Phase Retrieval Using Iterative Projections: Dynamics in the Large Systems Limit

Gen Li, Yuantao Gu
Department of Electronic Engineering
Tsinghua University
Beijing 100084, China

Yue M. Lu
Paulson School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138, USA

***Abstract*—We study phase retrieval by using a simple non-convex algorithm based on iterative projections. Our main contribution in this work is to establish an exact analysis of the dynamics of the algorithm in an online setting, with Gaussian measurement vectors. We show that the algorithm dynamics, measured by the squared distance between the current estimate and the true solution, can be fully characterized by a 2D Markov process, irrespective of the underlying signal dimension. Furthermore, in the large systems limit (*i.e.*, as the signal dimension tends to infinity), the random sample paths of the 2D Markov process converge to the solutions of two deterministic and coupled ordinary differential equations (ODEs). Numerical simulations verify the accuracy of our analytical predictions, even for moderate system sizes. This suggests that the ODE approach presented in this work provides an effective tool for analyzing the performance and convergence of the algorithm.**

I. Introduction

We consider the problem of *phase retrieval*. Let

$$y_m = |\langle \boldsymbol{a}_m, \boldsymbol{x}^* \rangle|, \qquad m = 1, 2, \ldots, M, \tag{1}$$

where $\boldsymbol{x}^* \in \mathbb{C}^N$ is an unknown signal, and $\left\{\boldsymbol{a}_m \in \mathbb{C}^N\right\}$ is a set of known measurement vectors. We aim to recover $\boldsymbol{x}^*$, up to a constant phase term, from the magnitude measurements $\{y_m\}$. The name "phase retrieval" comes from the fact that recovering $\boldsymbol{x}^*$ is equivalent to recovering the missing phases of the linear measurements $\langle \boldsymbol{a}_m, \boldsymbol{x}^* \rangle$.

The phase retrieval problem arises in many areas of science and engineering, such as X-ray crystallography, astronomy, diffraction imaging, and optics. Its frequent appearances in these fields are due to the limitations of most physical sensing devices: they can only record the intensities of an electromagnetic field, but not its phases.

There is a long line of research addressing this problem, starting from classical schemes based on error reduction (see, *e.g.*, [1], [2]). Alternating between estimates of the missing phase and those of the unknown signal $\boldsymbol{x}^*$, these simple iterative algorithms are often shown to be effective in empirical evaluations. However, they lack formal theoretical performance guarantees. A different line of work (*e.g.*, [3]–[6]) approaches the problem through "lifting" and convex relaxation, aiming to reconstruct a rank-one matrix $\boldsymbol{x}^*(\boldsymbol{x}^*)^H$, from which the unknown signal $\boldsymbol{x}^*$ can be obtained. When the sensing vectors $\{\boldsymbol{a}_m\}$ are drawn from certain random ensembles, these new methods, based on semidefinite programming, enjoy strong theoretical guarantees in terms of sample complexities. The challenges facing these schemes lie in their actual implementation. In practice, their computational complexity and memory requirement make it difficult to apply them to signal dimensions that are relevant to real-word applications such as imaging.

In light of these issues, there is recent interest in revisiting iterative methods that directly attack the phase retrieval problem in its original non-convex setting. Examples include the work of Netrapalli, Jain, and Sanghavi [7] in analyzing the alternating minimization scheme, and the Wirtinger Flow algorithm and its variants [8], [9]. In random acquisition settings and armed with suitable initializations [7], these non-convex iterative methods have been shown to achieve linear rate of convergence to the solution.

In this paper, we put forward a simple non-convex algorithm, based on iterative projections, to solve the phase retrieval problem. Unlike previous work such as [7]–[9], where each iteration of the algorithms involves all or a large block of the measurements, our algorithm only handles one measurement per iteration. This setting leads to a computational complexity of only $\mathcal{O}(N)$ per iteration, and is in the same vain as general row-action methods such as the Kaczmarz [10]–[12] and online stochastic gradient descent algorithms.

Our main contribution in this work is to establish an exact analysis of the dynamics of the algorithm in the large systems limit (*i.e.*, as the signal dimension N tends to infinity.) Let d_k be the squared distance between the estimate of the algorithm at the kth iteration and the true signal $\boldsymbol{x}^*$, we study the dynamics of the algorithm through the function $d(t) \overset{\text{def}}{=} d_{\lfloor tN \rfloor}$, where $t = k/N$ is the *rescaled* time variable. We show that, as N tends to infinity, the random sample paths of $d(t)$ will converge to a continuous time function governed by the solutions of two deterministic, coupled ordinary differential equations (ODEs) [see Proposition 2]. In other words, as long as N is sufficiently large, the original random dynamics of the algorithm can be accurately represented by a deterministic surrogate, the latter of which is fairly easy to analyze. In

The first author performed this work when he was a visiting student at the Harvard Paulson School of Engineering and Applied Sciences.

The work of Y. M. Lu was supported in part by the U.S. National Science Foundation under grant CCF-1319140 and in part by a Harvard-UTEC faculty grant.

```
Phase Retrieval by Iterative Projections
Input: $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ with rows $\boldsymbol{a}_1^T, \boldsymbol{a}_2^T, \ldots, \boldsymbol{a}_m^T$; $\boldsymbol{y} \in \mathbb{R}^M$
    such that $y_m = |\boldsymbol{a}_m^T \boldsymbol{x}^*|$ for some unknown signal $\boldsymbol{x}^*$;
    iteration count $K$.
Output: $\widehat{\boldsymbol{x}} \in \mathbb{R}^N$, an estimate for $\boldsymbol{x}^*$.
    Initialize $\boldsymbol{x}_0$ using the spectral method proposed in [7].
    for $k = 1$ to $K$ do
        Choose the $m$th row. The rows are chosen either sequentially or uniformly at random.
        $\boldsymbol{x}_k \leftarrow \boldsymbol{x}_{k-1} + \frac{y_m \operatorname{sgn}(\boldsymbol{a}_m^T \boldsymbol{x}_{k-1}) - \boldsymbol{a}_m^T \boldsymbol{x}_{k-1}}{\|\boldsymbol{a}_m\|^2} \boldsymbol{a}_m$
    end for
    $\widehat{\boldsymbol{x}} \leftarrow \boldsymbol{x}_K$.
```

particular, one can establish simple convergence bounds for our algorithm by analyzing properties of the underlying ODEs.

The rest of the paper is organized as follows: We present a simple iterative algorithm for phase retrieval in Section II. The algorithm is based on iterative projections, and we also establish its connections to Kaczmarz methods and Newton's algorithm. In Section III, we analyze the dynamics of this algorithm, and establish its ODE limit. Numerical simulations demonstrate the remarkable accuracy of our analytical predictions, even for moderate system sizes.

In this paper, we present our analysis for the real-valued version of the phase retrieval problem in (1) and assume that the measurements $\{y_m\}$ are noiseless. However, our analysis techniques are general and can be easily extended to the complex case and to noisy acquisition setups. These extensions will be reported in a forthcoming paper.

II. Phase Retrieval by Iterative Projections

Given a set of vectors $\left\{\boldsymbol{a}_m \in \mathbb{R}^N\right\}$ and measurements $\{y_m\}$, we present an iterative algorithm that attempts to find a solution $\boldsymbol{x}^* \in \mathbb{R}^N$ to (1).

The algorithm is listed above. The idea behind the algorithm is simple. In the real-valued case, the challenge of the phase retrieval problem lies in the missing sign information. If such information were available, we would then be dealing with linear measurements of the form

$$\widetilde{y}_m = \boldsymbol{a}_m^T \boldsymbol{x}^*$$

for $m = 1, 2, \ldots, M$. Each measurement defines a hyperplane on which the solution $\boldsymbol{x}^*$ must lie. To enforce all these linear constraints, we can select one of these hyperplanes at each iteration and project the iterand onto it, thus getting closer to the true solution with each step. Note that this is the same idea behind the Kaczmarz algorithm [10], also known under the name Algebraic Reconstruction Technique (ART) [13], a popular method for solving a large-scale overdetermined system of linear equations.

Let $\boldsymbol{x}_0$ be an initial guess. At each step k, a *single* hyperplane is chosen. We can choose these hyperplanes sequentially, starting with the first measurement, proceeding in succession to the last measurement, and then cycling back to the first; Or, they can be chosen uniformly at random at each iteration. In either case, let m_k be the index of the hyperplane chosen at the kth step. The previous iterand $\boldsymbol{x}_{k-1}$ is then projected onto the chosen hyperplane, and updated according to the formula

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \frac{\widetilde{y}_{m_k} - \boldsymbol{a}_{m_k}^T \boldsymbol{x}_{k-1}}{\|\boldsymbol{a}_{m_k}\|^2} \boldsymbol{a}_{m_k}. \tag{2}$$

To implement the above procedure, we need to have the linear measurements $\widetilde{y}_m$, which can be written as

$$\widetilde{y}_m = |\widetilde{y}_m| \operatorname{sgn}(\widetilde{y}_m) = y_m \operatorname{sgn}(\boldsymbol{a}_m^T \boldsymbol{x}^*).$$

For phase retrieval, we only have access to the absolute value y_m but not the sign. To proceed, we simply use $\operatorname{sgn}(\boldsymbol{a}_m^T \boldsymbol{x}_{k-1})$ to approximate the missing value $\operatorname{sgn}(\boldsymbol{a}_m^T \boldsymbol{x}^*)$. As a result, we modify (2) as

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \frac{y_{m_k} \operatorname{sgn}(\boldsymbol{a}_{m_k}^T \boldsymbol{x}_{k-1}) - \boldsymbol{a}_{m_k}^T \boldsymbol{x}_{k-1}}{\|\boldsymbol{a}_{m_k}\|^2} \boldsymbol{a}_{m_k}, \tag{3}$$

which is the actual update formula used in our algorithm.

The iterative procedure in (3) can also be interpreted in terms of Newton's method. To see this, let $\boldsymbol{x}_{k-1}$ be the current iterand, and suppose we want to find a new estimate $\boldsymbol{x}_k$ that satisfies a single equation

$$f(\boldsymbol{x}) \overset{\text{def}}{=} |\boldsymbol{a}_{m_k}^T \boldsymbol{x}| - y_{m_k} = 0.$$

Writing the Taylor series of $f(\boldsymbol{x})$ about the point $\boldsymbol{x}_{k-1}$ and keeping terms only to first order, we have

$$\begin{aligned} 0 &= f(\boldsymbol{x}_k) \\ &\approx f(\boldsymbol{x}_{k-1}) + (\boldsymbol{x}_k - \boldsymbol{x}_{k-1})^T \nabla f(\boldsymbol{x}_{k-1}) \\ &= |\boldsymbol{a}_{m_k}^T \boldsymbol{x}_{k-1}| - y_{m_k} + (\boldsymbol{x}_k - \boldsymbol{x}_{k-1})^T \operatorname{sgn}(\boldsymbol{a}_{m_k}^T \boldsymbol{x}_{k-1}) \boldsymbol{a}_{m_k}, \end{aligned}$$

which implies that

$$(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})^T \boldsymbol{a}_{m_k} = y_{m_k} \operatorname{sgn}(\boldsymbol{a}_{m_k}^T \boldsymbol{x}_{k-1}) - \boldsymbol{a}_{m_k}^T \boldsymbol{x}_{k-1}. \tag{4}$$

This expression does not uniquely specify the update step $\boldsymbol{x}_k - \boldsymbol{x}_{k-1}$, because we can add to it any vector that is orthogonal to $\boldsymbol{a}_{m_k}$ without changing the left-hand side of (4). However, if we seek an update that has the minimum norm, then $\boldsymbol{x}_k - \boldsymbol{x}_{k-1}$ must be fully aligned with $\boldsymbol{a}_{m_k}$, in which case we can obtain $\boldsymbol{x}_k$ as in (3).

III. Performance Analysis and Dynamics in the Large Systems Limit

In this section, we analyze the dynamics of the algorithm in an online (*i.e.*, single-pass) setting: Starting from the first measurement, the algorithm proceeds sequentially, processing each measurement $y_m = |\boldsymbol{a}_m^T \boldsymbol{x}^*|$ once and only once. The algorithm stops as soon as the last measurement is reached, so that the total number of iterations is exactly equal to the number of measurements M.

For online settings where the data arrive in streams, this single-pass assumption accurately models the behavior of the algorithm; For offline settings, requiring that the each measurement be used exactly once is a restriction, but it simplifies our theoretical analysis.

In our subsequent analysis, we also assume that the measurement vectors $\boldsymbol{a}_m$ are independent Gaussian vectors drawn from $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_N)$.

A. Markovian Dynamics and Order Parameters

Under the single-pass setting that we consider, the update formula in (3) can be simplified as

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \frac{y_k \operatorname{sgn}(\boldsymbol{a}_k^T \boldsymbol{x}_{k-1}) - \boldsymbol{a}_k^T \boldsymbol{x}_{k-1}}{\|\boldsymbol{a}_k\|^2} \boldsymbol{a}_k, \tag{5}$$

since $m_k = k$, *i.e.*, the kth measurement y_k is chosen at step k. This update formula implies that the iterand $\boldsymbol{x}_k$ of the algorithm follows a random dynamic process in $\mathbb{R}^N$, with the randomness coming from the Gaussian vectors $\{\boldsymbol{a}_k\}$. Since $\boldsymbol{a}_k$ is independent of the previous iterands $\boldsymbol{x}_{k-1}, \boldsymbol{x}_{k-2}, \ldots, \boldsymbol{x}_1, \boldsymbol{x}_0$ (again thanks to the single-pass nature of the algorithm), the dynamics is in fact a Markov process (random walk) in $\mathbb{R}^N$.

We measure the performance of the algorithm at the kth iteration in terms of

$$d_k \overset{\text{def}}{=} \min\left\{\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2, \|\boldsymbol{x}_k + \boldsymbol{x}^*\|^2\right\}, \tag{6}$$

taking into account the fact that both $\boldsymbol{x}^*$ and $-\boldsymbol{x}^*$ are valid solutions. Expanding the right-hand side of (6), we get

$$d_k = \|\boldsymbol{x}^*\|^2 + \|\boldsymbol{x}_k\|^2 - 2|\boldsymbol{x}_k^T \boldsymbol{x}^*|, \tag{7}$$

and thus the performance of the algorithm is fully specified by the three terms on the right-hand side of (7). As $\|\boldsymbol{x}^*\|$ is a fixed quantity, we just need to monitor the remaining two:

$$b_k \overset{\text{def}}{=} \|\boldsymbol{x}_k\|^2 \quad \text{and} \quad c_k \overset{\text{def}}{=} \boldsymbol{x}_k^T \boldsymbol{x}^*, \tag{8}$$

which are related to the square norm of the iterand $\boldsymbol{x}_k$ and to the angle made between $\boldsymbol{x}_k$ and $\boldsymbol{x}^*$, respectively. In what follows, we shall refer to b_k and c_k as "order parameters", borrowing this terminology from statistical physics.

B. Dimensionality Reduction

Next, we will show that the sequence $\{b_k, c_k\}$ forms a Markov process in $\mathbb{R}^2$.

Remark 1: This statement is significant, for two reasons. First, b_k and c_k are functions of $\boldsymbol{x}_k$, the latter of which forms a Markov process in $\mathbb{R}^N$. So in principle, $\{b_k, c_k\}$ follows a hidden Markov process, and its dynamics is not expected to form a closed loop. The fact that it actually does is surprising. Second, since $d_k = \|\boldsymbol{x}^*\|^2 + b_k - 2|c_k|$, our statement implies that the performance of the algorithm can be fully analyzed by studying a Markov process in $\mathbb{R}^2$, irrespective of the underlying signal dimension N and the full dynamics in $\mathbb{R}^N$. This represents a substantial dimensionality reduction.

Proposition 1: The dynamics of the order parameters $\{b_k, c_k\}$ are described by

$$b_k = b_{k-1} + \frac{u_k^2 - v_k^2}{p_k + z_k}, \tag{9}$$

$$c_k = c_{k-1} + \frac{u_k^2 \operatorname{sgn}(u_k) \operatorname{sgn}(v_k) - u_k v_k}{p_k + z_k}, \tag{10}$$

where $u_k, v_k, p_k, \boldsymbol{z}_k$ are random variables that are conditionally independent of the history $\{b_{k-2}, c_{k-2}\}, \ldots, \{b_0, c_0\}$, given the current values $\{b_{k-1}, c_{k-1}\}$. Moreover,

$$p(u_k, v_k | b_{k-1}, c_{k-1}) \sim \mathcal{N}(\mathbf{0}, \Sigma_k),$$

where $\Sigma_k \overset{\text{def}}{=} \begin{bmatrix} \|\boldsymbol{x}^*\|^2 & c_{k-1} \\ c_{k-1} & b_{k-1} \end{bmatrix}$; p_k is a deterministic function of $u_k, v_k, b_{k-1}, c_{k-1}$; and finally, z_k follows a χ^2-distribution with $\left(N - \dim(\operatorname{span}\{\boldsymbol{x}^*, \boldsymbol{x}_{k-1}\})\right)$ degrees of freedom, and it is conditionally independent of u_k, v_k, given $\{b_{k-1}, c_{k-1}\}$.

Proof: (sketch) Using the definitions in (8) and the updating formula in (5), we can easily verify that

$$b_k = b_{k-1} + \frac{u_k^2 - v_k^2}{\|\boldsymbol{a}_k\|^2}, \tag{11}$$

$$c_k = c_{k-1} + \frac{u_k^2 \operatorname{sgn}(u_k) \operatorname{sgn}(v_k) - u_k v_k}{\|\boldsymbol{a}_k\|^2}, \tag{12}$$

where $u_k = \boldsymbol{a}_k^T \boldsymbol{x}^*$ and $v_k = \boldsymbol{a}_k^T \boldsymbol{x}_{k-1}$ are the projections of an independent Gaussian vector $\boldsymbol{a}_k$ onto $\boldsymbol{x}^*$ and $\boldsymbol{x}_{k-1}$, respectively. A key observation is that, conditioned on $\{b_{k-1}, c_{k-1}\}$, the random variables u_k and v_k are jointly Gaussian with zero mean and covariance matrix Σ_k, and that they are independent of the past values $\{b_{k-2}, c_{k-2}\}, \ldots, \{b_0, c_0\}$. To see this, we write

$$\begin{aligned}
& p(u_k, v_k | b_{k-1}, c_{k-1}, b_{k-2}, c_{k-2}, \ldots, b_0, c_0) \\
&= \int p(u_k, v_k | \boldsymbol{x}_{k-1}, b_{k-1}, c_{k-1}, \ldots, b_0, c_0) \\
&\qquad \times p(\boldsymbol{x}_{k-1} | b_{k-1}, c_{k-1}, \ldots, b_0, c_0) \, \mathrm{d}\boldsymbol{x}_{k-1} \\
&= \int \frac{1}{\sqrt{\det(2\pi\Sigma_k)}} \exp(-[u_k, v_k] \Sigma_k^{-1} [u_k, v_k]^T / 2) \\
&\qquad \times p(\boldsymbol{x}_{k-1} | b_{k-1}, c_{k-1}, \ldots, b_0, c_0) \, \mathrm{d}\boldsymbol{x}_{k-1} \\
&= \frac{1}{\sqrt{\det(2\pi\Sigma_k)}} \exp(-[u_k, v_k] \Sigma_k^{-1} [u_k, v_k]^T / 2).
\end{aligned} \tag{13}$$

To finalize our proof, we just need to show that the denominator $\|\boldsymbol{a}_k\|^2$ on the right-hand sides of (11) and (12) can be written as $p_k + z_k$, with p_k and z_k satisfying the stated properties. To see that, we denote by $\mathbb{P}_{\mathcal{S}}$ the orthogonal projection operator onto the space spanned by $\boldsymbol{x}^*$ and $\boldsymbol{x}_k$. Let $p_k = \|\mathbb{P}_{\mathcal{S}} \boldsymbol{a}_k\|^2$, and $z_k = \|(\boldsymbol{I} - \mathbb{P}_{\mathcal{S}}) \boldsymbol{a}_k\|^2$. Clearly, $\|\boldsymbol{a}_k\|^2 = p_k + z_k$. It is also clear that p_k is a deterministic function of $u_k, v_k, b_{k-1}, c_{k-1}$. Finally, using similar derivations that lead to (13), we can show that z_k is the sum of $N - \dim(\operatorname{span}\{\boldsymbol{x}^*, \boldsymbol{x}_{k-1}\})$ independent standard Gaussian random variables, and thus it follows the χ^2-distribution. We omit the details here. ■

Example 1: We simulate the proposed algorithm under parameter settings $N = 400$, $M/N = 20$, and $\|\boldsymbol{x}^*\| = 1$. Figure 1 shows the dynamics of the order parameters $\{b_k, c_k\}$ in two independent realizations of the Gaussian sensing vectors. Due to the Cauchy-Schwarz inequality, the order parameters are bounded within the two blue curves $c = \pm\sqrt{b}$. Starting from different initial points, the two independent realizations converge towards the correction solutions, corresponding to $b = c = 1$ or $b = -c = 1$.

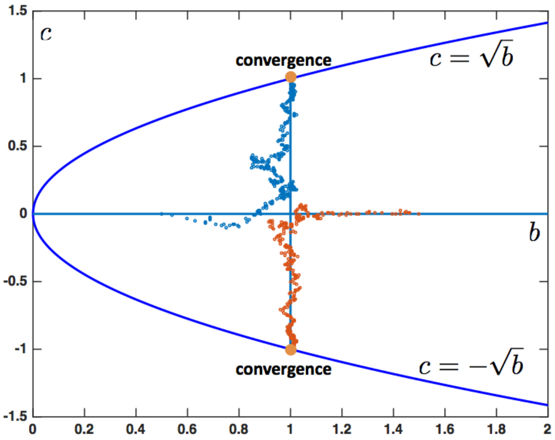

Fig. 1: The dynamics of the order parameters $\{b_k, c_k\}$. The blue and red dots illustrate the dynamics in two independent realizations of the Gaussian sensing vectors. The parameters used in this experiment are detailed in Example 1.

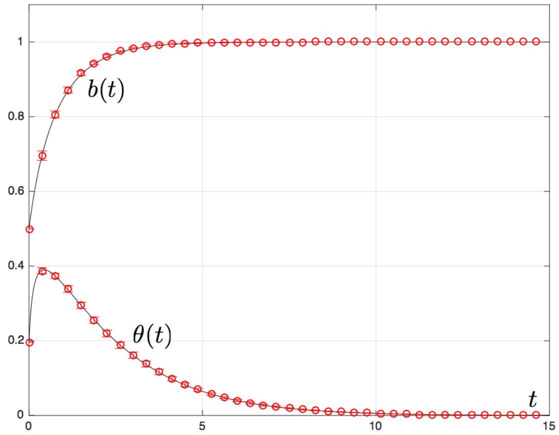

Fig. 2: The comparison between the analytical predictions given by the ODE limit and Monte Carlo simulations. The dynamics of $b(t)$ and $\theta(t) = \arccos(c(t)/\sqrt{b(t)})$ are shown, with the ODE solutions plotted as black solid lines and the average values of Monte Carlo simulations plotted as red dots. The error bars show the confidence intervals of ± one standard deviation.

C. The ODE Limit

The result of Proposition 1 ensures that we can analyze the dynamics of the algorithm by tracking a 2-D Markov process, irrespective of the signal dimension N. Although this represents a significant reduction in complexity, it is still not easy to work with the random recursion formulas in (9) and (10) since the underlying Markov process has a state space that is uncountably infinite.

Fortunately, it is possible to obtain even simpler characterizations of the algorithm dynamics by considering the large systems limit. To see this, we first notice that the update steps $b_k - b_{k-1}$ and $c_k - c_{k-1}$ in (9) and (10) are both random variables of order $\mathcal{O}(1/N)$. As N tends to infinity, the randomness of the associated Markov process will be reduced; in the limit, its sample path will converge in probability to deterministic functions that are governed by certain ODEs.

To properly state the large systems and ODE limit, we need to introduce some new notation: first, we write b_k^N and c_k^N so that the underlying signal dimension is explicitly specified by the superscript; second, we *rescale* the time axis by defining

$$b^N(t) \overset{\text{def}}{=} b^N_{\lfloor tN \rfloor} \quad \text{and} \quad c^N(t) \overset{\text{def}}{=} c^N_{\lfloor tN \rfloor}$$

where $t = k/N$ is the rescaled time variable.

Proposition 2: For a sequence of initial values $\{b_0^N, c_0^N\}$ satisfying $\lim_{N\to\infty} b_0^N = b(0)$ and $\lim_{N\to\infty} c_0^N = c(0) \neq 0$, we have

$$\lim_{N\to\infty} b^N(t) = b(t) \quad \text{and} \quad \lim_{N\to\infty} b^N(t) = b(t),$$

where the convergence is in probability, and $b(t), c(t)$ are the solutions of the following coupled ODEs with initial value $\{b(0), c(0)\}$:

$$\frac{\mathrm{d}b(t)}{\mathrm{d}t} = -b(t) + \|\boldsymbol{x}^*\|^2, \tag{14}$$

$$\frac{\mathrm{d}c(t)}{\mathrm{d}t} = -c(t) + \|\boldsymbol{x}^*\|^2 - \frac{\|\boldsymbol{x}^*\|^2}{\pi}\psi\left(\frac{c(t)}{\|\boldsymbol{x}^*\|\sqrt{b(t)}}\right), \tag{15}$$

with $\psi(x) \overset{\text{def}}{=} 2\arccos(x) - \sin(2\arccos(x))$.

Remark 2: The ODE limit of Markov processes was introduced by Kurtz [14], [15]. In what follows, we provide an informal derivation to highlight the ideas behind the ODE limit, leaving the formal proof of Proposition 2 to a forthcoming paper.

To see how the update formulas in (9) and (10) converge to ODEs in the large systems limit, we consider $k = tN$ for some $t > 0$, and rewrite (9) as

$$\frac{b(t) - b(t - 1/N)}{1/N} = \frac{u_k^2 - v_k^2}{(p_k + z_k)/N} \tag{16}$$

As N tends to infinity, the left-hand side converges to the time-derivative $\mathrm{d}b(t)/\mathrm{d}t$. The expectation of the right-hand side can be computed as

$$\lim_{N\to\infty} \mathbb{E}\frac{u_k^2 - v_k^2}{(p_k + z_k)/N} = \lim_{N\to\infty} \mathbb{E}\left[u_k^2 - v_k^2\right] = \|\boldsymbol{x}^*\|^2 - b(t),$$

where in reaching the second equality we have used the law of large numbers to write $\lim_{N\to\infty}(p_k + z_k)/N = \lim_{N\to\infty}\|\boldsymbol{a}_k\|^2/N = 1$.

Thus, in the large N limit, the update equation (16) becomes the ODE in (14). The expression in (15) can be obtained similarly, after some technical manipulations to compute the expected value of the term $u_k^2 \operatorname{sgn}(u_k)\operatorname{sgn}(v_k) - u_k v_k$ in (10).

Example 2: To demonstrate the accuracy of the ODE limit, we plot in Figure 2 the analytical predictions given by the solutions of the ODEs (black solid lines) together with average values obtained from Monte Carlo simulations (red dots.) In our experiment, we set $N = 1024, M/N = 15$ and $\|\boldsymbol{x}^*\| = 1$. We can see that, even for a moderate size of N, the asymptotic analytical results match with simulation data very well, with the confidence intervals (± one standard deviation) smaller than the size of the dots.

IV. Conclusions

We studied a simple non-convex iterative algorithm for phase retrieval. Based on iterative projections, the algorithm only processes one measure at a time and it can be interpreted as a non-convex version of the Kaczmarz algorithm or an iterative Newton's method. We analyzed the performance of this algorithm in an online setting, with Gaussian measurement vectors. We showed that the dynamics of the algorithm can be fully specified by a 2D Markov process involving two order parameters. Furthermore, in the large systems limit, the algorithm dynamics will converge to a continuous-time function governed by the solutions of two deterministic, coupled, ODEs. By analyzing properties of the underlying ODEs, one can easily establish convergence rates of the algorithm. Results along this line as well as other generalizations will be reported in a forthcoming paper.

References

[1] R. W. Gerchberg, "A practical algorithm for the determination of phase from image and diffraction plane pictures," *Optik*, vol. 35, p. 237, 1972.

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

[3] E. J. Candes, T. Strohmer, and V. Voroninski, "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming," *Communications on Pure and Applied Mathematics*, vol. 66, no. 8, pp. 1241–1274, 2013.

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

[5] K. Jaganathan, S. Oymak, and B. Hassibi, "Sparse phase retrieval: Convex algorithms and limitations," in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 1022–1026.

[6] I. Waldspurger, A. d'Aspremont, and S. Mallat, "Phase recovery, maxcut and complex semidefinite programming," *Mathematical Programming*, vol. 149, no. 1-2, pp. 47–81, 2015.

[7] P. Netrapalli, P. Jain, and S. Sanghavi, "Phase retrieval using alternating minimization," in *Advances in Neural Information Processing Systems*, 2013, pp. 2796–2804.

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

[9] Y. Chen and E. J. Candes, "Solving Random Quadratic Systems of Equations Is Nearly as Easy as Solving Linear Systems," *arXiv preprint arXiv:1505.05114*, 2015.

[10] S. Kaczmarz, "Angenäherte auflösung von systemen linearer gleichungen," *Bull. Internat. Acad. Polon. Sci. Lettres A*, pp. 335–357, 1937.

[11] T. Strohmer and R. Vershynin, "A randomized solver for linear systems with exponential convergence," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 2006, pp. 499–507.

[12] A. Agaskar, C. Wang, and Y. M. Lu, "Randomized Kaczmarz algorithms: Exact MSE analysis and optimal sampling probabilities," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2014.

[13] R. Gordon, R. Bender, and G. T. Herman, "Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography," *Journal of theoretical Biology*, vol. 29, no. 3, p. 471–481, 1970.

[14] T. G. Kurtz, "Solutions of Ordinary Differential Equations as Limits of Pure Jump Markov Processes," *Journal of Applied Probability*, vol. 7, no. 1, p. 49, Apr. 1970.

[15] ——, "Limit Theorems for Sequences of Jump Markov Processes Approximating Ordinary Differential Processes," *Journal of Applied Probability*, vol. 8, no. 2, p. 344, Jun. 1971.