  $\tilde{\mathbf{b}}$  is formed by applying phase estimates to  $\mathbf{b}$ . This class of algorithms includes the classical Gerchberg-Saxton [14] and Fienup [12] methods, in addition to more recent methods proposed in [9], [23], [28].

Second, many approaches attack the non-convex formulation directly using least-squares formulations [6], [7], [10], [10], [21], [32]–[35], [38]. These methods use gradient descent to minimize an objective of the form

$$f(\mathbf{x}) = \|\hat{\mathbf{A}}\mathbf{x}^p - \hat{\mathbf{b}}^p\|_2 \quad (1)$$

where  $\hat{\mathbf{A}}$  and  $\hat{\mathbf{b}}$  contain either a subset of the measurements or a re-weighting of the system  $\mathbf{Ax} = \mathbf{b}$ , and  $p \in \{1, 2\}$  is an integer exponent. Because of the non-convexity of the formulation (1), these methods require careful initialization to avoid local minimizers.

The third class of methods convert phase retrieval into a convex problem. This includes the “lifting methods” PhaseLift [8]

and PhaseCut [31], which square the dimensionality of the problem. PhasePack also implements several low-rank approximate solvers for large instances of PhaseLift, including gauge duality methods [1], [13] and sketching methods [37]. Finally, PhasePack contains the newer non-lifting relaxations PhaseMax [2], [15], [16] and PhaseLamp [11].

#### IV. BUILDING PRACTICAL IMPLEMENTATIONS

Recent phase retrieval methods have been developed with the goal of proving rigorous guarantees for random Gaussian measurement matrices. As a result, methods described in the literature are not optimized for performance, and may become unstable when used on non-Gaussian measurements. This is particularly an issue for variations of Wirtinger flow, which require the choice of stepsize parameters. This is illustrated in Figure 1, which compares a “strict” implementation of Wirtinger flow using the stepsize rules described in [6] to PhasePack’s implementation with adaptive stepsize.

The implementations in PhasePack depart from the literature in several ways that make them more robust and efficient. We discuss several such issues below.

##### A. Adaptive stepsizes and backtracking

Wirtinger Flow and its variants rely on gradient descent methods, and are sensitive to the choice of stepsize parameters. Existing stepsize rules presented in the literature are designed for Gaussian measurement models, and may be unstable or slow when used on generic matrices. PhasePack solves least-squares formulations of the phase retrieval problem using the general gradient descent solver FASTA [17], which supports adaptive stepsizes, automated stopping conditions, and conjugate-gradient acceleration.

This gradient solver automates optimization in several ways. For stepsize selection, PhasePack uses the Barzilai-Borwein adaptive method [3]. To guarantee stability, the method uses a backtracking line-search similar to the classical Armijo line search [5]. Classical line searches enforce that the objective decreases monotonically. For non-convex problems where the local curvature changes rapidly and local minima are prevalent, monotonic searches may result in excessive backtracking, and may be more prone to getting stuck in bad local minimizers. For this reason, several authors have proposed non-monotonic line searches [18], [19]. The adaptive strategy and line search rule used by PhasePack is described in detail in [17].

Finally, the solver in PhasePack is capable of using L-BFGS acceleration (which was studied for phase retrieval in [21]) and non-linear conjugate gradient methods to speed up convergence.

##### B. Practical methods for spectral initialization

Non-convex methods for phase retrieval are prone to getting stuck in local minimizers if they are not initialized properly. Phasepack includes two main classes of initializers: (i) the spectral initializer and its variants and, (ii) the orthogonality promoting initializer. The initializers described in the literature are optimized for random Gaussian data, and may not perform well on non-Gaussian real-world data. To be robust to different

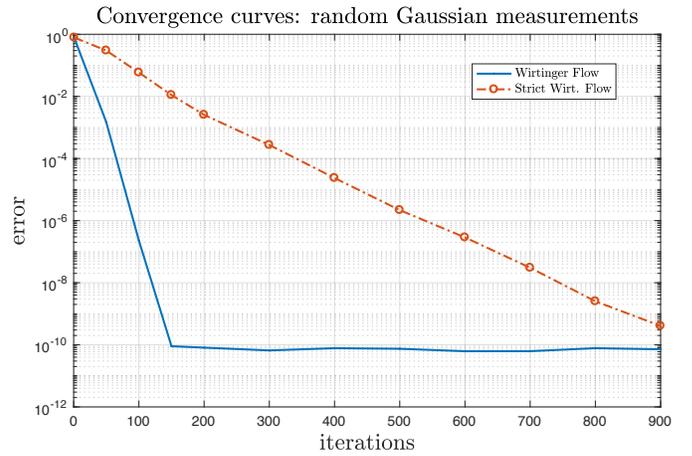


Fig. 1. Reconstruction of a signal using two variants of Wirtinger flow. (blue) Optimized Wirtinger flow with adaptive stepsize and backtracking line search. (red) A “strict” implementation of Wirtinger flow using the non-adaptive stepsize rule described in the original paper.

measurement models, PhasePack initializers depart from the literature in three main ways: we use a Krylov subspace-based eigensolver, we re-scale data before pre-processing, and we re-scale the initialization vector after computation. We discuss these important differences below after briefly describing how initialization methods work.

Spectral initialization methods begin by forming the matrix

$$\mathbf{Y} = \frac{1}{m} \sum_{i=1}^m T(b_i^2) \mathbf{a}_i \mathbf{a}_i^H \quad (2)$$

where  $\mathbf{a}_i$  is the  $i$ th row of  $\mathbf{A}$ , and  $T(\cdot)$  is a “pre-processing” function, which is simply the identity in the original spectral method [28]. These methods are motivated by the observation that if  $\mathbf{a}$  is a random Gaussian vector and  $b$  is its corresponding measurement, then we have

$$\mathbb{E}_{\mathbf{a}}[b^2 \mathbf{a} \mathbf{a}^H] = 2\mathbf{x} \mathbf{x}^H + \|\mathbf{x}\|^2 \mathbf{I}_n. \quad (3)$$

In fact, the leading eigenvector of (3) is the unknown signal  $\mathbf{x}$ . Spectral methods work by approximating the expectation (3) with the empirical matrix (2), computing the leading eigenvector, and then scaling it appropriately. A variety of methods have been proposed with different pre-processing functions, including the identity [6], [28], a “truncation” operator that is zero for small magnitude measurements [10], and a “re-weighting” operator that shrinks its argument [35]. A rigorous study of these methods appears in [22]. The default initialization in PhaseMax uses the “optimal” preprocessing function proposed in [27], which is

$$T(z) = \frac{z - 1}{z + \sqrt{\delta} - 1} \quad (4)$$

where  $\delta = m/n$  is the number of measurements divided by the signal dimension.

Orthogonality promoting initializers [33] work by identifying measurement vectors that are *not* correlated with the signal  $\mathbf{x}$ ,

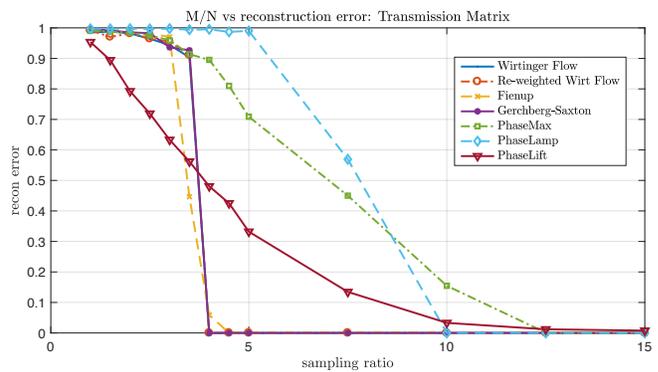
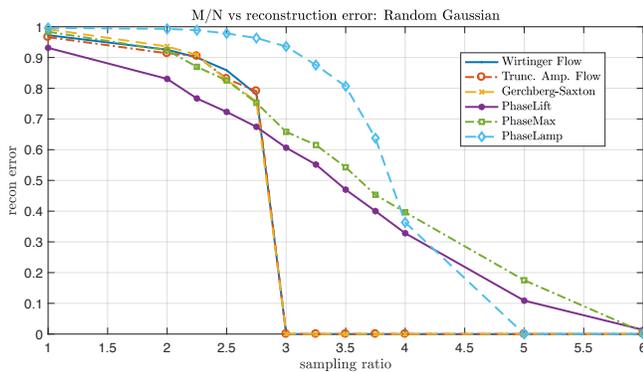


Fig. 2. Reconstruction error as a function of sampling ratio (number of measurements over number of unknowns) for a few algorithms. (left) Signal of length 100 using synthetic Gaussian measurements. (right) Signal of length 256 using an empirical transmission matrix measurement operator.

and finding an initialization vector that is orthogonal to those uncorrelated vectors. This is done by forming the matrix

$$\mathbf{Y} = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \frac{\mathbf{a}_i \mathbf{a}_i^H}{\|\mathbf{a}_i\|^2}$$

where  $\mathcal{I}$  contains the indices of measurement vectors that produce the smallest measurements. The initializer is then the eigenvector of  $\mathbf{Y}$  with *smallest* eigenvalue.

PhasePack initializers depart from the literature in several ways. We compute eigenvectors using the iterated Arnoldi method with restart rather than the power method. This method lacks the complexity guarantees of the power method, but exhibits faster convergence than the power method and can reliably resolve both the leading and trailing eigenvectors.

A more major departure from the literature is in how PhasePack scales data before pre-processing. The pre-processing functions  $T$  defined in the literature are optimized for random Gaussian measurement matrices and normalized signals, and initializers may be highly sub-optimal (or even ill-posed) when empirical data is used. For this reason, data is re-scaled to have the same statistics as the Gaussian model before pre-processing. Note that the measurement operator may be implemented as a black-box function, and so re-scaling needs to be done without access to the matrix entries or the underlying signal.

For example, the pre-processing operator (4) is optimized for Gaussian random measurements of variance  $1/n$  and a signal of length  $n$  [27], and has unpredictable results for any other model/signal. However, this particular measurement model produces measurements with  $\mathbb{E}_{\mathbf{a}} |\mathbf{a}^H \mathbf{x}|^2 = 1$ . For this reason, instead of applying the pre-processing operator to  $\mathbf{b}$ , we apply the pre-processing operator to the re-scaled measurement vector  $\frac{m\mathbf{b}}{\|\mathbf{b}\|}$ , which has the same mean as the Gaussian model.

Finally, PhasePack concludes initialization with a least-squares step to determine the optimal length of the initial vector. Most eigensolvers produce eigenvalues that have unit length by default, however, the length of the signal may effect the convergence of the phase retrieval solver. For this reason, after finding the initializer  $\hat{\mathbf{x}}$  we compute the least-squares solution to

$$\min_{\alpha} \|\alpha |\mathbf{A}\hat{\mathbf{x}}| - \mathbf{b}\|,$$

and then replace the initializer  $\hat{\mathbf{x}}$  with  $\alpha\hat{\mathbf{x}}$ . The spectral initialization methods described above come with their own scaling routines, however PhasePack’s least-squares scaling method is more practical for empirical data as it does not depend on the Gaussian measurement model.

## V. EMPIRICAL DATASETS

Public datasets for phase retrieval are scarce, and available datasets are often difficult to use because of lack of documentation or unusual data formats. One of PhasePack’s goals is to create a simple API for testing methods on empirical datasets. PhasePack provides routines for unpacking and pre-processing datasets, and prepares a simple measurement operator  $\mathbf{A}$  and measurement vector  $\mathbf{b}$  that can be used for phase retrieval. The datasets currently supported by PhasePack were obtained for the purpose of imaging an object through a diffusive medium, and are described in detail in [24]. We provide a brief description of this imaging modality below.

The imaging modality described [24] is for reconstructing an image from light that has passed through a diffusive medium (e.g., a light-scattering material like paper or eggshell). Imaging through such a material is impossible using a conventional camera, but this can be done using phase retrieval methods. A binary mask is created by placing a transparent LCD display in front of a light source. The pixels in this mask can be made either opaque or clear. An image pattern is loaded onto the mask, and the mask is illuminated from behind using a coherent (laser) light source. The light passes through the mask (creating an image), and then passes through a diffuser. The diffused light then lands on a photodetector that measures the intensity of light (but not the phase) at many locations, producing measurements.

The light pattern (image) passing through the mask is the unknown signal  $\mathbf{x}$ . The diffuser is modeled as a “transmission matrix”  $\mathbf{A}$  that describes how light passing through each pixel of the mask effects each of the detectors. Under this model, the signal received at each detector is given by  $\mathbf{A}\mathbf{x}$ . Because standard photodetectors measure intensity and not phase, the recorded signal is  $\mathbf{b} = |\mathbf{A}\mathbf{x}|$ , and the image must be reconstructed by phase retrieval.

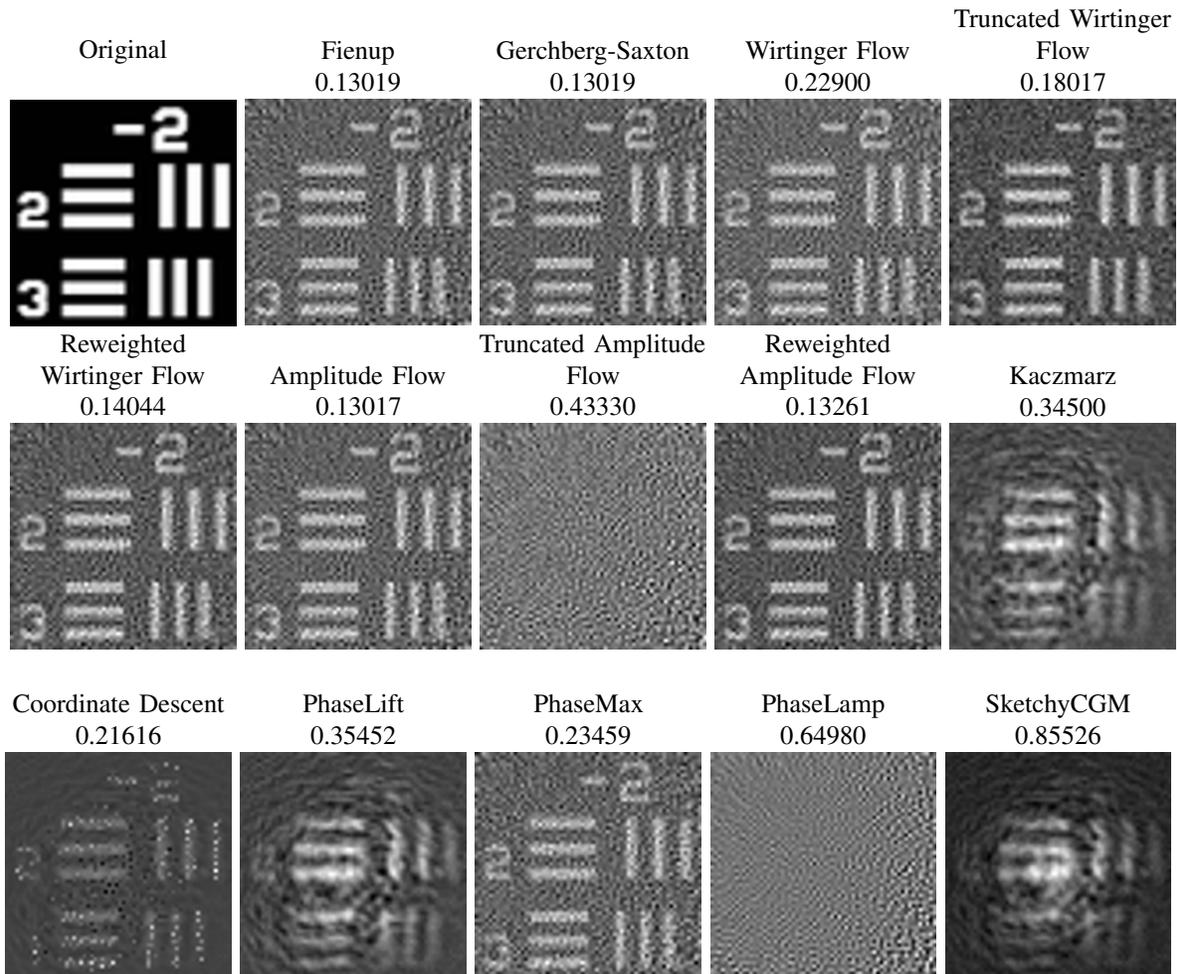


Fig. 3. Reconstruction of a  $64 \times 64$  pixel image (modified air force target) using empirical phaseless measurements. Below each algorithm name is the relative measurement error, defined as  $\| \|\mathbf{Ax} - \mathbf{b}\| / \|\mathbf{b}\|$ , achieved by each method.

This imaging modality is particularly useful for benchmarking phase retrieval methods because the ground-truth image (the pattern loaded onto the mask) is known, and can be used to evaluate the quality of reconstructions. A sample mask and its reconstructions is shown in Figure 3.

## VI. EXPERIMENTAL RESULTS

We demonstrate the capabilities of PhasePack using a range of experiments on real and synthetic data. All of these examples were produced using the scripts included in the `benchmark` sub-folder of the PhasePack distribution. Figure 2 shows the performance of several algorithms for recovering a random signal from random Gaussian measurements (left) and from measurements acquired using a real empirical transmission matrix (right). Figure 3 shows the reconstruction of an image from empirical measurements obtained using an optical device [24].

In most of our experiments, we found that the classical Fienup [12] and Gerchberg-Saxton [14] methods remain to be state-of-the-art. However, the re-weighted Amplitude flow

method [32] appears to be highly competitive, if not practically identical in performance.

Convex methods appear to under-perform their non-convex counterparts. The convex method PhaseMax [2], [15], [16] seems to outperform the lifted convex relaxation PhaseLift (and its approximate low-rank version SketchyCGM [36]) on real image data. However, PhaseLift outperforms PhaseMax by a small margin on random Gaussian measurements.

## VII. DISCUSSION

PhasePack was created with the goal of providing a unified framework in which researchers can investigate phase retrieval. By providing a common interface for different methods, and a simple API for testing methods on empirical datasets, we hope that PhasePack can help the community to better understand the strengths and weaknesses of different methods.

## ACKNOWLEDGMENTS

We would like to thank Ashok Veeraghavan and Chris Metzler at Rice University for making their empirical datasets available in PhasePack.

## REFERENCES

- [1] Alexandre Y Aravkin, James V Burke, Dmitriy Drusvyatskiy, Michael P Friedlander, and Kellie MacPhee. Foundations of gauge and perspective duality. *arXiv preprint arXiv:1702.08649*, 2017.
- [2] Sohail Bahmani and Justin Romberg. Phase retrieval meets statistical learning theory: A flexible convex relaxation. In *Intl. Conf. on Artificial Intelligence and Statistics (AISTATS)*, pages 252–260, May 2017.
- [3] Jonathan Barzilai and Jonathan M. Borwein. Two-point step size gradient methods. *IMA J Numer Anal*, 8(1):141–148, January 1988.
- [4] Liheng Bian, Jinli Suo, Guoan Zheng, Kaikai Guo, Feng Chen, and Qionghai Dai. Fourier ptychographic reconstruction using Wirtinger flow optimization. *Optics express*, 23(4):4856–4866, 2015.
- [5] R. Burachik, L. Mauricio, G. Drummond, A. Iusem, and E. Castorina. Full convergence of the steepest descent method with inexact line searches. *Optimization*, 32:137–146, 1995.
- [6] Emmanuel J Candès, Yonina C Eldar, Thomas Strohmer, and Vladislav Voroninski. Phase retrieval via matrix completion. *SIAM Rev.*, 57(2):225–251, Nov 2015.
- [7] Emmanuel J Candès, Xiaodong Li, and Mahdi Soltanolkotabi. Phase retrieval via Wirtinger flow: Theory and algorithms. *IEEE Trans. Inf. Theory*, 61(4):1985–2007, Feb. 2015.
- [8] Emmanuel J Candès, Thomas Strohmer, and Vladislav Voroninski. PhaseLift: Exact and stable signal recovery from magnitude measurements via convex programming. *Commun. Pure Appl. Math.*, 66(8):1241–1274, 2013.
- [9] Pengwen Chen, Albert Fannjiang, and Gi-Ren Liu. Phase retrieval with one or two diffraction patterns by alternating projection with null initialization. *arXiv preprint arXiv:1510.07379*, 2015.
- [10] Yuxin Chen and Emmanuel Candès. Solving random quadratic systems of equations is nearly as easy as solving linear systems. In *Adv. Neural Inf. Process. Syst.*, pages 739–747, 2015.
- [11] Oussama Dhifallah, Christos Thrampoulidis, and Yue M Lu. Phase retrieval via linear programming: Fundamental limits and algorithmic improvements. *arXiv preprint: 1710.05234*, 2017.
- [12] James R Fienup. Phase retrieval algorithms: a comparison. *Appl. Opt.*, 21(15):2758–2769, Aug. 1982.
- [13] Michael P Friedlander, Ives Macedo, and Ting Kei Pong. Gauge optimization and duality. *SIAM Journal on Optimization*, 24(4):1999–2022, 2014.
- [14] R. W. Gerchberg and W. O. Saxton. A practical algorithm for the determination of phase from image and diffraction plane pictures. *Optik*, 35:237–246, Aug. 1972.
- [15] T. Goldstein and C. Studer. PhaseMax: convex phase retrieval via basis pursuit. *arXiv preprint: 1610.07531*, Oct. 2016.
- [16] Tom Goldstein and Christoph Studer. Convex phase retrieval without lifting. *International Conference on Machine Learning*, 2017.
- [17] Tom Goldstein, Christoph Studer, and Richard Baraniuk. FASTA: A generalized implementation of forward-backward splitting, January 2015. ArXiv preprint, arXiv:1501.04979.
- [18] L. Grippo, F. Lampariello, and S. Lucidi. A Nonmonotone Line Search Technique for Newton’s Method. *SIAM Journal on Numerical Analysis*, 23(4):707–716, 1986.
- [19] Zhang. Hongchao and William Hager. A nonmonotone line search technique and its application to unconstrained optimization. *SIAM J. Optim.*, 14:1043–1056, 2004.
- [20] Kishore Jagannathan, Yonina C Eldar, and Babak Hassibi. Phase retrieval: An overview of recent developments. *arXiv:1510.07713*, Oct. 2015.
- [21] Ji Li and Tie Zhou. On gradient descent algorithm for generalized phase retrieval problem. *arXiv preprint arXiv:1607.01121*, 2016.
- [22] Yue M Lu and Gen Li. Phase transitions of spectral initialization for high-dimensional nonconvex estimation. *arXiv preprint arXiv:1702.06435*, 2017.
- [23] Stefano Marchesini, Yu-Chao Tu, and Hau-tieng Wu. Alternating projection, ptychographic imaging and phase synchronization. *Applied and Computational Harmonic Analysis*, 41(3):815–851, 2016.
- [24] Christopher A Metzler, Manoj K Sharma, Sudarshan Nagesh, Richard G Baraniuk, Oliver Cossairt, and Ashok Veeraraghavan. Coherent inverse scattering via transmission matrices: Efficient phase retrieval algorithms and a public dataset. In *Computational Photography (ICCP), 2017 IEEE International Conference on*, pages 1–16. IEEE, 2017.
- [25] Rick P Millane. Phase retrieval in crystallography and optics. *JOSA A*, 7(3):394–411, 1990.
- [26] DL Misell. A method for the solution of the phase problem in electron microscopy. *Journal of Physics D: Applied Physics*, 6(1):L6, 1973.
- [27] Marco Mondelli and Andrea Montanari. Fundamental limits of weak recovery with applications to phase retrieval. *arXiv preprint: 1708.05932*, 2017.
- [28] Praneeth Netrapalli, Prateek Jain, and Sujay Sanghavi. Phase retrieval using alternating minimization. In *Adv. Neural Inf. Process. Syst.*, pages 2796–2804, 2013.
- [29] Panos M Pardalos and Stephen A Vavasis. Quadratic programming with one negative eigenvalue is np-hard. *Journal of Global Optimization*, 1(1):15–22, 1991.
- [30] Yoav Shechtman, Yonina C Eldar, Oren Cohen, Henry Nicholas Chapman, Jianwei Miao, and Mordechai Segev. Phase retrieval with application to optical imaging: a contemporary overview. *IEEE Signal Processing Magazine*, 32(3):87–109, 2015.
- [31] Irène Waldspurger, Alexandre d’Aspremont, and Stéphane Mallat. Phase recovery, maxcut and complex semidefinite programming. *Math. Prog.*, 149(1-2):47–81, Feb. 2015.
- [32] G. Wang, G. B. Giannakis, Y. Saad, and J. Chen. Solving Almost all Systems of Random Quadratic Equations. *ArXiv e-prints*, 2017.
- [33] Gang Wang, Georgios B Giannakis, and Yonina C Eldar. Solving systems of random quadratic equations via truncated amplitude flow. *arXiv: 1605.08285*, Jul. 2016.
- [34] Ke Wei. Solving systems of phaseless equations via kacmarz methods: A proof of concept study. *Inverse Problems*, 31(12):125008, 2015.
- [35] Z Yuan and H Wang. Phase retrieval via reweighted Wirtinger flow. *Applied optics*, 56(9):2418, 2017.
- [36] A. Yurtsever, M. Udell, J. A. Tropp, and V. Cevher. Sketchy Decisions: Convex Low-Rank Matrix Optimization with Optimal Storage. *ArXiv e-prints*, February 2017.
- [37] Alp Yurtsever, Madeleine Udell, Joel Aaron Tropp, and Volkan Cevher. Sketchy decisions: Convex low-rank matrix optimization with optimal storage. In *Intl. Conf. on Artificial Intelligence and Statistics (AISTATS)*, May 2017.
- [38] Wen-Jun Zeng and HC So. Coordinate descent algorithms for phase retrieval. *arXiv preprint arXiv:1706.03474*, 2017.