

Optimization on Spheres: Models and Proximal Algorithms with Computational Performance Comparisons*

D. Russell Luke[†], Shoham Sabach[‡], and Marc Teboulle[§]

Abstract. We present a unified treatment of the abstract problem of finding the best approximation between a cone and spheres in the image of affine transformations. Prominent instances of this problem are phase retrieval and source localization. The common geometry binding these problems permits a generic application of algorithmic ideas and abstract convergence results for nonconvex optimization. We organize variational models for this problem into three different classes and derive the main algorithmic approaches within these classes (13 in all). We identify the central ideas underlying these methods and provide thorough numerical benchmarks comparing their performance on synthetic and laboratory data. The software and data of our experiments are all publicly accessible. We also introduce one new algorithm, a cyclic relaxed Douglas–Rachford algorithm, which outperforms all other algorithms by every measure: speed, stability, and accuracy. The analysis of this algorithm remains open.

Key words. phase retrieval, nonconvex optimization, nonsmooth optimization, proximal algorithms, feasibility, fixed points, source localization

AMS subject classifications. 90C26, 49M99, 65K10

DOI. 10.1137/18M1193025

1. Introduction. Nonconvex optimization is maturing into a major focus within continuous optimization. The numerical intuition that one brings to this area from convex optimization can be misleading. The purpose of this work is to provide thorough numerical benchmarks on a prevalent type of nonconvex problem which will establish guideposts for the good, the bad, and the ugly of numerical methods. The problem we consider is the following:

The cone and sphere problem:

Find a point nearest (in some sense) to a cone and to spheres in the image of affine transformations.

There are two prevalent examples of this problem that appear regularly in the literature: phase retrieval and source localization. There is no shortage of numerical schemes for addressing these problems, but most of these methods can be grouped into less than a handful

*Received by the editors June 8, 2018; accepted for publication (in revised form) May 28, 2019; published electronically August 13, 2019.

<https://doi.org/10.1137/18M1193025>

Funding: This work was supported in part by GIF grant G-1253-304.6/2014. The first author's research was partially supported by Deutsche Forschungsgemeinschaft grant SFB755. The third author's research was supported in part by the Israel Science Foundation, ISF grant 998/12.

[†]Institut für Numerische und Angewandte Mathematik, Universität Göttingen, Göttingen 37083, Germany (r.luke@math.uni-goettingen.de).

[‡]Faculty of Industrial Engineering and Management, Technion–Israel Institute of Technology, Haifa 3200003, Israel (ssabach@ie.technion.ac.il).

[§]School of Mathematical Sciences, Tel-Aviv University, Ramat-Aviv 69978, Israel (teboulle@post.tau.ac.il).

of distinctly different ideas. We compare these different classes of algorithms, 13 algorithms in all (not counting equivalent algorithms under different names), on two different source localization problems (with and without noise) and 15 phase retrieval problems, 12 synthetic and three involving laboratory data. Our results summarize more than 19,000 individual numerical experiments. A comparable benchmark study for phase retrieval was carried out by Marchesini [54] over a decade ago.

Phase retrieval has been rediscovered recently in the statistics and applied mathematics communities. Amongst the latest trends is to change the problem via randomization. This together with convex relaxations and reformulations to semidefinite programs has inspired a number of new algorithms. A collection of data and routines has been made available that compares many of the recent proposals for phase retrieval, including variants of the Wirtinger Flow and phase lift approaches [21]. The benchmarks we present here are complementary to these and expand the list of publicly available datasets. The data and software for our experiments are all publicly accessible [48] and will hopefully serve as a starting point for future numerical methods. One distinction between our benchmarks and the ones appearing in [21] is the scale of the problems: the largest datasets available in [21] are 64×64 . Having to work with datasets three orders of magnitude larger, as is the case for the X-ray diffraction experiments we use for benchmarks, make approaches surveyed in [21], like phase lift, simply not feasible. Still other algorithms, like Wirtinger Flow, are limited to the context of a certain kind of phase retrieval and do not generalize to the source localization benchmarks we present. We compare representative algorithms for phase retrieval featured in [21] formally and numerically to algorithms that are standard in the convex optimization literature, like Cyclic Projections, Douglas–Rachford, and the Alternating Directions Method of Multipliers. Our results confirm other studies that expose some unwarranted enthusiasm for recently proposed approaches for phase retrieval [56] and show that algorithms that are beloved in the convex setting are not the best performers for nonconvex problems.

In addition to the benchmarks, one major contribution of our work is a new algorithm, Cyclic Relaxed Douglas–Rachford (Algorithm 3.3), that appears to outperform every other algorithm by a remarkable margin in all respects. The other main contribution is a systematic categorization of algorithms from the perspective of variational analysis. Many of the algorithms we include in our benchmarks are standard approaches in convex optimization, but they have not received much attention in the most recent applied mathematics literature on phase retrieval. Our categorization provides for tremendous insight towards interpreting the numerical results and in seeing what features are mainly responsible for better (or worse) performance.

The understanding of first-order *prox*-based algorithms, like Douglas–Rachford, applied to nonconvex problems has matured to the point that quantitative local convergence guarantees are now possible with the tools and framework introduced in [52]. A global analysis is incomplete, but some progress has been made in [11]. The issue of convergence, however, is either misinterpreted or misunderstood in many contemporary numerical comparisons. We are interested only in fixed points of algorithms and the relation of these points to the problem one *thinks* these algorithms solve. Nonconvex minimization problems have, in general, multiple local minima and, more generally, multiple critical points. Most algorithms for solving optimization problems, on the other hand, are designed to find critical points. It is not uncommon,

however, to see fixed points of algorithms conflated with solutions to some optimization problem. Even worse, algorithms are compared according to some metric like least squares residual to some true solution, which in many instances has no relation to the variational problem upon which an algorithm is based, if such a basis exists. Even when an algorithm is derived from a particular variational problem, the fixed points of the algorithm need not coincide with critical points of the motivating optimization problem, much less solutions (see Cyclic Projections (Algorithm 3.1), Douglas–Rachford equation (3.3), and subsequent discussion). Our study of phase retrieval will hopefully remind readers of this distinction.

We present the algorithms through the lens of feasibility problems and their relaxations to smooth optimization. This helps to underscore the difference between critical points of an optimization problem and fixed points of an algorithm. Indeed, *no honest feasibility model of a phase retrieval problem has critical points, much less a solution*, yet the algorithms still converge to good fixed points that defy any obvious variational characterization. In fact, the algorithms built on feasibility models outperformed all other classes of algorithms by every relevant performance measure: *computational efficiency, speed of convergence, accuracy, and robustness against local minimums*.

2. Problem instances. We begin with a brief description of the two fundamental problem instances from which our numerical benchmarks are taken.

2.1. Phase retrieval.

The approximate physical model for phase retrieval is

$$(2.1) \quad \|(\mathcal{F}\mathcal{P}_j(z))_i\| = b_{ij} \quad \forall j = 1, 2, \dots, m, \forall i = 1, 2, \dots, n.$$

Here $\mathcal{F} : \mathbb{C}^n \rightarrow \mathbb{C}^n$ is a unitary linear operator accounting for the propagation of an electromagnetic wave, $\mathcal{P}_j : \mathbb{C}^n \rightarrow \mathbb{C}^n$ for $j = 1, 2, \dots, m$ is a linear operator accounting for m different settings of the instrument through which the wave travels, $z \in \mathbb{C}^n$ is the unknown object that interacts with the wave at one end (the *pupil or object plane*) of the instrument, and $b_{ij} \in \mathbb{R}_+$ is the i th pixel ($i = 1, 2, \dots, n$) of the j th measurement. The phase retrieval problem involves determining the phase of z (that is, the real and imaginary parts) from the amplitude of its image under $\mathcal{F}\mathcal{P}_j$. The problem is fundamental to diffraction imaging experiments. This includes near-field holography, far-field diffraction imaging, (classical) ptychography, and wavefront sensing, to name just a few observation techniques where this problem arises; see e.g., [47] and references therein. In all instances, the measurements are real and nonnegative numbers.

The different settings of the measurement device are accounted for in different operators \mathcal{P}_j —the mapping \mathcal{F} at the foundation of the imaging model remains unchanged. There are a number of different experimental settings that can be captured by the mapping \mathcal{P}_j ; we mention a few here. In a typical ptychographic experiment, an X-ray beam scans a specimen from side to side and an image is recorded at each scan position of the beam. The j th spatial translation of the beam is represented by \mathcal{P}_j . The difference here to the model discussed in [36], for instance, is that the beam is assumed to be fully known and shift invariant. Alternatively, as discussed in [49, 34, 70, 69] one could record the image at different positions along the axis of propagation of the field, again represented by different mappings \mathcal{P}_j which account for magnification and defocus factors involved in such spatial translations. More recently, there have been proposals for adding random phase masks at the object plane. The idea of phase

masks has its detractors [47], but we include this dataset as a point of reference to the more recent phase retrieval literature and it too fits the general model: the m different masks can be represented by \mathcal{P}_j for $j = 1, 2, \dots, m$.

In addition to these data equations, there are certain a priori qualitative constraints that can (and should) be added depending on the type of experiment that has been conducted. Often these are support constraints, or real-valuedness, or nonnegativity. In astronomy, for instance, the constraints are often support and magnitude constraints describing a field of constant magnitude across the aperture of the telescope. These types of constraints are described by the model (2.1), where \mathcal{F} is the identity mapping. In X-ray diffraction imaging, support or support and nonnegativity constraints are frequently imposed. In the early 00's, Oszlányi and Sütő [62, 63] proposed a simple *charge flipping* procedure that was quickly integrated into the software of the crystallography community for structure determination [73, 64, 24]. Marchesini [55] identified the charge flipping operation as a reflector of a hard thresholding operation in the first algorithm for *sparse phase retrieval*. Sparsity constraints were more recently applied by Loock and Plonka [42, 43] for phase retrieval of objects enjoying a sparse representation in a wavelet-type dictionary.

The solution to the phase retrieval problem as presented here is a complex-valued vector $z \in \mathbb{C}^n$ that satisfies (2.1) for all i and j in addition to the a priori information implicit in the experiment (support, nonnegativity, sparsity, etc.). Representing n -dimensional complex-valued vectors instead as 2-dimensional vectors on an n -dimensional product space, the phase retrieval problem is to find $z = (z_1, z_2, \dots, z_n) \in (\mathbb{R}^2)^n$ ($z_j \in \mathbb{R}^2$) satisfying (2.1) for all i and j in addition to qualitative constraints. We return below to the issue of existence of such a vector. The short answer is that, unless the data is the pattern created by a periodic object, there does not exist a solution to problem (2.1), much less a *unique* solution. But this is not a concern for us, either practically or theoretically. One of the main goals of our study here is to convince readers that there are mathematically sound approaches that do not involve unrealistic assumptions of existence and uniqueness of solutions to equations that do not possess solutions.

The models and methods discussed here are based on a *feasibility* approach to this problem; that is, at least initially, we will be happy with *any* point that comes as close as possible to matching the measurements. This, in short, is the way around existence and uniqueness. The points satisfying single measurements can be represented by sets. With this perspective, the question of existence is transformed into the question of whether the sets corresponding to different measurements have points in common. The question of uniqueness amounts to whether there is only a single common point between the sets.

2.2. Source localization. The source localization problem appears in a broad range of applications, including, for instance, mobile communication and wireless networks [23, 74], acoustic/sound source localization [41], GPS localization [7], and brain activity identification [29], to mention just a few. The problem is based on distance measurements from an array of sensors (also called anchors). Here one is given a collection of m sensors which are denoted by $a_j \in \mathbb{R}^d$, where $d = 2$ or 3 and $j = 1, 2, \dots, m$. Each a_j contains the exact location of the j th sensor. The data consists of distance measurements between the unknown source and the sensors; this data is represented by $b_j > 0$, $j = 1, 2, \dots, m$, and denotes the (possibly noisy)

measurement of the range between the source and the j th sensor a_j . This is described by the following equations:

$$(2.2) \quad b_j = \|\mathcal{P}_j(z)\|, \quad j = 1, 2, \dots, m,$$

where $\mathcal{P}_j(z) \equiv z - a_j$, $j = 1, 2, \dots, m$, is the linear shift mapping. The problem is then to find an adequate approximation of the unknown source \bar{z} satisfying the system (2.2). The only difference between source localization and the phase retrieval problems presented in the previous section is that source localization does not involve a Fourier-like transform \mathcal{F} (see model (2.1)). While this paper was under review, we became aware of another paper where the link between phase retrieval and source localization was observed [59].

2.3. Unifying representation. There are a number of different ways to represent the sets which we show are all equivalent with regard to the algorithms. We place all models in a real vector space, though in the context of phase retrieval it is understood that this is a reformulation of a complex-valued model space. The model mappings in the previous sections are then linear mappings $\mathcal{F} : (\mathbb{R}^d)^n \rightarrow (\mathbb{R}^d)^n$ and $\mathcal{P}_j : (\mathbb{R}^d)^n \rightarrow (\mathbb{R}^d)^n$ with $\mathcal{F}\mathcal{P}_j(z) = \hat{z} = (\hat{z}_1, \hat{z}_2, \dots, \hat{z}_n)$ for $\hat{z}_i \in \mathbb{R}^d$. The sets of possible vectors satisfying the measurements are given by

$$(2.3) \quad C_j \equiv \left\{ z \in (\mathbb{R}^d)^n \mid \|(\mathcal{F}\mathcal{P}_j(z))_i\| = b_{ij} \quad \forall i = 1, 2, \dots, n \right\}.$$

Alternatively, one can work with the sets

$$(2.4) \quad C'_j \equiv \left\{ z \in (\mathbb{R}^d)^n \mid \|(\mathcal{F}(z))_i\| = b_{ij} \quad \forall i = 1, 2, \dots, n \right\}$$

or

$$(2.5) \quad \hat{C}_j \equiv \left\{ z \in (\mathbb{R}^d)^n \mid \|z_i\| = b_{ij} \quad \forall i = 1, 2, \dots, n \right\}.$$

Note that, for all $j = 1, 2, \dots, m$, we have the following relations between these sets:

$$(2.6) \quad C_j = \mathcal{P}_j^* C'_j = \mathcal{P}_j^* \mathcal{F}^* \hat{C}_j,$$

where \mathcal{P}_j^* and \mathcal{F}^* are the adjoints of \mathcal{P}_j and \mathcal{F} , respectively.

For phase retrieval $d = 2$, n is large (1024^2 is not uncommon) and m is anywhere from 1 to 10. For sensor localization, $d = 2$ or 3 , $n = 1$, and m is at least 3 but not usually greater than 100. These sets are in different spaces relative to one another, but for both applications \mathcal{F} and \mathcal{P}_j , $j = 1, 2, \dots, m$, are often unitary, and so the choice of space to work in is a matter of convenience only. Even though the sets C_j are nonconvex (a line segment between any two points in the sets does not belong to the sets), they have very nice structure. The sets \hat{C}_j , for instance, are just ℓ_2 -spheres in an n -dimensional product space of (\mathbb{R}^d) with componentwise radii given by the elements of the vector b . This is true regardless of whether or not the measurement b is contaminated with noise. As such, these sets are smooth, *semialgebraic* (constructible by finite systems of polynomial inequalities), and *prox-regular* (loosely defined as sets with locally single-valued projections [68]). By the relationship (2.6), the sets C_j and

C'_j also enjoy this nice structure. These facts were already observed in [36, Proposition 3.5] and [46].

We reserve the set C_0 for the qualitative constraints. The qualitative constraints that most often apply are either of the same form as the sets above—as in the case of support and magnitude constraints—or cones. A support constraint alone is a restriction of points to a subspace. Support and nonnegativity is the positive orthant of a subspace, that is, a convex cone. It is easy to see that sets of points satisfying sparsity constraints can be characterized as unions of subspaces (see, for instance, [5]), and hence the set of points satisfying a sparsity constraint is also a cone.

3. Variational models. A key feature that allows one to classify algorithms is the degree of smoothness in the underlying model. We group the algorithms into two classes below in increasing order of model smoothness. A third class of algorithms mixes smooth and nonsmooth formulations and involves a *product space formulation* that gathers constraints/sets into blocks at a modest cost in increased dimensionality. Nonsmooth aspects are maintained within the blocks, but communication between blocks is modeled, most often, by smooth operators. The numerical results presented in section 4 indicate that the smoother the model is, the slower the algorithm progresses toward a fixed point. Smoothness is also sometimes associated with stability or reliability. Here again, our results do not support this intuition for this family of nonconvex problems: the most reliable and robust algorithms are also in the nonsmooth and feasibility-based category. We do not take into account advantages of parallelization and other architecture-dependent features that could make a difference in clock times.

3.1. Model Category I: Multiset feasibility. The most natural place to start is by naively trying to find a point in the intersection of the data generated sets C_j , $j = 1, 2, \dots, m$, given by (2.3) together with the possible qualitative constraint set C_0 :

$$\text{Find } z^* \in \bigcap_{j=0}^m C_j.$$

We show in section 4 that this leads to the most effective methods for solving the problem. Keeping in mind, however, that for all practical purposes the intersection above is empty, we also show that the algorithm is not solving the problem we thought it was solving.

For our purposes we will prefer to pose this problem equivalently in an optimization format

$$(3.1) \quad \min_{z \in (\mathbb{R}^d)^n} \sum_{j=0}^m \iota_{C_j}(z),$$

where

$$\iota_{C_j}(z) \equiv \begin{cases} 0 & \text{if } z \in C_j, \\ +\infty & \text{else.} \end{cases}$$

The indicator function ι_{C_j} , $j = 0, 1, \dots, m$, is an *extended real-valued* function from $(\mathbb{R}^d)^n$ to

the extended real-line $\mathbb{R} \cup \{+\infty\}$. The fact that the intersection is empty is reflected in the fact that the optimal value to problem (3.1) is $+\infty$.

Despite these worrisome issues, we examine the *method of Cyclic Projections*.

Algorithm 3.1 Cyclic Projections - CP.

Initialization. Choose $z^0 \in (\mathbb{R}^d)^n$.

General Step ($k = 0, 1, \dots$)

$$z^{k+1} \in P_{C_0} P_{C_1} \cdots P_{C_m} z^k.$$

Here the *orthogonal projector* P_{C_j} for $j = 0, 1, \dots, m$ is defined by

$$P_{C_j}(z) \equiv \operatorname{argmin}_{y \in C_j} \|y - z\|.$$

Since C_j is nonconvex, the projector is, in general, a set-valued mapping. This can be computed explicitly [49, Corollary 4.3], for all $j = 1, 2, \dots, m$, by the formula

$$(3.2) \quad y \in P_{C_j}(z) \iff y = \mathcal{P}_j^* \mathcal{F}^* \hat{y}, \quad \hat{y}_i \in \begin{cases} b_{ij} \frac{(\mathcal{F}\mathcal{P}_j(z))_i}{\|(\mathcal{F}\mathcal{P}_j(z))_i\|} & \text{if } (\mathcal{F}\mathcal{P}_j(z))_i \neq 0, \\ b_{ij} \mathbb{S} & \text{if } (\mathcal{F}\mathcal{P}_j(z))_i = 0. \end{cases}$$

The unit sphere in \mathbb{R}^d is denoted by \mathbb{S} above. The projector P_{C_0} is also explicitly known and has a structure that is no more complicated than (3.2); often it is simpler.

An early champion of feasibility models was Censor, who together with Cegielski has written an excellent review on the extensive literature on numerical methods for these problems [19]. For a recent review of results on algorithms for inconsistent feasibility, see [20]. The analysis of Cyclic Projections for consistent and inconsistent nonconvex problems has been established in [52]. For consistent feasibility, the only critical points are points of intersection, i.e., globally optimal solutions. For inconsistent feasibility, there do not exist critical points and the optimal value of the problem is ∞ ; that is, the problem is *infeasible*. For nonconvex feasibility, both consistent and inconsistent, the fixed points of the algorithm consist of points of intersection, when they exist, as well as points that are not points of intersection. Under mild regularity assumptions of the sets, the fixed points correspond to cycles of smallest length locally over all other possible cycles generated by projecting onto the sets in the same order. Strong guarantees on convergence of the iterates, like a local linear rate, have been established generically for problems with this structure (see [52, Example 3.6]). What remains is to establish guarantees for global convergence to fixed points as well as assurances that the fixed points correspond to cycles of globally minimal length.

Another Cyclic Projection-type method for inconsistent feasibility problems is based on what is commonly known as the *Douglas–Rachford (DR)* algorithm (the original algorithm is a domain splitting method for partial differential equations [25]). The method can only be applied directly to two-set feasibility problems:

$$\text{Find } x \in C_0 \cap C_1.$$

The fixed point iteration is given by

$$(3.3) \quad (DR) \quad z^{k+1} \in \frac{1}{2} (R_{C_0} R_{C_1} + \text{Id}) z^k,$$

where $R_C \equiv 2P_C - \text{Id}$ is the *reflector* operator of the set C and Id denotes the identity operator. It is important not to forget that, even if the feasibility problem is consistent, the fixed points of the Douglas–Rachford algorithm will not in general be points of intersection. Instead, the *shadows* of the iterates defined as $P_{C_1} z^k$, $k \in \mathbb{N}$, converge to intersection points when they exist [4].

To extend this to more than two sets, Borwein and Tam [13, 14] proposed the following variant.

Algorithm 3.2 Cyclic Douglas–Rachford - CDR.

Initialization. Choose $z^0 \in (\mathbb{R}^d)^n$.

General Step ($k = 0, 1, \dots$)

$$z^{k+1} \in \left(\frac{1}{2} (R_{C_0} R_{C_1} + \text{Id}) \right) \left(\frac{1}{2} (R_{C_1} R_{C_2} + \text{Id}) \right) \cdots \left(\frac{1}{2} (R_{C_m} R_{C_0} + \text{Id}) \right) z^k.$$

Borwein and Tam showed that, in an infinite-dimensional Hilbert space setting, the Cyclic Douglas–Rachford algorithm applied to collections of convex sets converges weakly to a fixed point, when it exists [14], regardless of whether the intersection of sets is empty.

It is easy to envision different sequencing strategies than the one presented above. In [6], one of the pair of sets is held fixed, and this has some theoretical advantages in the convex setting. We did not observe any advantage for the problems studied here. A comprehensive investigation of optimal sequencing strategies for problems with different structures has not been published.

A relaxation to the Douglas–Rachford algorithm first proposed in [44] is described below in Algorithm 3.8. At this stage, we just motivate it as a convex combination of the Douglas–Rachford mapping and the projection onto the “inner” set C_1 above: for $\lambda \in (0, 1]$,

$$(3.4) \quad (DR\lambda) \quad z^{k+1} \in \left(\frac{\lambda}{2} (R_{C_0} R_{C_1} + \text{Id}) + (1 - \lambda) P_{C_1} \right) z^k.$$

Extending this to more than two sets yields Algorithm 3.3 below.

In an infinite-dimensional Hilbert space setting, Luke, Martins, and Tam [50] have shown that the iterates of the $CDR\lambda$ algorithm converge weakly to the intersection of *convex sets*. The analysis of this algorithm for the cone and sphere problem remains open.

In the convex setting, the Douglas–Rachford algorithm can be derived from the Alternating Directions Method of Multipliers (ADMM) [32] for solving a dual problem [30]. This algorithm is extremely popular at the moment for large-scale convex problems with linear constraints, and the literature in this context is massive (see, for instance, [72] and references therein). In the nonconvex setting, the dual correspondence is lost, though there have been some recent

Algorithm 3.3 Cyclic Relaxed Douglas–Rachford - CDR λ .

Initialization. Choose $z^0 \in (\mathbb{R}^d)^n$ and $\lambda \in [0, 1]$.

General Step ($k = 0, 1, \dots$)

$$\begin{aligned} z^{k+1} &\in \left(\frac{\lambda}{2} (R_{C_0} R_{C_1} + \text{Id}) + (1 - \lambda) P_{C_1} \right) \left(\frac{\lambda}{2} (R_{C_1} R_{C_2} + \text{Id}) + (1 - \lambda) P_{C_2} \right) \\ &\quad \cdots \left(\frac{\lambda}{2} (R_{C_m} R_{C_0} + \text{Id}) + (1 - \lambda) P_{C_0} \right) z^k. \end{aligned}$$

developments and studies [39, 65, 11]. The ADMM falls into the category of augmented Lagrangian-based methods. Thus, we can reformulate problem (3.1) as

$$(3.5) \quad \min_{x, z_j \in (\mathbb{R}^d)^n} \left\{ \iota_{C_0}(x) + \sum_{j=1}^m \iota_{C_j}(z_j) \mid z_j = x, j = 1, 2, \dots, m \right\},$$

and then one can apply ADMM to the augmented Lagrangian of this formulation; specifically, we have

$$(3.6) \quad \tilde{L}_\eta(x, z_j, v_j) \equiv \iota_{C_0}(x) + \sum_{j=1}^m \left(\iota_{C_j}(z_j) + \langle v_j, x - z_j \rangle + \frac{\eta}{2} \|x - z_j\|^2 \right),$$

where $\eta > 0$ is a penalization parameter and $v_j, j = 1, 2, \dots, m$, are the multipliers associated with the linear constraints. Therefore, ADMM applied to finding critical points of the corresponding augmented Lagrangian (see (3.6)) is given by Algorithm 3.4 below.

An ADMM scheme for phase retrieval has appeared in [40] and for sensor localization in [51]. Our numerical experiments below do not indicate any advantage of this approach. We include it, however, as a point of reference to both the Douglas–Rachford algorithm and the smoother AvP² algorithm discussed below in Algorithm 3.13. Seeing what changes render a bad algorithm reasonable provides tremendous insight. Note that the projections in step 2 of Algorithm 3.4 can be computed in parallel, while the Cyclic Projections and the Cyclic Douglas–Rachford algorithms must be executed sequentially. The benchmarking comparisons carried out in section 4 do not reflect the advantages of parallelizable methods like ADMM₁ when implemented on multiple CPU/GPU architectures.

Before moving on to the two other categories, there is one other idea that is worth mentioning. While the sets $C_j, j = 1, 2, \dots, m$, are nonconvex, one could in principle *convexify* the indicator function ι_{C_j} by computing its *Fenchel conjugate*, defined as

$$\sigma_{C_j}(y) \equiv \sup_x \{ \langle y, x \rangle - \iota_{C_j}(x) \}.$$

The function σ_{C_j} is the *support function* of the set C_j [71] and is a *convex* function. It is easily shown that $\sigma_{C_j}(y) = \sigma_{\text{co}(C_j)}(y)$, where $\text{co}(C_j)$ is the convex hull of C_j . Similarly, the support function of $C \equiv \cap_j C_j$ in the direction y is just $\sigma_C(y)$ and this is equivalent to $\sigma_{\text{co}(C)}(y)$.

Algorithm 3.4 Nonsmooth ADMM₁.

Initialization. Choose $x^0, z_j^0, v_j^0 \in (\mathbb{R}^d)^n$ and fix $\eta > 0$.

General Step ($k = 0, 1, \dots$)

1. Update

$$(3.7) \quad \begin{aligned} x^{k+1} &\in \operatorname{argmin}_{x \in (\mathbb{R}^d)^n} \left\{ \iota_{C_0}(x) + \sum_{j=1}^m \left(\langle v_j^k, x - z_j^k \rangle + \frac{\eta}{2} \|x - z_j^k\|^2 \right) \right\} \\ &= P_{C_0} \left(\frac{1}{m} \sum_{j=1}^m \left(z_j^k - \frac{1}{\eta} v_j^k \right) \right). \end{aligned}$$

2. For all $j = 1, 2, \dots, m$, update (in parallel)

$$(3.8) \quad \begin{aligned} z_j^{k+1} &\in \operatorname{argmin}_{z_j \in (\mathbb{R}^d)^n} \left\{ \iota_{C_j}(z_j) + \langle v_j^k, x^{k+1} - z_j \rangle + \frac{\eta}{2} \|x^{k+1} - z_j\|^2 \right\} \\ &= P_{C_j} \left(x^{k+1} - \eta v_j^k \right). \end{aligned}$$

3. For all $j = 1, 2, \dots, m$, update (in parallel)

$$(3.9) \quad v_j^{k+1} = v_j^k + \eta \left(x^{k+1} - z_j^{k+1} \right).$$

The problem here is that the set C is, for most practical purposes, empty. The intersection $\mathcal{C} \equiv \bigcap_j \text{co}(C_j)$, however, is never empty for the phase retrieval problem (it always contains the origin), and so instead one could consider computing $\sigma_C(y)$ and then trying to *maximize* this with respect to y , in order to come as close as possible to the boundaries of all the sets. The numerical procedure one arrives at via this strategy was proposed independently in [2, 33], though the connection to Fenchel conjugation has not been recognized and the argument y of the support function is kept fixed. This numerical approach is presented as a convex relaxation, which it would be were it not for the fact that the desired phase is found at the point where $\sigma_C(y)$ attains its *maximum value* on the unit ball. Instead of maximizing $\sigma_C(y)$, the approach involves a mixture of statistics and luck to guess an appropriate fixed \hat{y} . Interested readers are referred to [56], where numerical comparisons of this method against some of the methods considered here were carried out.

3.2. Model Category II: Product space formulations. The second category of algorithms occupies the middle ground between the category of nonsmooth and feasibility models above and the third category of smooth optimization approaches. Here the basic idea is to lift the problem to the product space $((\mathbb{R}^d)^n)^{m+1}$, which can then be formulated as a two-set feasibility problem:

$$\text{Find } \mathbf{z}^* \in C \cap D,$$

where $\mathbf{z}^* = (z_0^*, z_1^*, \dots, z_m^*)$, $C := C_0 \times C_1 \times \dots \times C_m$, and D is the diagonal set of $(\mathbb{R}^d)^{n(m+1)}$ which is defined by $\{\mathbf{z} = (z, z, \dots, z) : z \in (\mathbb{R}^d)^n\}$. Two important features of this formulation are the following: (i) the projection onto the set C can be easily computed since

$$P_C(\mathbf{z}) = (P_{C_0}(z_0), P_{C_1}(z_1), \dots, P_{C_m}(z_m)),$$

where P_{C_j} , $j = 1, 2, \dots, m$, are given in (3.2), and (ii) D is a subspace which also has a simple projection given by $P_D(\mathbf{z}) = \bar{\mathbf{z}}$, where

$$\bar{z}_j = \frac{1}{m+1} \sum_{j=0}^m z_j.$$

The most natural algorithm for this formulation is the one we began with, namely Cyclic Projections (Algorithm 3.1). In the case of just two sets, this is known as the *Alternating Projections* algorithm.

Algorithm 3.5 Alternating Projections - AP.

Initialization. Choose $\mathbf{z}^0 \in (\mathbb{R}^d)^{n(m+1)}$.

General Step ($k = 0, 1, \dots$)

$$\mathbf{z}^{k+1} \in P_D P_C \mathbf{z}^k.$$

One can easily verify that Algorithm 3.5 is equivalent to the following iteration:

$$z_j^{k+1} \in \frac{1}{m+1} \sum_{j=0}^m P_{C_j}(z_j^k), \quad j = 0, 1, \dots, m;$$

in other words, the Alternating Projections algorithm on the product space coincides with Averaged Projections (Algorithm 3.11) and Alternating Minimization (Algorithm 3.10). It is also interesting to note that, in the product space, the Alternating Projections algorithm is equivalent to the very popular *Projected Gradient* method. To see this, consider the following minimization problem:

$$(3.10) \quad \underset{\mathbf{z} \in C}{\text{minimize}} \quad \frac{1}{2} \text{dist}^2(\mathbf{z}, D).$$

It should be noted that a point in the intersection of C and D (if one exists) corresponds to the case that the optimal value of problem (3.10) is zero.

The objective of this minimization problem is convex and continuously differentiable (since D is a closed and convex set) with a Lipschitz continuous gradient (with constant 1) given by $\nabla \text{dist}^2(\mathbf{z}, D) = 2(\mathbf{z} - P_D \mathbf{z})$ (see (3.15)). The classical Projected Gradient algorithm applied to this problem follows immediately.

Algorithm 3.6 Projected Gradient - PG.

Initialization. Choose $\mathbf{z}^0 \in (\mathbb{R}^d)^{n(m+1)}$.

General Step ($k = 0, 1, \dots$)

$$\begin{aligned} \mathbf{z}^{k+1} \in P_C \left(\mathbf{z}^k - \frac{1}{2} \nabla \text{dist}_D^2(\mathbf{z}^k) \right) &\iff \mathbf{z}^{k+1} \in P_C P_D \mathbf{z}^k \\ &\iff u^{k+1} \in \frac{1}{m+1} \sum_{j=0}^m P_{C_j} u^k, \text{ where } z_j^{k+1} = P_{C_j} u^{k+1}. \end{aligned}$$

It is well known that the Projected Gradient algorithm can be accelerated in the convex setting [58, 8]. Although there is no theory that supports acceleration in the nonconvex setting, the recent work [66] demonstrates successful empirical results for a class of phase retrieval problems. Following along this line, a Fast Projected Gradient algorithm for problem (3.10) reads as follows.

Algorithm 3.7 Fast Projected Gradient - FPG.

Initialization. Choose $\mathbf{z}^0, \mathbf{y}^1 \in (\mathbb{R}^d)^{n(m+1)}$ and $\alpha_k = \frac{k-1}{k+2}$ for all $k \in \mathbb{N}$.

General Step ($k = 0, 1, \dots$)

$$\begin{aligned} \mathbf{z}^k &\in P_C \left(\mathbf{y}^k - \frac{1}{2} \nabla \text{dist}^2(\mathbf{y}^k, D) \right), \\ \mathbf{y}^{k+1} &= \mathbf{z}^k + \alpha_k (\mathbf{z}^k - \mathbf{z}^{k-1}) \\ &\iff \\ \mathbf{z}^k &\in P_C P_D \mathbf{y}^k, \\ \mathbf{y}^{k+1} &= \mathbf{z}^k + \alpha_k (\mathbf{z}^k - \mathbf{z}^{k-1}). \end{aligned}$$

There is no theory for the choice of acceleration parameter α_k , $k \in \mathbb{N}$, in Algorithm 3.7 for nonconvex problems, but our numerical experiments indicate that an investigation into this would be fruitful.

Another approach to accelerate the Projected Gradient algorithm is by adding an inertial term (see [61, 67] for two recent studies of this idea in the nonconvex setting, including references therein).

Another algorithmic approach that can fit to the setting of two-set feasibility is the now popular Douglas–Rachford algorithm [25]. To compensate for the absence of fixed points for inconsistent feasibility, Luke proposed in [44] a relaxation of this algorithm, which can be viewed as the usual Douglas–Rachford algorithm for more general proximal mappings applied to a relaxation of the feasibility problem to a smooth constrained optimization problem

$$(3.11) \quad \underset{\mathbf{z} \in (\mathbb{R}^d)^{n(m+1)}}{\text{minimize}} \quad \left\{ \frac{\lambda}{2(1-\lambda)} \text{dist}^2(\mathbf{z}, D) + \iota_C(\mathbf{z}) \right\}.$$

Algorithm 3.8 Relaxed Douglas–Rachford - DR λ .

Initialization. Choose $\mathbf{z}^0 \in (\mathbb{R}^d)^{n(m+1)}$ and $\lambda \in [0, 1]$.

General Step ($k = 0, 1, \dots$)

$$(3.12) \quad \mathbf{z}^{k+1} \in \frac{\lambda}{2} \left(R_D R_C \mathbf{z}^k + \mathbf{z}^k \right) + (1 - \lambda) P_C \mathbf{z}^k.$$

In [44, 45], this algorithm is called the *Relaxed Averaged Alternating Reflections* (RAAR) algorithm. It was shown in [45] that this fixed point mapping is precisely the proximal Douglas–Rachford algorithm applied to the problem (3.11), that is,

$$\frac{1}{2} (R_1 R_C + \text{Id}) = \frac{\lambda}{2} (R_D R_C + \text{Id}) + (1 - \lambda) P_C,$$

where R_1 is the *proximal reflector* of the function $f_\lambda(\mathbf{z}) \equiv \frac{\lambda}{2(1-\lambda)} \text{dist}^2(\mathbf{z}, D)$, that is, $R_1(\mathbf{z}) = 2 \text{prox}_{1,f_\lambda}(\mathbf{z}) - \mathbf{z}$. The analysis of [52] also applies to the product space formulation of phase retrieval and source localization.

A similar algorithm called Hybrid Projection Reflections was proposed in [3], but we do not include this in our comparisons because this algorithm, like the original HIO algorithm [28] that inspired it, is not stable.

An interesting alternative to the Douglas–Rachford and Alternating Projections algorithms is the following algorithm, which in a limiting case is simply a convex combination of the two algorithms. For an analysis of this algorithm and a characterization of the set of fixed points, see [75].

Algorithm 3.9 Douglas–Rachford-Alternating Projections - DRAP.

Initialization. Choose $\mathbf{z}^0 \in (\mathbb{R}^d)^{n(m+1)}$ and $\lambda \in [0, 1]$.

General Step ($k = 0, 1, \dots$)

$$(3.13) \quad \mathbf{z}^{k+1} \in P_D \left((1 + \lambda) P_C \mathbf{z}^k - \lambda \mathbf{z}^k \right) - \lambda \left(P_C \mathbf{z}^k - \mathbf{z}^k \right).$$

3.3. Model Category III: Smooth nonconvex optimization. First, consider the problem of minimizing the sum of squared distances to the sets C_j , $j = 0, 1, \dots, m$, that is,

$$(3.14) \quad \underset{z \in (\mathbb{R}^d)^n}{\text{minimize}} \quad f(z) \equiv \frac{1}{2(m+1)} \sum_{j=0}^m \text{dist}^2(z, C_j).$$

Since the sets C_j , $j = 0, 1, \dots, m$, are nonconvex, the functions $\text{dist}^2(z, C_j)$ are clearly not differentiable, and hence the same is true for the objective function $f(z)$. However, in our context, the sets C_j , $j = 0, 1, \dots, m$, are prox-regular (cf. section 2.3). From elementary properties of prox-regular sets [68] it can be shown that the gradient of the squared distance is defined and differentiable with a Lipschitz continuous derivative (that is, the corresponding

Hessian) up to the boundary of C_j , $j = 0, 1, \dots, m$, and points where the coordinate elements of the vector z vanish. Indeed, for f given by (3.14) we have

$$(3.15) \quad \nabla f(z) \equiv \frac{1}{m+1} \sum_{j=0}^m (\text{Id} - P_{C_j})(z).$$

Thus, applying the Gradient Descent method with *unit step size* to problem (3.14), one immediately recovers Averaged Projections (Algorithm 3.11).

Readers familiar with variational analysis will also recognize (3.14) as the relaxation of (3.1) via the *Moreau envelope* [57] of the indicator functions ι_{C_j} , $j = 0, 1, \dots, m$. But even without these sophisticated tools, inspection shows that the objective in (3.14) is smooth, has full domain, and takes the value zero at points of intersection, if such points exist. These kinds of models are much more prevalent in applications than the more severe-looking feasibility format of problem (3.1).

There is another interesting way to tackle problem (3.14), allowing one to make links with another fundamental algorithmic paradigm. Ignoring the weighting factor for the moment, we consider the following problem:

$$(3.16) \quad \min_{z \in (\mathbb{R}^d)^n} f(z) \equiv \frac{1}{2} \sum_{j=0}^m \text{dist}^2(z, C_j).$$

Using the definition of the function $\text{dist}^2(\cdot, C_j)$, $j = 0, 1, \dots, m$, we can reformulate problem (3.16) as follows:

$$(3.17) \quad \min_{z, \mathbf{u}} \left\{ \sum_{j=0}^m \frac{1}{2} \|z - u_j\|^2 \mid u_j \in C_j, \quad j = 0, 1, \dots, m \right\},$$

where $\mathbf{u} = (u_0, u_1, \dots, u_m) \in ((\mathbb{R}^d)^n)^{m+1}$.

Note that problem (3.17) always has an optimal solution (since we minimize a continuous function over a closed and bounded set). The structure of the optimization problem (3.17), which includes constraint sets that are separable over the variables u_j , $j = 0, 1, \dots, m$, suggests that one can exploit this generous property when developing an optimization algorithm. Alternating Minimization (AM) is a classical optimization technique which was designed exactly for these situations and involves updating each variable separately in a cyclic manner. More precisely, AM when applied to problem (3.17) generates sequences defined by Algorithm 3.10.

By combining (3.18) and (3.19), the AM algorithm yields exactly *Averaged Projections* (Algorithm 3.11), which is recorded below.

This algorithm, which could be motivated purely from the feasibility framework detailed above, is often preferred not only because it is parallelizable but also because it appears to be more robust to problem inconsistency. The Averaged Projections algorithm can be equivalently viewed as the Projected Gradient algorithm when applied to problem (3.10).

Algorithm 3.10 Alternating Minimization - AM.

Initialization. Choose $(z^0, u_0^0, u_1^0, \dots, u_m^0) \in ((\mathbb{R}^d)^n)^{m+2}$.

General Step ($k = 0, 1, \dots$)

1. Update

$$(3.18) \quad z^{k+1} = \operatorname{argmin}_{z \in (\mathbb{R}^d)^n} \sum_{j=0}^m \frac{1}{2} \|z - u_j^k\|^2 = \frac{1}{m+1} \sum_{j=0}^m u_j^k.$$

2. For all $j = 0, 1, \dots, m$, update (in parallel)

$$(3.19) \quad u_j^{k+1} \in \operatorname{argmin}_{u_j \in C_j} \frac{1}{2} \|u_j - z^{k+1}\|^2 = P_{C_j}(z^{k+1}).$$

Algorithm 3.11 Averaged Projections - AvP.

Initialization. Choose $z^0 \in (\mathbb{R}^d)^n$.

General Step ($k = 0, 1, \dots$)

$$z^{k+1} \in \frac{1}{m+1} \sum_{j=0}^m P_{C_j} z^k.$$

Indeed, the problem (3.10) and (3.17) are equivalent, as we show now:

$$\begin{aligned} \min_{\mathbf{z} \in C} \frac{1}{2} \operatorname{dist}^2(\mathbf{z}, D) &= \min_{\mathbf{z} \in C} \min_{\mathbf{u} \in D} \frac{1}{2} \|\mathbf{z} - \mathbf{u}\|^2 = \min_{\mathbf{z} \in C} \min_{u \in (\mathbb{R}^d)^n} \frac{1}{2} \sum_{j=0}^m \|u - z_j\|^2 \\ &= \min_{u, \mathbf{z}} \left\{ \sum_{j=0}^m \frac{1}{2} \|u - z_j\|^2 \mid z_j \in C_j, \quad j = 0, 1, \dots, m \right\}. \end{aligned}$$

In the case that $m = 1$, i.e., only one image is considered, the AM algorithm discussed above coincides with what was called the *Error Reduction* algorithm in [27]. In [36, Remark 2.2(i)], the more general PHeBIE algorithm applied to the problem of blind ptychography was shown to reduce to the Averaged Projections algorithm for phase retrieval when the illuminating field is known. The PHeBIE algorithm is a slight extension of the PALM algorithm [10]. A partially preconditioned version of PALM was studied in [22] for phase retrieval, with improved performance over PALM.

The analysis of the Averaged Projections algorithm for problems with this geometry is covered by the analysis of nonlinear/nonconvex gradient descent. Much of this is classical and can be found throughout the literature, but it is limited to guarantees of convergence to *critical points* (see, for instance, [1, 10]). This begs the question as to which of the critical points are global minima. The answer to this is unknown.

Instead of always taking the fixed average $1/(m+1)$, in the formulation of problem (3.14),

it is possible to derive a variational interpretation of dynamically weighted averages between the projections to the sets C_j , $j = 0, 1, \dots, m$. This idea was proposed in [49], where it is called *extended least squares*. A similar approach was also proposed in [9], where the resulting algorithm is called the Sequential Weighted Least Squares (SWLS) algorithm. The underlying model in [49] is the negative log-likelihood measure of the sum of squared set distances:

$$(3.20) \quad \underset{z \in (\mathbb{R}^d)^n}{\text{minimize}} \sum_{j=0}^m \ln (\text{dist}^2(z, C_j) + c) \quad (c > 0).$$

Gradient descent applied to this objective yields the following *Dynamically Reweighted Averaged Projections* algorithm.

Algorithm 3.12 Dynamically Reweighted Averaged Projections - DyRePr.

Initialization. Choose $z^0 \in (\mathbb{R}^d)^n$ and $c > 0$.

General Step ($k = 0, 1, \dots$)

$$(3.21) \quad z^{k+1} \in z^k - \sum_{j=0}^m \frac{2}{(\text{dist}^2(z^k, C_j) + c)} (z^k - P_{C_j}(z^k)).$$

Algorithm 3.13 Two-Step Averaged Projections Algorithm - AvP².

Initialization. Choose $x^0 \in \mathbb{R}^n$ and $\rho_j > 0$, $j = 0, 1, \dots, m$. Compute $u_j^1 \in P_{C_j}(z^0)$ ($j = 0, 1, \dots, m$) and $z^1 \equiv (1/(m+1)) \sum_{j=0}^m u_j^1$.

General Step ($k = 0, 1, \dots$)

- Compute

$$(3.22) \quad z^{k+1} = \frac{1}{m} \sum_{j=1}^m \left(u_j^k + \frac{1}{\rho_j} (z^k - z^{k-1}) \right).$$

- For each $j = 1, 2, \dots, m$, compute

$$(3.23) \quad u_j^{k+1} = P_{C_j} \left(u_j^k + \frac{1}{\rho_j} (2z^k - z^{k-1}) \right).$$

Least squares-based models. In [53], Marchesini studies an augmented Lagrangian approach to solving

$$(3.24) \quad \underset{z \in (\mathbb{R}^d)^n}{\text{minimize}} \frac{1}{2n} \sum_{j=0}^m \sum_{i=1}^n (\|(\mathcal{F}\mathcal{P}_j(z))_i\| - b_{ij})^2.$$

It is not difficult to see that this is a nonsmooth least squares relaxation of problem (2.1) [66, Lemma 3.1]. In reduced form, the resulting ADMM algorithm has the following prescription,

which corrects an error in [51].¹ The reduced form is recognizable as Averaged Projections with a two-step recursion; we therefore refer to this as AvP² (Algorithm 3.13), which stands for *Two-Step Averaged Projections* algorithm.

This algorithm also can be viewed as a smoothed/relaxed version of Algorithm 3.4.

The appearance of the norm in (3.24) makes the analysis of the least squares approach inconvenient without the tools of nonsmooth analysis. A popular way around this is to formulate (2.1) as a system of quadratic equations:

$$(3.25) \quad \|(\mathcal{F}\mathcal{P}_j(z))_i\|^2 = b_{ij}^2 \quad \forall j = 1, 2, \dots, m, \quad \forall i = 1, 2, \dots, n.$$

The corresponding squared least squares residual of the quadratic model is exceedingly smooth:

$$(3.26) \quad \min_{z \in (\mathbb{R}^d)^n} G(z) \equiv \frac{1}{2} \sum_{j=0}^m \sum_{i=1}^n \left(\|(\mathcal{F}\mathcal{P}_j(z))_i\|^2 - b_{ij}^2 \right)^2.$$

A popular approach in the applied mathematics and statistics communities for solving the squared least squares formulation (3.26) is based on a trick from conic programming for turning quadratics into linear functions in a lifted space of much higher dimension. The lifted linear objective is still constrained in rank, which is a nonconvex constraint, but this is typically relaxed to a convex constraint. The idea, called *phase lift* when applied to phase retrieval, is not an efficient approach for a number of reasons, not the least of which being the increase in dimension (the square of the dimensionality of the original problem). Indeed, the phase lift method is not implementable on standard consumer-grade architectures for almost all of the benchmarking experiments conducted below.

Similarly to phase lift, the recent paper [18] has inspired many studies of the Wirtinger Flow (WF) algorithm for solving problem (3.26). The WF method is a Gradient Descent algorithm applied in the image space of the mapping \mathcal{F} to minimize the function G . This leads to Algorithm 3.14 below.

Algorithm 3.14 Wirtinger Flow - WF.

Initialization. Choose $z^0 \in (\mathbb{R}^d)^n$ and step size $\mu > 0$.

General Step ($k = 0, 1, \dots$)

$$(3.27) \quad z^{k+1} = z^k - \frac{\mu}{\|z^0\|^2} \left(\left\| \left(\mathcal{F}\mathcal{P}_j(z^k) \right)_i \right\|^2 - b_{ij}^2 \right) z^k.$$

While smoothness makes the analysis nicer, the quartic objective has almost no curvature around critical points, which makes convergence of first-order methods much slower than first-order methods applied to nonsmooth objectives. See [49, section 5.2] for a discussion of this. The numerical comparisons here support this conclusion. More recently, algorithms for minimizing (3.24) (amplitude-based) were compared with those for minimizing (3.26)

¹The simplified form of [51, Algorithm 1 (simplified)] has an error in equation (5.2).

(intensity-based) from the practitioner's perspective [77]. Here also the less smooth variational models perform better from the standpoint of image quality in the presence of noise and model misspecification.

Beyond first-order methods. The twice continuous differentiability of the sum of squared distances at most points suggests that one could try higher-order techniques from nonlinear optimization in order to accelerate the basic Gradient Descent method. Higher-order accelerations, like quasi-Newton methods, rely on extra smoothness in the objective function. However, as observed in [38], a quasi-Newton-based method can be used to solve nonsmooth problems. In the numerical comparisons in section 4, we benchmark a limited memory BFGS method (Algorithm 3.15) applied in [49] against the other techniques. Recently, studies have appeared comparing second-order and nonlinear Gauss–Newton methods for ptychography with Wirtinger Flow (Algorithm 3.14) and Cyclic Projection-type algorithms [77].

This rather complicated-looking algorithm is a standard in nonlinear optimization, and it even shows unexpected (and largely unexplained) good performance for nonsmooth problems [38]. Convergence is still open, but we include this algorithm as one of the typical types of accelerations via higher-order methods one might try.

A summary of the algorithms above together with the theoretical state of the art is presented in Table 1.

Table 1

State of the theory for the algorithms in this study. A + indicates that the theory is well developed for settings that cover the cone and sphere problem; a 0 indicates that the theory is developed for certain settings—convex, for instance—but that the setting of the cone and sphere problem remains open. A – indicates that the theory remains unexplored.

	Literature	Global		Local	
		crit./fixed pts	rates	crit./fixed pts	rates
Model Category I					
CP	[52]	+	0	+	+
CDR	[13, 14]	0	–	0	–
CDR λ	[50]	0	–	0	–
ADMM ₁	[11]	+	0	+	0
Model Category II					
FPG	[66]	–	–	–	–
DR	[35, 52]	0	–	+	+
DR λ	[52]	0	–	+	+
DRAP	[75]	0	–	+	+
Model Category III					
Wirtinger	[18]	–	–	+	–
AP/AvP/PG	[1, 49, 37, 46, 52]	+	+	+	+
AvP ²	[51]	+	–	+	–
DyRePr	[49, 9]	–	–	0	–
QNAvP	[49]	–	–	+	0

4. Numerical comparisons. The benchmarking study we present here is intended to compare numerical performance between the categories of algorithms delineated above. We do not attempt to show which algorithm is best within each category, but stark differences between algorithms based on different variational models (amplitude-based versus intensity-based, for

Algorithm 3.15 Limited Memory BFGS with Trust Region - QNAvP

Initialization. Choose $z^0 \in (\mathbb{R}^d)^n$ and $c > 0$.

General Step ($k = 0, 1, \dots$)

[1.]

1. (Initialization) Choose $\tilde{\eta} > 0$, $\zeta > 0$, $\bar{\ell} \in \{1, 2, \dots, n\}$, $z^0 \in \mathbb{C}^n$, and set $\nu = \ell = 0$. Compute $\nabla f(z^0)$ and $\|\nabla f(z^0)\|$ for

$$f(z) \equiv \frac{1}{2(m+1)} \sum_{j=0}^m \text{dist}^2(z, C_j), \quad \nabla f(z) \equiv \frac{1}{m+1} \sum_{j=0}^m (\text{Id} - P_{C_j})(z).$$

2. (L-BFGS step) For each $k = 0, 1, 2, \dots$, if $\ell = 0$, compute z^{k+1} by some line search algorithm; otherwise compute

$$s^k = -\left(M^k\right)^{-1} \nabla f(z^k),$$

where M^k is the L-BFGS update [17], $z^{k+1} = z^k + s^k$, $f(z^{k+1})$, and the *predicted change* (see, for instance, [60]).

3. (Trust region) If $\rho(s^k) < \tilde{\eta}$, where

$$\rho(s^k) = \frac{\text{actual change at step } k}{\text{predicted change at step } k},$$

reduce the trust region Δ^k , solve the trust region subproblem for a new step s^k [16], and return to the beginning of step 2. If $\rho(s^k) \geq \tilde{\eta}$, compute $z^{k+1} = z^k + s^k$ and $f(z^{k+1})$.

4. (Update) Compute $\nabla f(z^{k+1})$, $\|\nabla f(z^{k+1})\|$,

$$y^k \equiv \nabla f(z^{k+1}) - \nabla f(z^k), \quad s^k \equiv z^{k+1} - z^k,$$

and $s^{kT} y^k$. If $s^{kT} y^k \leq \zeta$, discard the vector pair $\{s^{k-\ell}, y^{k-\ell}\}$ from storage and set $\ell = \max\{\ell - 1, 0\}$, $\Delta^{k+1} = \infty$, $\mu^{k+1} = \mu^k$, and $M^{k+1} = M^k$ (i.e., shrink the memory and do not update); otherwise set $\mu^{k+1} = \frac{y^{kT} y^k}{s^{kT} y^k}$ and $\Delta^{k+1} = \infty$, add the vector pair $\{s^k, y^k\}$ to storage, if $\ell = \bar{\ell}$, and discard the vector pair $\{s^{k-\ell}, y^{k-\ell}\}$ from storage. Update the Hessian approximation M^{k+1} [17]. Set $\ell = \min\{\ell + 1, \bar{\ell}\}$, $\nu = \nu + 1$ and return to step 2.

instance) are demonstrated. We therefore do not include all possible algorithms in the comparisons. The algorithms that simply are not competitive, like *phase lift* or *HPR*, are not included either because they simply cannot be implemented on the problem (as for phase lift) or because it is noncompetitive (as for both phase lift and HPR). Other algorithms that are frequently used in the applications literature are equivalent to one of the algorithms above (HIO [28] and the *Difference Map* method [26] are either Douglas–Rachford or HPR

depending on the qualitative constraints; Gerchberg–Saxton [31] and Error Reduction [28] are the Alternating and Averaged Projections algorithms). Refinements to basic algorithmic templates (preconditioning, step-length optimization, etc.) can lead to speedups, but these improvements are often marginal, and so we do not compare implementations of optimized representatives from the categories. Where algorithms are sensitive to parameter selection, we make no claim to having found the optimal parameter values. The sensitivity of an algorithm to parameter values can also be taken as an indicator of reliability and robustness, but a thorough investigation of this aspect of the algorithms in our comparisons is beyond the scope of our study. We have, however, made every reasonable effort to find parameter values that put each algorithm in the best light possible.

In all of the tabulated numerical experiments below, the algorithms are randomly initialized and performance statistics are compiled for 100 trials. This also includes the tables where a “warm start” is benchmarked—here the warm start procedure is randomly initialized. In each row of the tables, the performance statistics of the algorithms are reported. The column “failure” reports the number of times in 100 trials that the algorithm failed to achieve a specified distance to the true solution. Our tolerance for “failure” for experiments without noise is set to the reasonable expectation for numerical accuracy given the problem size. For noisy experiments, the tolerated distance to the true solution is based on the variance of the noise. The other columns report median, maximum, and minimum iteration counts needed to achieve the mathematically justified stopping criteria, namely the iterate step size, over 100 trials. The step size of the iterates is the only mathematically relevant indicator of convergence to a critical point, and this can be monitored regardless of whether the algorithm iterates can be interpreted as a minimizing sequence of some objective function. For the interpretation of the critical point (whether a local minimizer, global minimizer point, or otherwise), see the references in Table 1. The column “failure” serves as an empirical test as to whether the critical point to which an algorithm appears to be converging is reasonably close to the true solution, regardless of what the theory tells us (usually much more conservative than observed in practice). Iteration counts are justified as a reasonable comparison metric since all of the algorithms in this comparison have similar per-iteration computational complexity. The exception to this is the quasi-Newton accelerated Averaged Projections (QNAvP) algorithm. This algorithm has a per-iteration cost of approximately five times that of the other algorithms.

In numerical tests with experimental data, only a single random initialization was chosen and the resulting phase reconstructions are presented. For all but one of the experimental datasets, there is no ground truth, and so the termination of the algorithms is based on the best available mathematical convergence theory, namely convergence of the iterate steps. Evaluation of the quality of the critical points of the algorithms for experimental data is beyond the scope of this study.

4.1. Wavefront sensing: JWST dataset. We begin with numerical comparisons on the James Webb Space Telescope wavefront reconstruction dataset presented in [49] (at that time the telescope was known only as the Next Generation Space Telescope). This is synthetic data, but it was carefully constructed to simulate the actual telescope. For our numerical experiments, we use a resolution of 128×128 for each of the images. The data for this

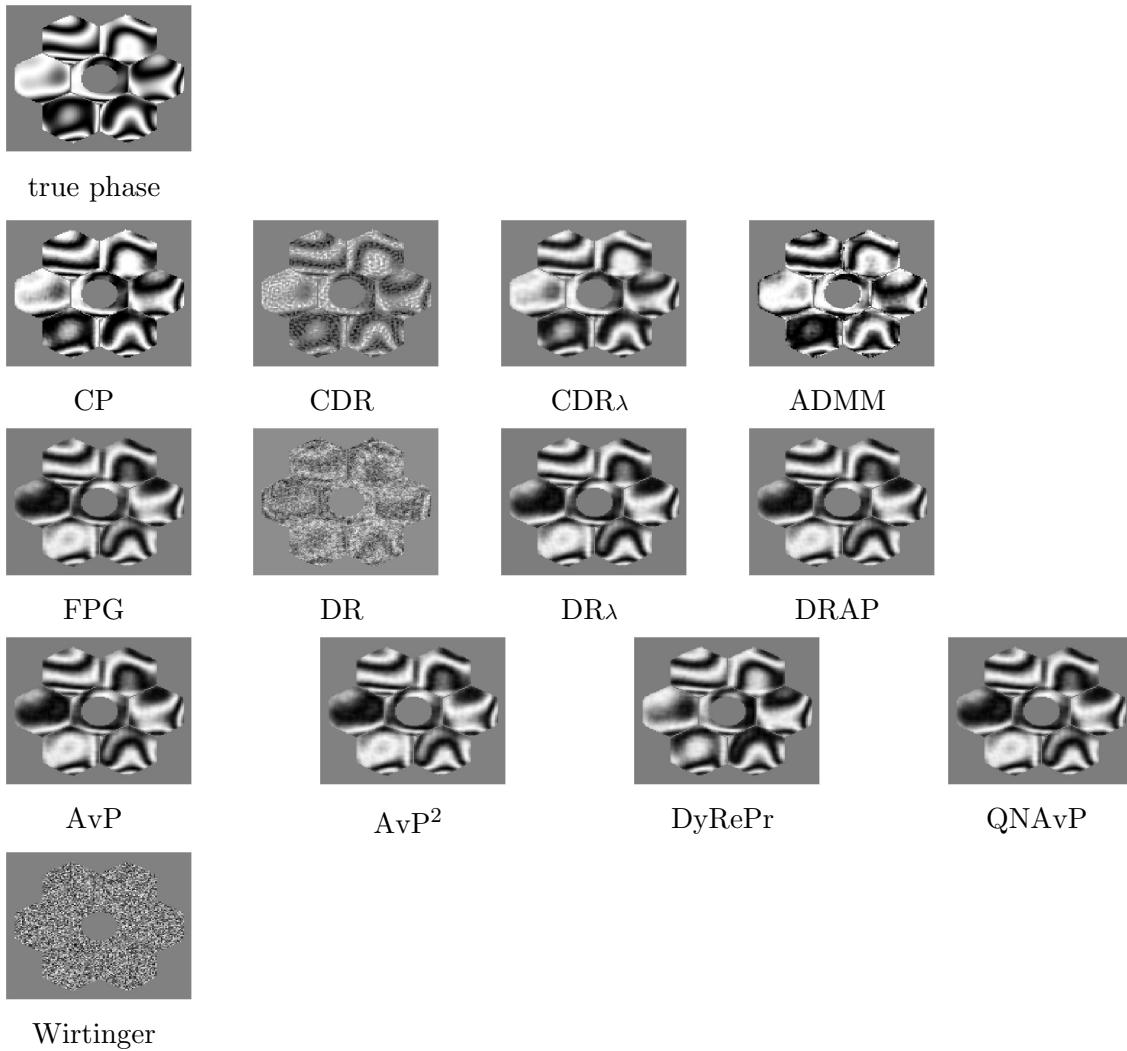


Figure 1. Representative phase reconstructions from algorithms applied to the noisy JWST dataset. The frame “true phase” was reprinted from [49].

problem, $b_{ij} \in (\mathbb{R}^2)$ in problem (2.1) with $n = 128^2$, consists of two out-of-focus images and one in-focus image of a known star ($m = 3$) together with the constraint that the wavefront to be constructed has unit amplitude across the aperture of the telescope (see [49]). The focus settings and aperture support are accounted for in the mappings \mathcal{P}_j in problem (2.1).

All algorithms start from the same randomly chosen initial points and are terminated once the difference between successive iterates falls below 5×10^{-5} or a maximum iteration count (6,000) is exceeded. The algorithm is judged to have failed if it does not achieve a distance to the true solution of 10^{-2} relative to global phase shift for noiseless data, or a distance of 0.35 for noisy data.

In Figure 1, we show typical reconstructions of each of the algorithms. It is clear from this

figure that the WF, Douglas–Rachford (DR), and Cyclic Relaxed Douglas–Rachford (CDR λ) algorithms fail with noisy data. But the figure does not reveal *how* they fail. The behavior of the iterations is contained in Table 2.

Table 2

JWST *wavefront reconstruction problem, noiseless and noisy, 100 random initializations.* The asterisks on CP and CDR λ are to note that these algorithms with noisy JWST data do not converge to fixed points but rather to a fixed amplitude with a constant global phase shift from one iterate to the next. The * for DyRePr indicates that the algorithm's fixed point still looks good, even if it does not lie within the specified distance to the true solution.

JWST	No noise				Noise			
	iteration count				iteration count			
	failure	median	high	low	failure	median	high	low
Model Category I								
CP*	0	27	201	17	0	21	48	14
CDR	0	9	18	8	100	*	*	*
CDR λ^* ($\lambda = 0.25$)	0	25	423	15	0	21	81	14
ADMM ₁ ($\eta = 3$)	0	70.5	115	47	100	*	*	*
Model Category II								
FPG	0	191	789	87	0	145.5	589	80
DR	68	704.5	821	631	100	*	*	*
DRA λ ($\lambda = 0.85/0.55$)	0	64.5	145	53	0	109	515	62
DRAP ($\lambda = 0.55/0.25$)	0	72.5	319	45	0	97	362	57
Model Category III								
Wirtinger	100	86	87	85	100	86	87	85
AP/AvP/PG	0	247	1700	95	0	138	508	77
AvP ² ($\rho_j = .75$)	1	383.5	6000	152	0	300	895	138
DyRePr	100*	221	2039	121	100*	151.5	586	88
QNAvP	3	95	240	47	3	67.5	206	34

- **Model Category I.** The benchmarked algorithms in this category are Cyclic Projections (CP - Algorithm 3.1), Cyclic Douglas–Rachford (CDR - Algorithm 3.2), Relaxed Cyclic Douglas–Rachford (CDR λ - Algorithm 3.3), and ADMM₁ (Algorithm 3.4). The CDR and ADMM₁ algorithms do not converge in the noisy case. A focused study of these two algorithms for nonconvex feasibility has not been carried out, but the connection of these algorithms to the Douglas–Rachford algorithm makes this behavior not surprising. The CP and CDR λ algorithms are marked with an asterisk to note that these algorithms with noisy JWST data *do not converge* to fixed points but rather to a fixed amplitude with a constant global phase shift from one iterate to the next. This is a surprising discovery that does not contradict the known theory for these algorithms but rather indicates the existence of limit cycles for this problem. The limit cycles, however, are simply fixed global phase shifts from the true solution, and so they do not detract from the algorithms' ability to deliver the true solution, up to global phase shift. A geometrical understanding of this phenomenon remains open.
- **Model Category II.** Due to the equivalence of the Averaged Projections algorithm to Alternating Projections (AP - Algorithm 3.5) and Projected Gradients (PG - Algorithm 3.6) on the product space, the performance of these algorithms is the same

as that for AvP (Algorithm 3.11). In addition to these, we tested Fast Projected Gradient (FPG - Algorithm 3.7), the product space implementation of Douglas–Rachford (DR - Algorithm 3.3), Relaxed Douglas–Rachford (DR λ - Algorithm 3.8), and the DRAP (Algorithm 3.9) variant. The only algorithm in this class that does not achieve the required tolerance to the true solution is the DR algorithm, which does not even converge in the noisy experiments, as predicted by the theory for this algorithm. All other algorithms converged reliably with little variance. The DRAP algorithm performed best overall for these choices of parameters, but none of the algorithms was particularly sensitive to the choice of parameters.

- **Model Category III.** The benchmarked algorithms belonging to this category are Averaged Projections (AvP - Algorithm 3.11), Dynamically Reweighted Averaged Projections (DyRePr - Algorithm 3.12), the ADMM-motivated Two-Step Averaged Projections recursion (AvP² - Algorithm 3.13), Wirtinger Flow (WF - Algorithm 3.14), and Limited Memory Quasi-Newton acceleration of Averaged Projections (QNAvP - Algorithm 3.15). The results for this class vary, depending primarily on the algorithm. Averaged Projections (Algorithm 3.11), Dynamically Reweighted Averaged Projections (Algorithm 3.12), and Limited Memory BFGS (Algorithm 3.15) were already compared in [49].

The WF algorithm was the worst of all algorithms in any class, and the AvP² algorithm was quite sensitive to the choice of parameter. The other algorithms (AvP, DyRePr, QNAvP) are for all intents and purposes parameter-free, and these performed about the same, regardless of the experiment. The WF algorithm failed to attain the required distance to the true solution in both the noiseless and noisy JWST experiments from each of the randomly generated initializations (see Table 2). It did, however, converge to a critical point in every experiment after, on average, 86 iterations with very little variance in the convergence. The WF algorithm does depend on parameter choices, but we found no parameters for which the algorithm could find a reasonable neighborhood of the solution for the JWST dataset. The other algorithms from this category performed much better than WF, though the interpretation of the fixed points of the DyRePr algorithm is not clear. The Averaged Projections algorithm found fixed points within the required tolerance of the true solution with and without noise every time, AvP² failed only once (without noise), and QNAvP failed 3% of the time. The median iteration counts for these algorithms ranged from 67 (QNAvP) to 384 (AvP²) with considerable variation at the upper end of iteration counts. The Dynamically Reweighted Projections (DyRePr) algorithm performed similarly to AvP² in terms of median iteration counts to the stopping criteria and variance, but the fixed points were never within the required distance of the true solution. Nevertheless, as seen in Figure 1 the fixed points look no different from the best of the reconstructions. The ambiguity of the failure of the DyRePr algorithm is marked with a “100*” in the “failure” column of Table 2.

4.2. Coded diffraction: CDP dataset. A phase problem for demonstrating the WF algorithm was presented in [18]. The experiments are of synthetic 1- and 2-dimensional signals. We compare the algorithms surveyed here on this problem instance for the same data made

available in [18]. The results of this comparison are summarized in Tables 3 and 4. There are several features of this problem that have attracted attention in the applied mathematics and statistics communities in recent years. Regarding the physical experiment that this dataset mimics, it is imagined that one has a *phase mask* at the object plane that one can randomly generate. The data consists of 10 observations of a true signal, each observation $b_{ij} \in \mathbb{R}^2$ in problem (2.1) with $n = 128^2$ and $m = 10$ made with a different, randomly generated, phase mask— \mathcal{P}_j , $j = 1, 2, \dots, m$, in problem (2.1). There is no qualitative constraint set C_0 . To avoid getting stuck in a bad local minimum, the WF approach involves a *warm start* procedure which is a power series iteration (50 in our experiments) that is meant to land one in the neighborhood of a global minimum.

Table 3

1-dimensional phase retrieval, problem CDP [18], cold/warm start, 100 random initializations. The “*” in the results for Douglas–Rachford indicate that this algorithm did not converge, despite remaining in an acceptable neighborhood of the true solution.

CDP	1-d cold iteration count				1-d warm iteration count			
	failure	median	high	low	failure	median	high	low
Model Category I								
CP	0	16	20	13	0	13	15	12
CDR	0	39	6000	20	2	38.5	6000	18
CDR λ ($\lambda = 0.33$)	0	6	6	5	0	6	6	5
Model Category II								
FPG	0	86	102	78	0	84	98	76
DR	0	*	*	131	0	*	*	136
DRL ($\lambda = 0.75$)	0	97	105	92	0	91	97	87
DRAP ($\lambda = 0.55$)	0	68.5	86	63	0	60	67	58
Model Category III								
Wirtinger	15	322	6000	290	9	279	6000	269
AP/AvP/PG	0	131.5	217	107	0	99	121	89
AvP ² ($\rho_j = .75$)	0	63	68	65	0	69	69	60
DyRePr	0	54	60	49	0	53	59	47
QNAP	0	25	28	21	0	24	28	20

This experiment also includes a 1-dimensional phase retrieval benchmark. It is held that 1-dimensional phase problems of this type are very different from 2-dimensional instances. This is a reasonable claim due to the early theoretical work showing that the 1-dimensional phase retrieval problem suffers from nonuniqueness, while 2-dimensional phase retrieval problems are almost always uniquely solvable when they are solvable at all [15]. The distinction, however, is spurious when the phase problem is *overdetermined*, as it is here. The numerical tests bear this out: there does not appear to be any qualitative computational difference between 1- and 2-dimensional phase retrieval problems when both are overdetermined. As with the JWST dataset, the stopping criterion was on the step size, and this was more stringent for the 1-dimensional example than the 2-dimensional example simply because of greater numerical accuracy.

All algorithms started from the same randomly chosen initial points and were terminated

Table 4

2-dimensional phase retrieval, problem CDP [18], cold/warm start, 100 random initializations. The “*” for DyRePr indicates that the algorithm’s fixed point still yields a point close to the true solution, though it does not lie within the specified distance for this comparison. The “*” in the results for Douglas–Rachford and Cyclic Douglas–Rachford indicate that these algorithms did not converge.

CDP	2-d cold iteration count				2-d warm iteration count			
	failure	median	high	low	failure	median	high	low
Model Category I								
CP	0	19	26	17	0	15	19	12
CDR	100	*	*	*	100	*	*	*
CDR λ ($\lambda = 0.33$)	0	11	14	11	0	9	12	8
Model Category II								
FPG	0	3178	7596	714	0	1375.5	5866	156
DR	100	*	*	*	100	*	*	*
DR λ ($\lambda = 0.75$)	0	88	97	85	0	79	87	73
DRAP ($\lambda = 0.55$)	0	57	93	68	0	58	72	49
Model Category III								
Wirtinger	0	412	999	332	12	266.5	404	6
AP/AvP/PG	0	229	334	179	0	122	227	80
AvP 2 ($\rho_j = .75$)	0	502	1061	373	0	191.5	572	119
DyRePr	100*	202.5	366	151	100*	91.5	390	47
QNAvP	3	247	616	150	12	52.5	230	28

once the difference between successive iterates fell below $1e-9$ (1-d) or $1e-7$ (2-d), respectively, or a maximum iteration count (6000) was exceeded.

A summary of the benchmarks is given below.

- **Model Category I.** The benchmarked algorithms in this category are Cyclic Projections (CP - Algorithm 3.1), Cyclic Douglas–Rachford (CDR - Algorithm 3.2), and Relaxed Cyclic Douglas–Rachford (CDR λ - Algorithm 3.3). The ADMM₁ algorithm did not converge and did not come close enough to the true solution for these experiments (much like Douglas–Rachford), and so it was not included in the tabulated results. Cyclic Douglas–Rachford does not converge for 2-dimensional CDP, but it does converge for 1-dimensional CDP. The CP and CDR λ are clearly the best performers.
- **Model Category II.** Due to the equivalence of the Averaged Projections algorithm to Alternating Projections (AP - Algorithm 3.5) and Projected Gradient (PG - Algorithm 3.6) on the product space, the performance of these algorithms is the same as that for AvP. In addition to these, we tested Fast Projected Gradient (FPG - Algorithm 3.7), the product space implementation of Douglas–Rachford (DR - Algorithm 3.3), Relaxed Douglas–Rachford (DR λ - Algorithm 3.8), and the DRAP (Algorithm 3.9) variant. The relative performance of these algorithms is very similar to the JWST experiments. There does not appear to be a significant qualitative difference between 1- and 2-dimensional experiments, though FPG does appear to require significantly more iterations for the 2-dimensional experiments. The DR and CDR algorithms are unstable for the 2-dimensional signals where for the 1-dimensional signals they

are more stable. This could indicate larger basins of attraction for 1-dimensional compared to 2-dimensional phase retrieval, but otherwise it does not point to any more interesting qualitative difference that we can imagine.

- **Model Category III.** The benchmarked algorithms belonging to this category are Averaged Projections (AvP - Algorithm 3.11), Dynamically Reweighted Averaged Projections (DyRePr - Algorithm 3.12), the ADMM-motivated Two-Step Averaged Projections recursion (AvP² - Algorithm 3.13), Wirtinger Flow (WF - Algorithm 3.14), and Limited Memory Quasi-Newton acceleration of Averaged Projections (QNAvP - Algorithm 3.15). The WF algorithm performed better on these experiments than with the JWST data, converging to within the tolerance of the true solution at least 85% of the time (column “failure”), though interestingly the warm start procedure recommended in [18] made things worse, both for 1- and 2-dimensional examples. The iteration counts to reach the stopping criteria were among the highest with little variability when it indeed converged (it did not always converge). The other algorithms performed similarly to the JWST experiment. The warm start sometimes made an improvement in iteration counts to the stopping criteria and only rarely degraded performance. The 1-dimensional experiments required fewer iterations to achieve a more stringent stopping criterion than the 2-dimensional experiments. The only algorithms to fail to achieve the required distance to the true solution were QNAvP (failed between 3% and 12% of the time on 2-dimensional experiments) and DyRePr (failed to reach the required distance to the true solution 100% of the time, though, as with the JWST experiment, the fixed points still *looked* good). Neither of these algorithms failed to find the true solution for the 1-dimensional CDP experiments. The ambiguity of the failure of the DyRePr algorithm is marked with a “100*” in the “failure” column of Table 4.

4.3. Sparse phase retrieval. The first to examine sparsity constrained phase retrieval was Marchesini in [55], where he recognized the implicit use of sparsity and hard thresholding in phase retrieval in the *Charge Flipping* algorithm of Oszlányi and Sütő [62, 63]. Since then, sparse phase retrieval has received intense study. More than one quarter of the results returned from a web query of the term “phase retrieval” are about sparse phase retrieval. Our contribution to this wave includes the recent studies [66, 12]. All of the algorithms surveyed in the previous section have analogues as more general iterated prox algorithms (the projection onto a set is just the prox of the indicator function), and so the projections can be replaced quite easily with hard and soft thresholders which are, respectively, the prox mappings of the counting function ($\|x\|_0 = \sum_{j=1}^n \|x_j\|_0$) and the ℓ_1 -norm.

The sparsity example constructed for our experiment is modeled after [22]. We generate a fixed number of Gaussian densities of different height and width randomly placed on a 128×128 discretization of the plane. The data, $b_{ij} \in \mathbb{R}^{128^2}$ in problem (2.1) with $n = 128^2$ and $m = 1$, is the amplitude of the discrete Fourier transform of the collection of Gaussian distributions. Instead of promoting sparsity using the counting function or the ℓ_1 -norm, we specify an estimate of the maximum number, s , of nonzero entries (which for our experiments was an overestimate of the true sparsity by 20%) and project onto the set

$$S_s \equiv \left\{ z \in (\mathbb{R}^2)^{128^2} \mid \|z\|_0 \leq s \right\}.$$

To this we add the constraint that the object is real and nonnegative, i.e., belongs to
(4.1)

$$C_+ \equiv \left\{ z = (z_1, z_2, \dots, z_{128^2}) \in (\mathbb{R}^2)^{128^2} \mid z_j = (x_j, 0) \in \mathbb{R}^2, x_j \in [0, \infty) \ (1 \leq j \leq 128^2) \right\}.$$

The qualitative constraint is then

$$C_0 \equiv C_+ \cap S_s.$$

It is easy to show that C_0 is a cone, and so this constraint fits the feasibility format of the cone and sphere problem. Moreover, it is easy to show that $P_{S_s} P_{C_+} = P_{S_s \cap C_+}$. As such, the analysis for the other instances of this problem demonstrated above also apply here. It is possible to incorporate real-nonnegativity constraints into the hard- and soft-thresholding operators, but since these operators showed no advantage over our formulation here, we do not include this comparison.

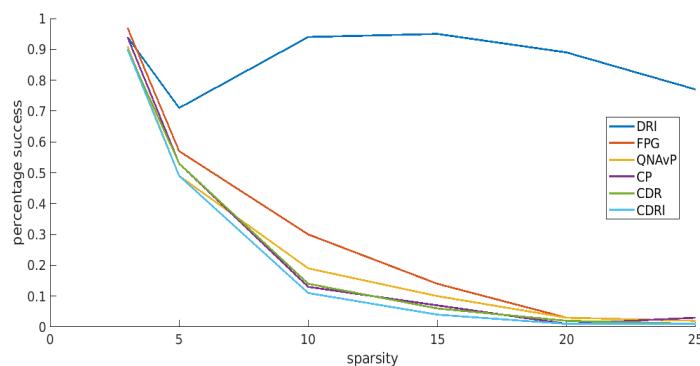


Figure 2. Leading algorithms applied to the feasibility formulation of sparse phase retrieval with nonnegative real sparse objects. Shown are the percentage of successful exact recoveries of the support of the sparse object (up to shifts and reflections) over 100 random trials for objects with 3, 5, 10, 15, 20, and 25 Gaussian “dots.”

For this experiment, we only compare in Figure 2 the most successful of the algorithms from the previous experiments. The goal here is to compare robustness of the best algorithms against local minimums. The algorithm was judged to have succeeded if it correctly identified the support of the Gaussian dots (modulo reflections and translations—phase retrieval is not unique) up to an error of 5×10^{-4} . The algorithms in the comparison are CP, CDR, and CDR λ (Model Category I), FPG and DR λ (Model Category II), and QNAvP (Model Category III). The DR λ algorithm stands out in these experiments as being the only algorithm to maintain a relatively high percentage of successful support recovery as the sparsity decreased (no less than 70% of the 100 random trials for each sparsity level). The other algorithms performed more or less the same, dropping to between 10% reliability (CDR λ) and 30% reliability (FPG) at a sparsity of 10 Gaussian dots and a reliability of around 5% for 20 Gaussian dots. This is a two-set feasibility problem, and so the advantage of the cyclic algorithms CDR and CDR λ is not clear. What is not reflected in these graphs is the average number of iterations each algorithm required to reach convergence (they all converged). The relative performance followed the same pattern as the other experiments above, with efficiencies differing by up to two orders of

magnitude. The algorithms not included in the comparison did not successfully recover the support and so were not considered.

4.4. Source localization. The sensors are located on a 100×100 unit grid, and for the noisy experiments the placement of the sensors has an error of about 3 units (SNR 30). The data consists of $m = 3$ or 10 measurements $b_{1j} \in \mathbb{R}$, $j = 1, 2, \dots, m$, with corresponding shift operators \mathcal{P}_j , $j = 1, 2, \dots, m$, defined by the randomly determined locations of the sensors (see problem (2.2)). All algorithms are started from the same randomly chosen initial points. All algorithms are terminated once the difference between successive iterates falls below $1e - 11$ or a maximum iteration count (10000) is exceeded. The algorithm is judged to have failed if it does not achieve a distance of $1e - 1$ to the true source location for noiseless data and 3 for noisy signals before the iterate difference tolerance is reached. The abstract formulation of the source localization problem has the same qualitative geometry as phase retrieval, and the numerical results shown in Tables 5 and 6 do not indicate any differences in the relative quantitative performance of the algorithms.

Table 5

100 random instances of three-sensor source localization problems, with and without noise. The “*” for DyRePr indicates that the algorithm’s fixed point is still reasonably close to the truth, even if it does not lie within the specified distance tolerance.

Three sensors	No noise				Noise			
	iteration counts				iteration counts			
	failure	median	high	low	failure	median	high	low
Model Category I								
CP	0	183.5	10000	9	0	169.5	7867	11
CDR	0	65	10000	8	0	68	2871	8
CDR λ ($\lambda = 0.33$)	0	150	10000	8	0	149	6591	10
Model Category II								
FPG	0	847.5	10000	97	0	961	10000	103
DR	0	2248	10000	140	0	2320	10000	159
DR λ ($\lambda = 0.85/0.5$)	0	215.5	10000	90	0	2825	10000	241
DRAP ($\lambda = 0.55$)	0	761	10000	47	0	498.5	10000	45
Model Category III								
AP/AvP/PG	0	1001.5	10000	68	0	969	10000	75
AvP 2 ($\rho_j = .3$)	13	3724.5	10000	274	100	3723	10000	302
DyRePr	100*	7	23	5	0	6.5	23	5
QNAvP	1	47.5	995	14	0	38	956	5

A summary of the benchmarks is given below.

- **Model Category I.** The benchmarked algorithms in this category are Cyclic Projections (CP - Algorithm 3.1), Cyclic Douglas–Rachford (CDR - Algorithm 3.2), and Relaxed Cyclic Douglas–Rachford (CDR λ - Algorithm 3.3). This category of algorithms again performed the best overall, never failing to achieve the required tolerance to the true solution. The more overdetermined the problem (10 sensors) the better the algorithms performed both in iteration counts to the stopping criteria and in variability of those counts. That the maximum iteration count was achieved during these experiments indicates that it is possible these algorithms could have sometimes con-

Table 6

100 random instances of 10-sensor source localization problems, with and without noise. The “*” for DyRePr indicates that the algorithm’s fixed point is still reasonably close to the truth, even if it does not lie within the specified distance tolerance.

10 sensors	No noise				Noise			
	iteration counts				iteration counts			
	failure	median	high	low	failure	median	high	low
Model Category I								
CP	0	10.5	55	5	0	11	61	5
CDR	0	9	54	4	0	9	51	4
CDR λ ($\lambda = 0.33$)	0	9	43	4	0	9	52	4
Model Category II								
FPG	0	168.5	518	93	0	182.5	764	100
DR	0	193.5	1197	143	0	10000	10000	10000
DR λ ($\lambda = 0.85/0.5$)	0	132.5	209	102	0/0	146.5	722	68
DRAP ($\lambda = 0.55$)	0	104	431	49	0	107.5	547	47
Model Category III								
AP/AvP/PG	0	141.5	570	71	0	146.5	722	68
AvP 2 ($\rho_j = .3$)	0	550.5	2141	284	90	569.5	10000	273
DyRePr	100*	7	16	5	0	7	16	5
QNAvP	0	17	889	12	0	33	46	13

verged to a limit cycle instead of a fixed point. But that the end points were without fail within the error tolerance of the true solution shows that the limit cycle was still in a good region. The three-sensor example is the only experiment where this appears to be a possibility, and this is reasonable since three sensors might not be enough to uniquely determine the source location. The 10-sensor experiment is overdetermined, and therefore one would expect this to be more robust against noise. The results from this model category confirm this.

- **Model Category II.** Due to the equivalence of the Averaged Projections algorithm to Alternating Projections (AP - Algorithm 3.5) and Projected Gradient (PG - Algorithm 3.6) on the product space, the performance of these algorithms is the same as that for AvP. In addition to these, we tested Fast Projected Gradient (FPG - Algorithm 3.7), the product space implementation of Douglas–Rachford (DR - Algorithm 3.3), Relaxed Douglas–Rachford (DR λ - Algorithm 3.8), and the DRAP (Algorithm 3.9) variant. It is remarkable that the Douglas–Rachford algorithm converges sometimes to a *fixed point* with three sensors and noise. We know that the Douglas–Rachford algorithm does not possess a fixed point for inconsistent feasibility problems, and so the fact that it converged in the noisy three-sensor experiment indicates that even with noise, the three-sensor case can still have nonempty intersection. Even when the intersection is empty, however, note that the iterates of the Douglas–Rachford algorithm, while they do not converge, never wander very far from an acceptable solution.
- **Model Category III.** The benchmarked algorithms belonging to this category are Averaged Projections (AvP - Algorithm 3.11), Dynamically Reweighted Averaged Projections (DyRePr - Algorithm 3.12), the ADMM-motivated Two-Step Averaged

Projections recursion (AvP² - Algorithm 3.13), and Limited Memory Quasi-Newton acceleration of Averaged Projections (QNAvP - Algorithm 3.15). The Wirtinger Flow algorithm does not extend to the source localization problem. The algorithms performed similarly on these experiments as with the phase data. The variability in the iteration counts to the stopping criteria was considerably higher than for the phase problems with the median iteration counts considerably higher for AvP and AvP². The sensitivity of AvP² to the parameter value ρ is shown in the noisy examples where it failed to achieve the prescribed distance to the true solution at least 90% of the time. The clear winner in this category is the quasi-Newton acceleration of AvP (QNAvP) algorithm. This algorithm failed only once to get within the required tolerance of the true solution, both with noisy and exact data, and achieved this in almost 25 times fewer iterations than AvP. Considering the per-iteration extra cost of the QNAvP, this results in a factor 5 speedup in CPU time—worth the effort.

4.5. Experimental data. We compare the quality of the most successful algorithms above on three sets of publicly available experimental data. These datasets are all similar in the sense that they consist of single images together with a simple a priori qualitative constraint. The first experiment, shown in Figure 3 (real part of the iterate at termination) was studied and developed for [56]. This is not a Fourier imaging dataset, but the (linear) imaging operator was provided with the image data. The second experiment shown in Figure 4 is a far-field optical diffraction imaging experiment. The third dataset shown in Figure 6 is a near-field X-ray hologram. The latter two experimental datasets are available from [48].

The first dataset is provided with a ground truth; there is no ground truth for the other experiments. The point of these experiments is to reproduce the kinds of environments and data for which these algorithms are intended; and in these settings, one only has mathematical theory as a guide for determining when to stop the algorithm and why.

- **Model Category I.** The benchmarked algorithms in this category are Cyclic Projections (CP - Algorithm 3.1), Cyclic Douglas–Rachford (CDR - Algorithm 3.2), and Relaxed Cyclic Douglas–Rachford (CDR λ - Algorithm 3.3). The algorithms quickly converged to the images shown. The contrast is not as good as with the other algorithms, but otherwise there is no observable difference in fidelity.
- **Model Category II.** Since there are only two constraint sets, we tested the non-product space implementation of the Douglas–Rachford (DR - Algorithm 3.3), Relaxed Douglas–Rachford (DR λ - Algorithm 3.8), and DRAP (Algorithm 3.9) variants. These also quickly converged to the images shown. The contrast is much better with these methods than with the cyclic implementations in the first model category.
- **Model Category III.** The benchmarked algorithms belonging to this category are Wirtinger Flow (WF - Algorithm 3.14), Averaged Projections (AvP - Algorithm 3.11), Dynamically Reweighted Averaged Projections (DyRePr - Algorithm 3.12), and Limited Memory Quasi-Newton acceleration of Averaged Projections (QNAvP - Algorithm 3.15). For this data, only the Averaged Projections algorithm retrieves a reasonable image.

The next dataset is an optical diffraction image produced by undergraduates at the Institute for X-Ray Physics at the University of Göttingen shown in Figure 4. This is a difficult dataset

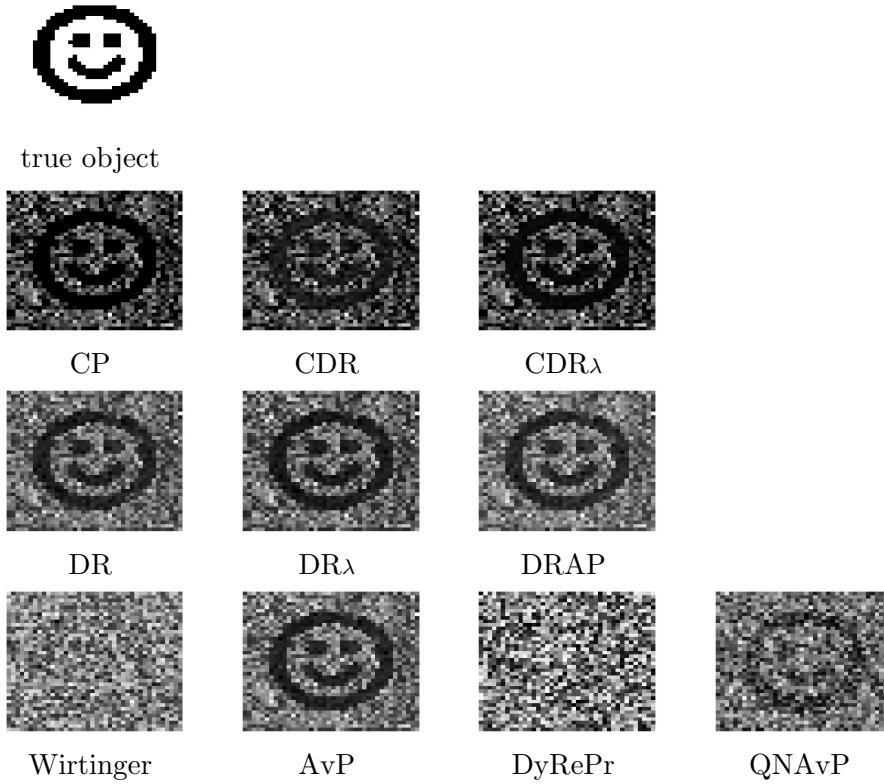


Figure 3. Representative phase reconstructions from algorithms with the Phasepack dataset PhaseSLM_40x40. Shown are the real parts of the final iterates of the respective algorithms.

because it is very noisy, and the physical parameters of the image (magnification factor, Fresnel number, etc.) were unknown to us. We optimistically assumed a perfect imaging system so that the imaging model is simply an unmodified Fourier transform. The object was a coffee cup: a real, nonnegative object, supported on some patch in the object plane; that is, the qualitative constraint C_0 is of the form (4.1). In such an experiment, one only has the successive iterates and a feasibility gap to observe in order to conclude that the algorithm is converging at least to a local best approximation point. The step size and feasibility gap are shown in Figure 5. Given the noise in the image, it is not clear that one desires a local best approximation point with the smallest feasibility gap since this will also mean that the noise has been recovered. For the numerical test reported here, we applied a low-pass filter to the data since almost all of the recoverable information about the object was contained in the low-frequency elements. This also had the numerically beneficial effect of rendering the problem *more inconsistent*. Thanks to the analysis in [52], we now understand why this can be helpful. All algorithms achieve the same gap distance between the sets, indicating that they attain points with the same local best approximation between the sets.

Noisy experiments like this one also bring to the foreground the issue of *regularization*. In [46], approximate projection algorithms for regularized sets were analyzed for cyclic projec-

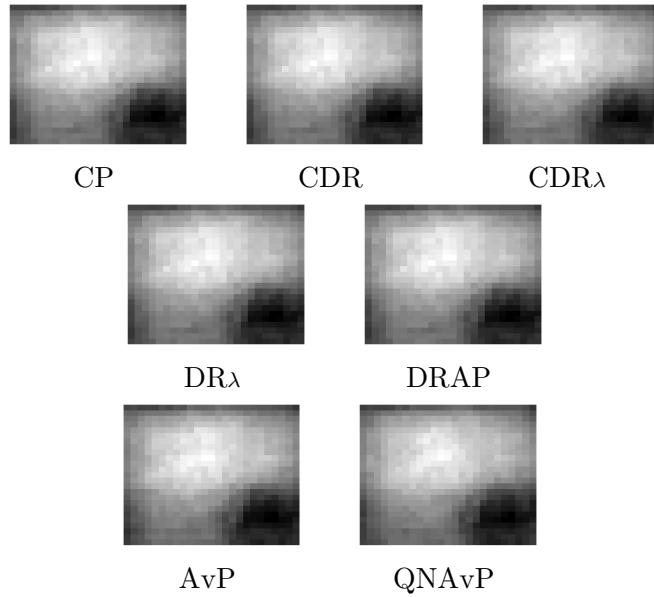


Figure 4. Object recovery of an optical diffraction experiment. The original object was a coffee cup. The recovered objects shown are upside-down images of the coffee cup with the handle on the left.

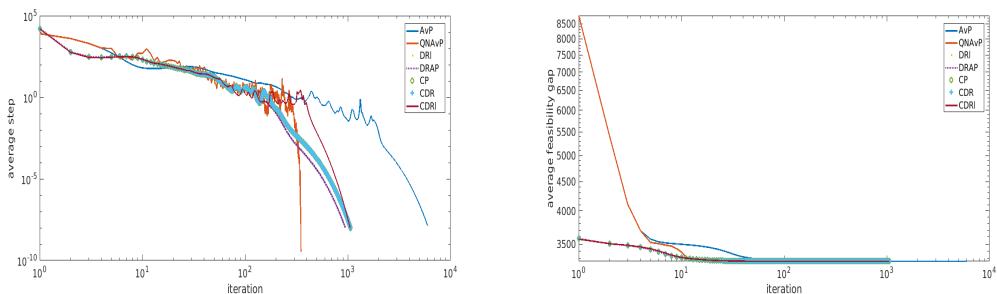


Figure 5. Step size and gap size between constraint sets versus iteration for several algorithms.

tions. The ideas of that work can be extended to the other algorithms considered here, namely that one can place balls (either Euclidean or Kullback–Leibler, as appropriate) around the measured data and project onto these “inflated” sets. Since the projections onto such sets is generally much more complicated to compute than the original sets, one can replace the exact projection with the projection onto the original, unregularized dataset, treating the latter projection as an approximation to the former. This has additional advantages of *extrapolation*, which yields *finite termination* at a feasible points in this case.

The payoff for early termination is demonstrated in Figures 6–8 with a near-field holography experiment provided to us by Tim Salditt’s laboratory at the Institute for X-Ray Physics at the University of Göttingen [76]. Here the structured illumination shown in Figure 6(a) left is modeled by \mathcal{P}_j , $j = 1, 2, \dots, m$, in problem (2.1) with $m = 1$. The image $-b_{ij}$, $j = 1, 2, \dots, m$, in problem (2.1) with $m = 1$, shown in Figure 6(a) right, is in the near field,

and so the mapping \mathcal{F} in problem (2.1) is the *Fresnel transform* [34]. The qualitative constraint is that the field in the object domain has amplitude 1 at each pixel, that is, the object is a *pure phase object*. A support constraint is not applied. Without regularization/early termination, the noise is recovered as shown in Figure 6(c). The data is regularized by accepting points within a fixed pointwise distance to the measured data at each pixel with respect to the Kullback–Leibler divergence. Rather than projecting onto these Kullback–Leibler balls, all algorithms compute the unregularized projection and move to the point given by the unregularized algorithm. The stopping rule, however, is with respect to the achieved feasibility gap between the regularized datasets and the qualitative constraint. This is effectively an early stopping rule for the unregularized algorithms. The result for different algorithms is shown in Figures 7 and 8.

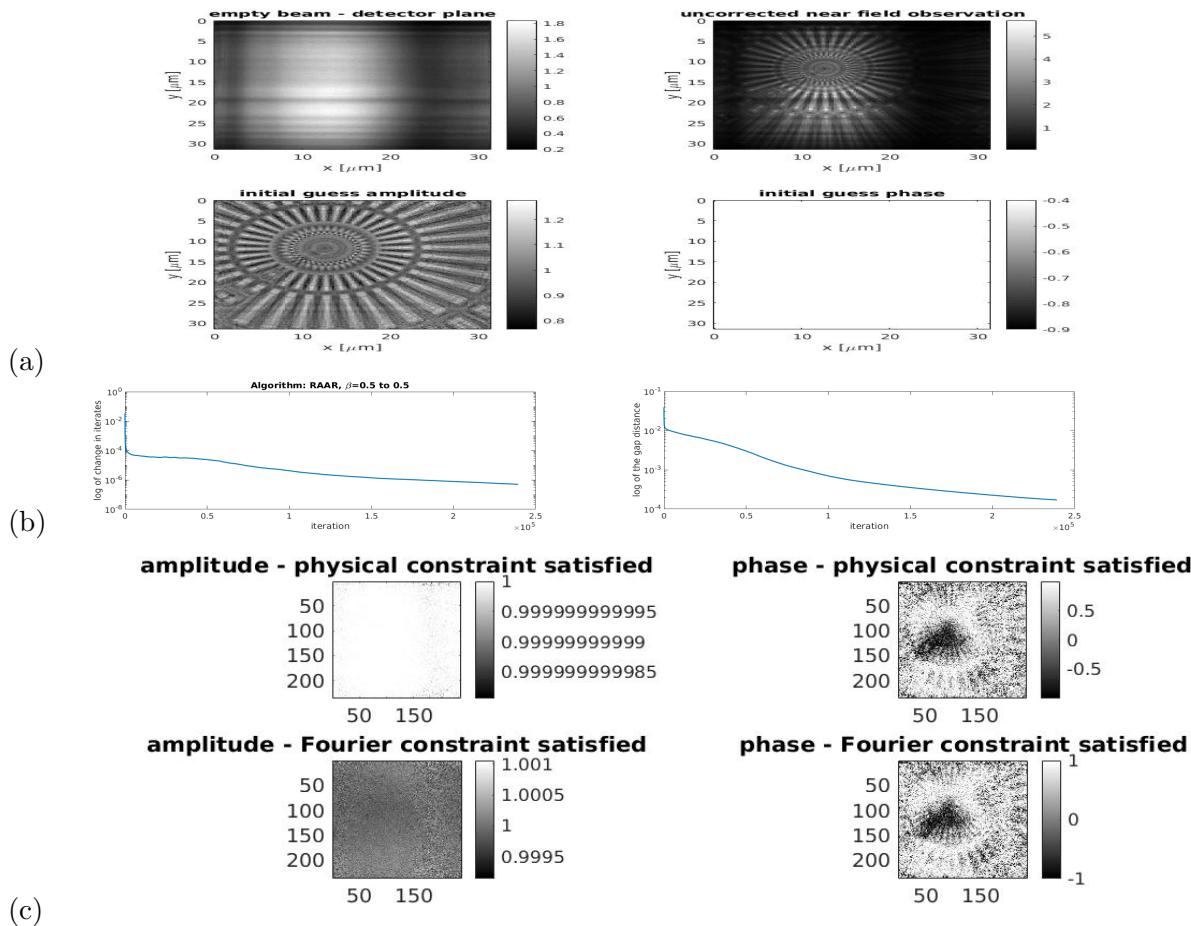


Figure 6. Unregularized reconstruction of a near-field holography experiment with empty beam correction using the $D\mathcal{R}\lambda$ algorithm with $\lambda = 0.5$. The algorithm is stable and converges as predicted by the theory; however, the noise is reconstructed without regularization of the dataset.

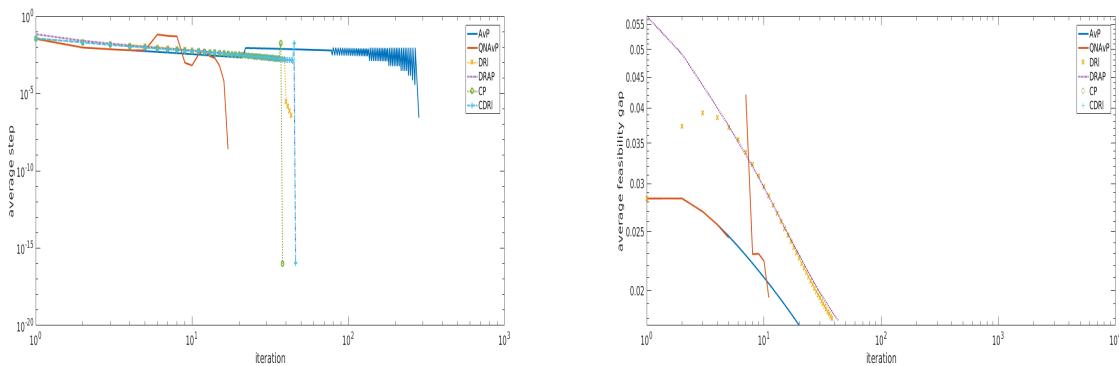


Figure 7. Step size and gap size between the constraint sets versus iteration.

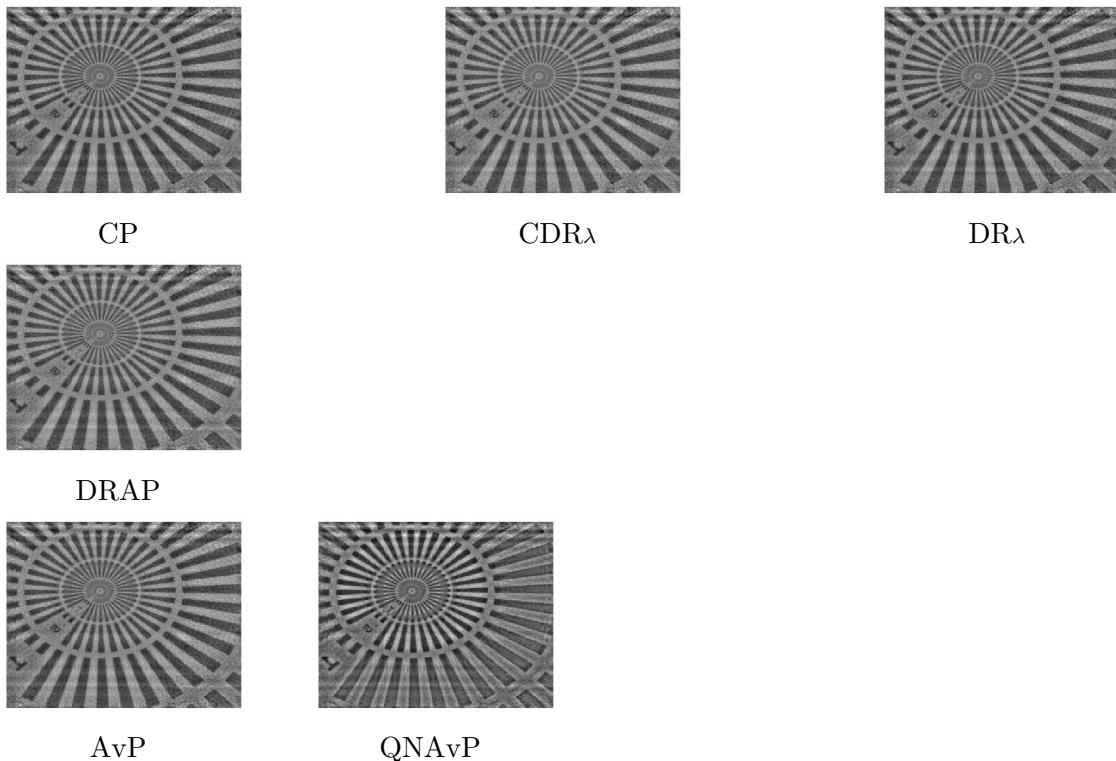


Figure 8. Reconstructed phase of a regularized near-field holography experiment with empty beam correction for the same data shown in Figure 6(a).

Acknowledgments. The authors thank the Erwin Schrödinger Institute for Mathematics and Physics at the University of Vienna for the stimulating environment at the final stages of this work. They also thank Katharina Echternkamp, Ann-Kathrin Günther, Daja Herzog, Dong-Du Mai, Jelena Panke, Aike Ruhlandt, and Jan Thiart at the Institut für Röntgenphysik

at the University of Göttingen for the diffraction data used in the numerical experiment in Figure 4. Finally, they thank Robin Wilke at the Institut für Röntgenphysik at the University of Göttingen for the diffraction data used in the numerical experiment in Figure 6.

REFERENCES

- [1] H. ATTOUCH, J. BOLTE, AND B. F. SVAITER, *Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods*, Math. Program., 137 (2013), pp. 91–129.
- [2] S. BAHMANI AND J. ROMBERG, *A flexible convex relaxation for phase retrieval*, Electron. J. Stat., 11 (2017), pp. 5254–5281.
- [3] H. H. BAUSCHKE, P. L. COMBETTES, AND D. R. LUKE, *A hybrid projection reflection method for phase retrieval*, J. Opt. Soc. Amer. A., 20 (2003), pp. 1025–1034.
- [4] H. H. BAUSCHKE, P. L. COMBETTES, AND D. R. LUKE, *Finding best approximation pairs relative to two closed convex sets in Hilbert spaces*, J. Approx. Theory, 127 (2004), pp. 178–192.
- [5] H. H. BAUSCHKE, D. R. LUKE, H. M. PHAN, AND X. WANG, *Restricted normal cones and sparsity optimization with affine constraints*, Found. Comput. Math., 14 (2014), pp. 63–83.
- [6] H. H. BAUSCHKE, D. NOLL, AND H. M. PHAN, *Linear and strong convergence of algorithms involving averaged nonexpansive operators*, J. Math. Anal. Appl., 421 (2015), pp. 1–20.
- [7] A. BECK AND D. PAN, *On the solution of the GPS localization and circle fitting problems*, SIAM J. Optim., 22 (2012), pp. 108–134, <https://doi.org/10.1137/100809908>.
- [8] A. BECK AND M. TEBOULLE, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202, <https://doi.org/10.1137/080716542>.
- [9] A. BECK, M. TEBOULLE, AND Z. CHIKISHEV, *Iterative minimization schemes for solving the single source localization problem*, SIAM J. Optim., 19 (2008), pp. 1397–1416, <https://doi.org/10.1137/070698014>.
- [10] J. BOLTE, S. SABACH, AND M. TEBOULLE, *Proximal alternating linearized minimization for nonconvex and nonsmooth problems*, Math. Program., 146 (2014), pp. 459–494.
- [11] J. BOLTE, S. SABACH, AND M. TEBOULLE, *Nonconvex Lagrangian-based optimization: Monitoring schemes and global convergence*, Math. Oper. Res., 43 (2018), pp. 1210–1232.
- [12] J. BOLTE, S. SABACH, M. TEBOULLE, AND Y. VAISBOURD, *First order methods beyond convexity and Lipschitz gradient continuity with applications to quadratic inverse problems*, SIAM J. Optim., 28 (2018), pp. 2131–2151, <https://doi.org/10.1137/17M1138558>.
- [13] J. M. BORWEIN AND M. K. TAM, *A cyclic Douglas–Rachford iteration scheme*, J. Optim. Theory Appl., 160 (2014), pp. 1–29.
- [14] J. M. BORWEIN AND M. K. TAM, *The cyclic Douglas–Rachford method for inconsistent feasibility problems*, J. Nonlinear Convex Anal., 16 (2015), pp. 537–584.
- [15] Y. M. BRUCK AND L. G. SODIN, *On the ambiguity of the image reconstruction problem*, Opt. Commun., 30 (1979), pp. 304–308.
- [16] J. V. BURKE AND A. WIEGMANN, *Low-Dimensional Quasi-Newton Updating Strategies for Large-Scale Unconstrained Optimization*, Technical report, Department of Mathematics, University of Washington, Seattle, WA, 1996.
- [17] R. H. BYRD, J. NOCEDAL, AND R. B. SCHNABEL, *Representations of quasi-Newton matrices and their use in limited memory methods*, Math. Programming, 63 (1994), pp. 129–156.
- [18] E. J. CANDES, X. LI, AND M. SOLTANOLKOTABI, *Phase retrieval via Wirtinger flow: Theory and algorithms*, IEEE Trans. Inform. Theory, 61 (2015), pp. 1985–2007.
- [19] Y. CENSOR AND A. CEGIELSKI, *Projection methods: An annotated bibliography of books and reviews*, Optimization, 64 (2015), pp. 2343–2358.
- [20] Y. CENSOR AND M. ZAKNOON, *Algorithms and convergence results of projection methods for inconsistent feasibility problems: A review*, Pure Appl. Funct. Anal., 3 (2018), pp. 565–586.
- [21] R. CHANDRA, Z. ZHONG, J. HONTZ, V. McCULLOCH, C. STUDER, AND T. GOLDSTEIN, *Phasepack: A phase retrieval library*, in Proceedings of the 51st Asilomar Conference on Signals, Systems, and Computers, 2017.

- [22] H. CHANG, S. MARCHESEINI, Y. LOU, AND T. ZENG, *Variational phase retrieval with globally convergent preconditioned proximal algorithm*, SIAM J. Imaging Sci., 11 (2018), pp. 56–93, <https://doi.org/10.1137/17M1120439>.
- [23] K. W. CHEUNG, H. C. SO, W.-K. MA, AND Y. T. CHAN, *Least squares algorithms for time-of-arrival-based mobile location*, IEEE Trans. Signal Process., 52 (2004), pp. 1121–1128.
- [24] A. A. COELHO, *A charge-flipping algorithm incorporating the tangent formula for solving difficult structures*, Acta Crystallogr. Sect. A, 63 (2007), pp. 400–406.
- [25] J. DOUGLAS, JR., AND H. H. RACHFORD, JR., *On the numerical solution of heat conduction problems in two or three space variables*, Trans. Amer. Math. Soc., 82 (1956), pp. 421–439.
- [26] V. ELSER, *Phase retrieval by iterated projections*, J. Opt. Soc. Amer. A, 20 (2003), pp. 40–55.
- [27] J. R. FIENUP, *Reconstruction of an object from the modulus of its Fourier transform*, Opt. Lett., 3 (1978), pp. 27–29.
- [28] J. R. FIENUP, *Phase retrieval algorithms: A comparison*, Appl. Opt., 21 (1982), pp. 2758–2769.
- [29] F. FRESCHEI, *Localization of sources of brain activity: A MILP approach*, IEEE Trans. Magn., 46 (2010), pp. 3429–3432.
- [30] D. GABAY, *Applications of the method of multipliers to variational inequalities*, in Augmented Lagrangian Methods, Applications to the Solution of Boundary-Value Problems, North-Holland, Amsterdam, 1983, pp. 299–331.
- [31] R. W. GERCHBERG AND W. O. SAXTON, *A practical algorithm for the determination of phase from image and diffraction plane pictures*, Optik, 35 (1972), pp. 237–246.
- [32] R. GLOWINSKI AND A. MARROCO, *Sur l'approximation, par éléments finis d'ordre un, et la résolution, par penalisation-dualité, d'une classe de problèmes de Dirichlet non linéaires*, Rev. Française Automat. Informat. Recherche Opérationnelle Sér. Rouge Anal. Numér., 9 (1975), pp. 41–76.
- [33] T. GOLDSTEIN AND C. STUDER, *Convex phase retrieval without lifting via PhaseMax*, in Proceedings of the 34th International Conference on Machine Learning, Vol. 70, Sydney, Australia, 2017, pp. 1273–1281.
- [34] J. HAGEMANN, A.-L. ROBISCH, D. R. LUKE, C. HOMANN, T. HOHAGE, P. CLOETENS, H. SUHONEN, AND T. SALDITT, *Reconstruction of wave front and object for inline holography from a set of detection planes*, Opt. Express, 22 (2014), pp. 11552–11569.
- [35] R. HESSE AND D. R. LUKE, *Nonconvex notions of regularity and convergence of fundamental algorithms for feasibility problems*, SIAM J. Optim., 23 (2013), pp. 2397–2419, <https://doi.org/10.1137/120902653>.
- [36] R. HESSE, D. R. LUKE, S. SABACH, AND M. K. TAM, *Proximal heterogeneous block implicit-explicit method and application to blind ptychographic diffraction imaging*, SIAM J. Imaging Sci., 8 (2015), pp. 426–457, <https://doi.org/10.1137/14098168X>.
- [37] A. S. LEWIS, D. R. LUKE, AND J. MALICK, *Local linear convergence of alternating and averaged projections*, Found. Comput. Math., 9 (2009), pp. 485–513.
- [38] A. S. LEWIS AND M. L. OVERTON, *Nonsmooth optimization via quasi-Newton methods*, Math. Program., 141 (2013), pp. 135–163.
- [39] G. LI AND T. K. PONG, *Global convergence of splitting methods for nonconvex composite optimization*, SIAM J. Optim., 25 (2015), pp. 2434–2460, <https://doi.org/10.1137/140998135>.
- [40] J. LIANG, P. STOICA, Y. JING, AND J. LI, *Phase retrieval via the alternating direction method of multipliers*, IEEE Signal Process. Lett., 25 (2018), pp. 5–9.
- [41] A. LOMBARD, Y. ZHENG, H. BUCHNER, AND W. KELLERMANN, *TDOA estimation for multiple sound sources in noisy and reverberant environments using broadband independent component analysis*, IEEE T. Audio Speech, 19 (2011), pp. 1490–1503.
- [42] S. LOOCK AND G. PLONKA, *Phase retrieval for Fresnel measurements using a shearlet sparsity constraint*, Inverse Problems, 30 (2014), pp. 1–17.
- [43] S. LOOCK AND G. PLONKA, *Iterative phase retrieval with sparsity constraints*, Proc. Appl. Math. Mech., 16 (2016), pp. 835–836.
- [44] D. R. LUKE, *Relaxed averaged alternating reflections for diffraction imaging*, Inverse Problems, 21 (2005), pp. 37–50.
- [45] D. R. LUKE, *Finding best approximation pairs relative to a convex and prox-regular set in a Hilbert space*, SIAM J. Optim., 19 (2008), pp. 714–739, <https://doi.org/10.1137/070681399>.

- [46] D. R. LUKE, *Local linear convergence of approximate projections onto regularized sets*, Nonlinear Anal., 75 (2012), pp. 1531–1546.
- [47] D. R. LUKE, *Phase retrieval, what's new?*, SIAG/OPT Views and News, 25 (2017), pp. 1–5.
- [48] D. R. LUKE, *ProxToolbox*, <http://num.math.uni-goettingen.de/proxtoolbox/>, 2017.
- [49] D. R. LUKE, J. V. BURKE, AND R. G. LYON, *Optical wavefront reconstruction: Theory and numerical methods*, SIAM Rev., 44 (2002), pp. 169–224, <https://doi.org/10.1137/S003614450139075>.
- [50] D. R. LUKE, A.-L. MARTINS, AND M. K. TAM, *Relaxed cyclic Douglas–Rachford algorithms for nonconvex optimization*, in ICML Workshop: Modern Trends in Nonconvex Optimization for Machine Learning, Stockholm, 2018.
- [51] D. R. LUKE, S. SABACH, M. TEBOULLE, AND K. ZATLAWEY, *A simple globally convergent algorithm for the nonsmooth nonconvex single source localization problem*, J. Global Optim., 69 (2017), pp. 889–909.
- [52] D. R. LUKE, N. H. THAO, AND M. K. TAM, *Quantitative convergence analysis of iterated expansive, set-valued mappings*, Math. Oper. Res., 43 (2018), pp. 1143–1176.
- [53] S. MARCHESEINI, *Phase retrieval and saddle-point optimization*, J. Opt. Soc. Amer. A, 24 (2007), pp. 3289–3296.
- [54] S. M. MARCHESEINI, *A unified evaluation of iterative projection algorithms for phase retrieval*, Rev. Sci. Instrum., 78 (2007), pp. 1–10.
- [55] S. M. MARCHESEINI, *Ab Initio Compressive Phase Retrieval*, preprint, <https://arxiv.org/abs/0809.2006v1>, 2008.
- [56] 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 Proceedings of the IEEE International Conference on Computational Photography (ICCP), 2017.
- [57] J. J. MOREAU, *Proximité et dualité dans un espace Hilbertian*, Bull. Soc. Math. France, 93 (1965), pp. 273–299.
- [58] Y. E. NESTEROV, *A method for solving the convex programming problem with convergence rate $O(1/k^2)$* , Dokl. Akad. Nauk SSSR, 269 (1983), pp. 543–547.
- [59] S. X.-Y. NI, M.-C. YUE, K.-F. CHEUNG, AND A. M.-C. SO, *Phase retrieval via sensor network localization*, J. Oper. Res. Soc. China, 7 (2019), pp. 127–146.
- [60] J. NOCEDAL AND S. WRIGHT, *Numerical Optimization*, Springer-Verlag, New York, 1999.
- [61] P. OCHS, Y. CHEN, T. BROX, AND T. POCK, *iPiano: Inertial proximal algorithm for nonconvex optimization*, SIAM J. Imaging Sci., 7 (2014), pp. 1388–1419, <https://doi.org/10.1137/130942954>.
- [62] G. OSZLÁNYI AND A. SÜTÓ, *Ab initio structure solution by charge flipping*, Acta Crystallogr. Ser. A, 60 (2004), pp. 134–141.
- [63] G. OSZLÁNYI AND A. SÜTÓ, *The charge flipping algorithm*, Acta Crystallogr. Ser. A, 64 (2008), pp. 123–134.
- [64] L. PALATINUS AND G. CHAPUIS, *SUPERFLIP - a computer program for the solution of crystal structures by charge flipping in arbitrary dimensions*, J. Appl. Crystallogr., 40 (2007), pp. 786–790.
- [65] P. PATRINOS, L. STELLA, AND A. BEMPORAD, *Douglas–Rachford splitting: Complexity estimates and accelerated variants*, in Proceedings of the 53rd IEEE Conference on Decision and Control, 2014, pp. 4234–4239.
- [66] E. PAUWELS, A. BECK, Y. C. ELDAR, AND S. SABACH, *On Fienup methods for sparse phase retrieval*, IEEE Trans. Signal Process., 66 (2018), pp. 982–991.
- [67] T. POCK AND S. SABACH, *Inertial proximal alternating linearized minimization (iPALM) for nonconvex and nonsmooth problems*, SIAM J. Imaging Sci., 9 (2016), pp. 1756–1787, <https://doi.org/10.1137/16M1064064>.
- [68] R. A. POLIQUIN, R. T. ROCKAFELLAR, AND L. THIBAULT, *Local differentiability of distance functions*, Trans. Amer. Math. Soc., 352 (2000), pp. 5231–5249.
- [69] A.-L. ROBISCH, K. KRÖGER, A. RACK, AND T. SALDITT, *Near-field ptychography using lateral and longitudinal shifts*, New J. Phys., 17 (2015), 073033.
- [70] A.-L. ROBISCH AND T. SALDITT, *Phase retrieval for object and probe using a series of defocus near-field images*, Opt. Express, 21 (2013), pp. 23345–23357.
- [71] R. T. ROCKAFELLAR AND R. J. WETS, *Variational Analysis*, Grundlehren Math. Wiss. 317, Springer-Verlag, Berlin, 1998.

- [72] R. SHEFI AND M. TEBOULLE, *Rate of convergence analysis of decomposition methods based on the proximal method of multipliers for convex minimization*, SIAM J. Optim., 24 (2014), pp. 269–297, <https://doi.org/10.1137/130910774>.
- [73] S. VAN SMAALEN, L. PALATINUS, AND M. SCHNEIDER, *The maximum-entropy method in superspace*, Acta Crystallogr. Ser. A, 59 (2003), pp. 459–469.
- [74] P. STOICA AND J. LI, *Source localization from range-difference measurements*, IEEE Signal Process. Mag., 23 (2006), pp. 63–66.
- [75] N. H. THAO, *A convergent relaxation of the Douglas–Rachford algorithm*, Comput. Optim. Appl., 70 (2018), pp. 841–863.
- [76] R. N. WILKE, M. PRIEBE, M. BARTELS, K. GIEWEKEMEYER, A. DIAZ, P. KARVINEN, AND T. SALDITT, *Hard X-ray imaging of bacterial cells: Nano-diffraction and ptychographic reconstruction*, Opt. Express, 20 (2012), pp. 19232–19254.
- [77] L.-H. YEH, J. DONG, J. ZHONG, L. TIAN, M. CHEN, G. TANG, M. SOLTANOLKOTABI, AND L. WALLER, *Experimental robustness of Fourier ptychography phase retrieval algorithms*, Opt. Express, 23 (2015), pp. 33214–33240.