Lecture notes on non-convex algorithms for low-rank matrix recovery

Irène Waldspurger*

Abstract

Low-rank matrix recovery problems are inverse problems which naturally arise in various fields like signal processing, imaging and machine learning. They are non-convex and NP-hard in full generality. It is therefore a delicate problem to design efficient recovery algorithms and to provide rigorous theoretical insights on the behavior of these algorithms. The goal of these notes is to review recent progress in this direction for the class of so-called "non-convex algorithms", with a particular focus on the proof techniques.

Although they aim at presenting very recent research works, these notes have been written with the intent to be, as much as possible, accessible to non-specialists.

These notes were written for an eight-hour lecture at Collège de France. The original version, in French, is available online[1] and the videos of the lecture can be found on the Collège de France website[2].

The beginning takes inspiration from the review articles [Davenport and Romberg, 2016] and [Chen and Chi, 2018].

1 Introduction

In these notes, we consider a family of problems, called *low-rank matrix recovery problems*, which have various applications in data analysis or imaging sciences. The goal is to give an overview of recent theoretical results on a family of algorithms which can be used to solve them, namely *non-convex algorithms*.

The first part of the introduction (Subsection 1.1) defines low-rank matrix recovery problems and presents examples. The second one (Subsection 1.2) explains what non-convex algorithms

*CNRS, Université Paris Dauphine, Inria Mokaplan, France (waldspurger@ceremade.dauphine.fr)

[1] https://www.ceremade.dauphine.fr/~waldspurger/

[2] https://www.college-de-france.fr/site/cours-peccot/p10541769392068432_content.htm

are, which theoretical properties of these algorithms will be of interest for us, and why it is important to understand them. The third one (Subsection 1.3) presents the organization of these notes.

1.1 Low-rank matrix recovery: definition and examples

Let $\mathbb{K}$ be either $\mathbb{R}$ or $\mathbb{C}$, and let n_1, n_2 be positive integers.

A *low-rank matrix recovery problem* informally consists in recovering an unknown matrix $X^s \in \mathbb{K}^{n_1 \times n_2}$ from

- some "simple" information about X^s, modeled by the property "$X^s \in \mathcal{E}$" for a known subset $\mathcal{E}$ of $\mathbb{K}^{n_1 \times n_2}$; this information is often a set of linear measurements over the coefficients of X^s, so that $\mathcal{E}$ is an affine subspace of $\mathbb{K}^{n_1 \times n_2}$;
- the knowledge that X^s has low rank, that is, $\operatorname{rank}(X^s) \ll \min(n_1, n_2)$; sometimes, the rank is exactly known, sometimes not.

It is often solved by looking for the element in $\mathcal{E}$ with minimal rank:

$$\text{minimize } \operatorname{rank}(X) \text{ for } X \in \mathcal{E}. \tag{min-rank}$$

In the following paragraphs, we describe three important examples of low-rank matrix recovery problems.

1.1.1 First example: matrix completion

In matrix completion problems, the "simple" information is the knowledge of some coefficients of X^s, that is $X^s_{i,j}$ for all pairs (i, j) belonging to some set $\Omega \subset \{1, \dots, n_1\} \times \{1, \dots, n_2\}$. This leads to the following minimization problem:

$$\begin{aligned} &\text{minimize } \operatorname{rank}(X), \\ &\quad \text{with } X_{i,j} = X^s_{i,j}, \quad \forall (i,j) \in \Omega. \end{aligned}$$

One of the reasons which made matrix completion popular is that it serves as a modelization of the *Netflix* problem: for any k, l, the coefficient $X^s_{k,l}$ represents the rating user k would give to movie l if s/he watched this movie. The known coefficients $X^s_{i,j}, (i,j) \in \Omega$ are the ratings given by users to movies they indeed watched. The goal is to determine the "not yet given" ratings. The low-rank assumption models the similarities between movies, as well as between users.

1.1.2 Second example: phase retrieval

At first sight, phase retrieval is not a matrix recovery problem: the unknown object is not a matrix, but a vector $x^s \in \mathbb{C}^n$, for some $n \in \mathbb{N}^*$. For some *measurement vectors* $v_1, \dots, v_m \in \mathbb{C}^n$, we have access to

$$|\langle x^s, v_k \rangle|, \quad \text{for all } k \leq m.$$

Here, "$|.|$" denotes the standard complex modulus, and "$\langle ., . \rangle$" the usual Hermitian product. The goal is to recover x^s. Since, for any $\phi \in \mathbb{R}$,

$$|\langle e^{i\phi} x^s, v_k \rangle| = |e^{i\phi}| \, |\langle x^s, v_k \rangle| = |\langle x^s, v_k \rangle|, \quad \forall k = 1, \dots, m,$$

exact identification of x^s is actually never possible. We therefore only try to recover x^s *up to a global phase*.

This problem is called "phase retrieval" because finding the phases of $\langle x^s, v_1 \rangle, \dots, \langle x^s, v_m \rangle$ suffices to solve it (if the phases are known, recovering x^s simply amounts to solving a linear system).

Phase retrieval problems naturally appear in optics, and have for this reason been studied since the 1950s. This is in part due to the fact that electromagnetic waves can be modeled as complex-valude functions, whose modulus is much easier to measure than the phase. Detailed explanations can be found in the review article [Schechtman, Eldar, Cohen, Chapman, Miao, and Segev, 2015].

For some families of measurement vectors $v_1, \dots, v_m$, x^s is not uniquely determined by the modulus $|\langle x^s, v_1 \rangle|, \dots, |\langle x^s, v_m \rangle|$, even up to a global phase: another vector x', different from x^s (and from $e^{i\phi} x^s$ for all $\phi \in \mathbb{R}$) can exist such that

$$|\langle x^s, v_k \rangle| = |\langle x', v_k \rangle|, \quad \forall k = 1, \dots, m.$$

In this case, reconstructing x^s with certainty is impossible. In the following, we assume that the phase retrieval problems we consider do not suffer from this uniqueness issue: the modulus uniquely determine x^s up to a global phase. This property is notably known to hold for "generic"[3] families of measurement vectors when $m \geq 4n - 4$ [Balan, Bodmann, Casazza, and Edidin, 2009].

A phase retrieval problem can be turned into an equivalent low-rank matrix recovery problem with the change of variable $X^s = x^s (x^s)^* = (x_i^s \overline{x_j^s})_{1 \leq i,j \leq n} \in \mathbb{C}^{n \times n}$, called *lifting* [Chai, Moscoso, and Papanicolaou, 2011; Candès, Strohmer, and Voroninski, 2013]. Indeed, for all $x \in \mathbb{C}^n, k \in$

[3] We say that a property holds true for *generic* vectors if it is satisfied by all $(v_1, \dots, v_m)$ in $(\mathbb{C}^n)^m - \Gamma$, for some subset Γ of $(\mathbb{C}^n)^m$ with zero Lebesgue measure.

$\{1, \dots, m\}$,

$$\begin{aligned}
(|\langle x, v_k\rangle| = |\langle x^s, v_k\rangle|) &\iff \left(|\langle x, v_k\rangle|^2 = |\langle x^s, v_k\rangle|^2\right) \\
&\iff \left(\sum_{1\le i,j\le n} \overline{x_i} x_j v_{k,i} \overline{v_{k,j}} = |\langle x^s, v_k\rangle|^2\right) \\
&\iff \left(\langle xx^*, v_k v_k^*\rangle = |\langle x^s, v_k\rangle|^2\right).
\end{aligned}$$

(In the last line, the notation "$\langle ., . \rangle$" refers to the usual scalar product over $\mathcal{H}_n(\mathbb{C})$, the set of Hermitian $n \times n$ matrices: $\langle A, B\rangle = \mathrm{Tr}(A^*B)$, with A^* the conjugate transpose of A.)

Because a matrix $X \in \mathcal{H}_n(\mathbb{C})$ can be written as $X = xx^*$ for some $x \in \mathbb{C}^n$ if and only if

$$X \succeq 0 \quad \text{and} \quad \mathrm{rank}(X) = 1,$$

the following two problems are equivalent:

$$\begin{aligned}
&\text{find } x \in \mathbb{C}^n \\
&\text{such that } |\langle x, v_k\rangle| = |\langle x^s, v_k\rangle|, \quad \forall k \le m;
\end{aligned}$$

$$\begin{aligned}
&\text{find } X \in \mathcal{H}_n(\mathbb{C}) \\
&\text{such that } \langle X, v_k v_k^*\rangle = |\langle x^s, v_k\rangle|^2, \quad \forall k \le m, \\
&\qquad X \succeq 0, \\
&\qquad \mathrm{rank}(X) = 1.
\end{aligned}$$

The second one is a low-rank matrix recovery problem.

1.1.3 Third example: phase synchronization

A phase synchronization problem consists in recovering n complex numbers with unit modulus, $z_1^s, \dots, z_n^s$, from the approximate knowledge of $z_k^s \overline{z_l^s}$, for all $k, l \in \{1, \dots, n\}$, that is from

$$C_{k,l} = z_k^s \overline{z_l^s} + w_{k,l}, \quad \forall k, l \in \{1, \dots, n\},$$

where $w_{k,l}$ is a random noise.

As in phase retrieval, it is never possible to reconstruct the z_k^s more precisely than up to a global phase.

Let us note that reconstruction from the *exact* knowledge of the $z_k^s \overline{z_l^s}$ would be easy: because of the global phase ambiguity, we could assume $z_1^s = 1$ and then, for any $k = 2, \dots, n$,

$$z_k^s \overline{z_1^s} = z_k^s,$$

so we have access to z_k^s. What makes the problem difficult is the noise.

Phase synchronization can be seen as a low-rank matrix recovery problem, for the same reason as phase retrieval. The change of variable $Z^s = z^s(z^s)^*$ indeed allows to rewrite it under the equivalent form

$$\begin{aligned} &\text{find } Z \in \mathcal{H}_n(\mathbb{C}) \\ &\text{such that } Z_{k,l} \approx C_{k,l}, \quad \forall k, l \leq n, \\ &Z \succeq 0, \\ &\text{rank}(Z) = 1. \end{aligned}$$

This problem is motivated by applications like server synchronization in computer networks. It also serves as a simplified model for the rotation synchronization problem, where one must identify n rotations of $\mathbb{R}^3$, $R_1, \ldots, R_n$, from crude estimations of the $R_k R_l^{-1}$. Rotation synchronization is notably important for cryo-electron microscopy; precise references can be found in [Bandeira, Boumal, and Singer, 2017].

1.2 What are these notes about?

When confronted with a specific low-rank matrix recovery problem (as happens for most inverse problem), one has to face the following three questions:

- Identifiability: does the reformulation of the problem under form (min-rank) really allow to recover our matrix of interest X^s? More formally, does Problem (min-rank) have a unique solution, which is precisely X^s?
- Stability: if there are a few errors in the available information (that is, we only have approximate knowledge of the set $\mathcal{E}$), does Problem (min-rank) still allow to recover X^s, or at least a matrix close to X^s?
- Algorithms: which algorithms are able to solve Problem (min-rank) as precisely and fast as possible?

These three questions are interesting and difficult. In these notes, we only focus on the third one. And since it is only useful to numerically solve Problem (min-rank) when it allows to identify X^s, we will implicitely assume that we are only facing low-rank recovery problems for which identifiability holds.

More specifically, we will be interested in algorithms for which rigorous correctness guarantees are available. An ideal *correctness guarantee* is a statement of the form

"For all instances in some class A of low-rank recovery problems, algorithm B outputs the correct matrix X^s."

However, most natural classes of low-rank matrix recovery problems are NP-hard in full generality [Hardt, Meka, Raghavendra, and Weitz, 2014; Fickus, Mixon, Nelson, and Wang, 2014]. Consequently, if we restrict ourselves to algorithms running in a reasonable amount of time, statements of the above form are most of the time too strong to be true. Therefore, we will mostly discuss weaker forms of correctness guarantees:

"If we run it on a random element of some class A of low-rank recovery problems, algorithm B outputs the correct matrix X^s with probability close to 1."

This is the subject of these notes: the goal is to describe algorithms for which a guarantee of this form can be proved, and to explain the associated proof techniques.

Actually, we will restrict ourselves to *non-convex* algorithms with rigorous correctness guarantees. What are non-convex algorithms? Very broadly speaking, the last decades of research in optimization have allowed for the development of efficient algorithms able to solve with arbitrary precision problems of the form "minimize $f(x)$ over all $x \in \mathcal{E}$" when $\mathcal{E}$ is a convex set and $f : \mathcal{E} \to \mathbb{R}$ a convex function. This does not apply to Problem (min-rank) because the rank is not a convex function. Designing an algorithm applicable to Problem (min-rank) requires overcoming the non-convexity issue. Two strategies exist.

- Convex methods: their principle is to approximate the non-convex problem with a convex one, different but with the same minimizer, and solve the convex one. These methods generally work well (in the sense that they correctly recover the solution X^s). Moreover, powerful analysis techniques exist to establish correctness guarantees. Since their introduction around 2010, they have therefore been the subject of intense research. On the negative side, they are often impractical because of their large computational cost.

- Non-convex methods: these algorithms simply ignore the non-convexity of Problem (min-rank), and try to solve it with relatively simple strategies, often coming from the field of convex optimization. Most of them are iterative: starting from an initial estimate of the solution, they progressively refine it with natural heuristics. Much older and more widely-used than convex methods, these algorithms have the advantage to be fast. They can a priori fail if they reach an estimate of the solution which heuristics are not able to refine further although it is not the solution; this is called "reaching a *critical point*". However, in practice, they often work well.

The two strategies have led to the development of useful algorithms and to interesting theoretical analyses. But since it is not possible to cover both in a single eight-hour lecture, the notes focus on the second one only.

Let us stress that the development of rigorous theoretical guarantees for non-convex methods has been the subject of many publications in the last five years. These notes are not an attempt

to exhaustively describe all of them. As the main aim is to explain the most successful proof techniques of the domain, we will pick one or two representative articles for each family of techniques, and describe only these, but with some amount of detail.

1.3 Outline

In Section 2, we give a brief overview of convex methods. These methods have played a crucial role in the development of low rank recovery algorithms with correctness guarantees. As far as I know, they are the first ones for which convincing guarantees were established in relatively general settings, and they have served as examples for the theoretical study of other algorithms. It is therefore good to have an idea of what they are and how they work even if they are not our main subject of interest. In addition, convex and non-convex algorithms are not as disjoint categories as they seem, and it has recently been realized that establishing correctness guarantees for a convex method could help establishing guarantees for a non-convex one or vice-versa [Zhong and Boumal, 2018; Chen, Chi, Fan, Ma, and Yan, 2020].

The short Section 3 defines non-convex methods, and provides an example of such a method in the context of phase retrieval.

In Section 4, we describe a first set of techniques which can prove the correctness of some non-convex low-rank recovery algorithms. As said a few paragraphs ago, the main reason why non-convex algorithms can fail is the presence of *critical points*. But, in some settings, it turns out that critical points do simply not exist, which almost automatically implies that non-convex methods succeed. We provide examples of settings where this property holds, and explain how it can be proved.

In Section 5, we consider the situation where non-convex methods succeed but critical points exist. This situation is much more delicate than the previous one: establishing correctness guarantees then requires to carefully study the trajectory of iterates of the algorithm and to show that it does not come close to one of the critical points. Few technical tools exist for this. The main one is *leave-one-out*; we present it through the example of the *generalized power method* for phase synchronization [Zhong and Boumal, 2018].

Finally, Section 6 presents a different class of results. While Sections 4 and 5 explain how to prove the correctness of specific algorithms, applied to specific low-rank recovery problems, Section 6 presents a family of algorithms which can be applied to a large class of low-rank recovery problems, and explains why some algorithms in this family can be proved to (almost) always succeed. This provides much more general correctness guarantees than in the two preceding sections. The price to pay is that the algorithms for which these guarantees hold are not as satisfying, from a computational point of view, as the algorithms of Sections 4 and 5.

1.4 Notation

For any vector $x \in \mathbb{R}^n$ or $\mathbb{C}^n$, we denote $||x||$ its ℓ^2-norm.

Let $X \in \mathbb{K}^{n_1 \times n_2}$ be a matrix. We denote $\lambda_1(X), \dots, \lambda_{\min(n_1,n_2)}(X)$ its singular values, that is the nonnegative real numbers such that there exist orthogonal matrices $U \in \mathbb{K}^{n_1 \times n_1}, V \in \mathbb{K}^{n_2 \times n_2}$ for which

$$X = U \begin{pmatrix} \lambda_1(X) & & & \\ & \ddots & & \\ & & \lambda_{\min(n_1,n_2)}(X) & \\ 0 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & 0 \end{pmatrix} V.$$

The nuclear norm of X is

$$||X||_* = \sum_{s=1}^{\min(n_1,n_2)} \lambda_s(X)$$

and its operator norm is

$$|||X||| = \max_{s=1,\dots,\min(n_1,n_2)} \lambda_s(X).$$

When $\mathbb{K} = \mathbb{C}$, we denote X^* the conjugate transpose of X.

We denote $\mathcal{S}_n(\mathbb{R})$ the set of real symmetric $n \times n$ matrices, $\mathcal{S}_n^+(\mathbb{R})$ the set of positive semidefinite $n \times n$ matrices, and $\mathcal{H}_n(\mathbb{C})$ the set of Hermitian $n \times n$ complex matrices.

2 Convex methods

2.1 Principle

The principle of convex methods is to approximate the non-convex problem (min-rank) with a convex one. The approximation typically consists in replacing the rank (which is a non-convex function) with the nuclear norm (which is a convex one). This replacement is motivated by the fact that $||.||_*$ is the convex envelope of the rank on $\{X \in \mathbb{K}^{n_1 \times n_2}, |||X||| \leq 1\}$, that is, for all $X \in \mathbb{K}^{n_1 \times n_2}$ such that $|||X||| \leq 1$,

$$||X||_* = \max\{F(X), F : \{X \in \mathbb{K}^{n_1 \times n_2}, |||X||| \leq 1\} \to \mathbb{R} \text{ is convex}, F \leq \text{rank}\}.$$

The problem (min-rank)

$$\text{minimize rank}(X) \text{ for } X \in \mathcal{E}$$

is thus typically replaced with

$$\text{minimize } ||X||_* \text{ for } X \in \mathcal{E}. \qquad \text{(Convex relaxation)}$$

This latter problem is convex if $\mathcal{E}$ is a convex set. It is therefore easier to numerically solve. In addition, although it is a priori only an approximation of Problem (min-rank), it often happens to have the same solution, so that solving the convex problem also solves the non-convex one.

This idea was introduced in [Recht, Fazel, and Parrilo, 2010]. It echoes the theory of *compressed sensing* for the reconstruction of sparse signals.

2.2 Convex methods for matrix completion

Conditions under which the solution of Problem (Convex relaxation) is the same as the one of Problem (min-rank) have been established for all low-rank matrix recovery problems described in the introduction. In the case of matrix completion, for instance, Problem (Convex relaxation) is instanciated as

$$\begin{aligned} &\text{minimize } ||X||_* \\ &\quad \text{with } X_{i,j} = X^s_{ij}, \quad \forall (i,j) \in \Omega. \end{aligned} \tag{1}$$

It has notably been analyzed in [Candès and Recht, 2009; Candès and Tao, 2010; Gross, Krahmer, and Kueng, 2015]. The simplest situation in which one can guarantee that its solution is indeed X^s (the solution of Problem (min-rank)) is described by the following theorem.

Theorem 2.1 (Chen [2015])**.** *Let r be the rank of $X^s \in \mathbb{R}^{n_1 \times n_2}$ and μ_0 its incoherence*[4]*.*

Let us assume that Ω is chosen at random, and contains each of the $n_1 n_2$ pairs $(i,j), i \leq n_1, j \leq n_2$ with probability $p > 0$, independently one from each other.

There exist constants $C, c > 0$ such that, if

$$n_1 n_2 p \geq C \mu_0 r (n_1 + n_2) \log^2(\max(n_1, n_2)),$$

then, with probability at least $1 - \frac{1}{\max(n_1,n_2)^c}$, the solution of Problem (1) is X^s.

In other words, as soon as the number of revealed coefficients of X^s is of order $r(n_1 + n_2)$ (neglecting μ_0 and the logarithmic term), it is possible to recover X^s by solving Problem (1), with high probability. The number of degrees of freedom of a rank r matrix is of order $r(n_1 + n_2)$. The convex algorithm therefore succeeds with (almost) the smallest possible amount of information.

[4] Let $U \in \mathbb{R}^{n_1 \times r}$ (respectively $V \in \mathbb{R}^{n_2 \times r}$) be a matrix whose columns form an orthonormal basis of Range(X^s) (respectively Range(X^{sT})). Let $U_1, \ldots, U_{n_1}, V_1, \ldots, V_{n_2}$ be the lines of U, V. The incoherence μ_0 is defined as $\mu_0 = \max\left(\frac{n_1}{r}||U_1||^2, \ldots, \frac{n_1}{r}||U_{n_1}||^2, \frac{n_2}{r}||V_1||^2, \ldots, \frac{n_2}{r}||V_{n_2}||^2\right)$. Intuitively, it quantifies the maximal alignement between a vector of Range(X^s) (or Range(X^{sT})) and the canonical basis.

2.3 Convex methods for phase retrieval

We recall the "low-rank matrix recovery" formulation of phase retrieval problems:

$$\begin{aligned} &\text{find } X \in \mathcal{H}_n(\mathbb{C}) \\ &\text{such that } \langle X, v_k v_k^* \rangle = |\langle x^s, v_k \rangle|^2, \quad \forall k \leq m, \\ &\qquad X \succeq 0, \\ &\qquad \text{rank}(X) = 1. \end{aligned}$$

Here, as previously, x^s denotes the vector we want to recover, i.e. the true solution of the phase retrieval problem. The above matricial formulation has been obtained from the initial vectorial one through the change of variable $X = xx^*$, hence its solution must be $X^s = x^s x^{s*}$.

To approximate this non-convex problem with a convex one, we follow the method described in Subsection 2.1: we replace the constraint "$\text{rank}(X) = 1$" with the constraint "$||X||_*$ is minimal". For any $X \succeq 0$, it holds $||X||_* = \text{Tr}(X)$, which leads to the following convex problem (called (PhaseLift) in [Candès, Strohmer, and Voroninski, 2013]):

$$\begin{aligned} &\text{minimize } \text{Tr}(X) \\ &\quad \text{with } \langle X, v_k v_k^* \rangle = |\langle x^s, v_k \rangle|^2, \quad \forall k \leq m, \\ &\qquad X \succeq 0. \end{aligned} \tag{PhaseLift}$$

Several works have established correctness guarantees for (PhaseLift) (that is, have proved, in various situations, that the solution of (PhaseLift) was $X^s = x^s x^{s*}$). Most of them apply in the setting where $v_1, \ldots, v_m$ are chosen according to independent normal laws:

$$v_1, \ldots, v_m \overset{iid}{\sim} \mathcal{N}(0, I_n).$$

This assumption is somewhat unrealistic: in most applications, the measurement vectors are not random; even when they are, they seldom follow a normal law. However, this assumption turns out to simplify the computations a lot, and it is difficult to get rid of it.

Under this assumption, the strongest available guarantees have been proved in [Candès and Li, 2014], and are summarized in the following theorem.

Theorem 2.2. *There exist constants $C, c > 0$ such that, for any $n, m \in \mathbb{N}^*$, if $m \geq Cn$, it holds with probability at least $1 - e^{-cm}$ that, whatever $x^s \in \mathbb{C}^n$, the solution $\hat{X}$ of Problem* (PhaseLift) *is*

$$\hat{X} = x^s (x^s)^*.$$

Let us note that, when $m < n$, x^s is never uniquely determined by $|\langle x^s, v_1 \rangle|, \ldots, |\langle x^s, v_m \rangle|$ up to a global phase, hence the phase retrieval problem is intrinsically not solvable. The theorem therefore means that, with high probabiliy, (PhaseLift) solves phase retrieval problems with an optimal (up to a multiplicative constant) number of measurements.

Sketch of proof, adapted from [Chen, Chi, and Goldsmith, 2015]. We define

$$\mathcal{A} : X \in \mathcal{H}_n(\mathbb{C}) \to (\langle X, v_k v_k^* \rangle)_{k=1,\dots,m} \in \mathbb{R}^m;$$
$$\tilde{\mathcal{A}} : X \in \mathcal{H}_n(\mathbb{C}) \to (\mathcal{A}(X)_1 - \mathcal{A}(X)_2, \mathcal{A}(X)_3 - \mathcal{A}(X)_4, \dots) \in \mathbb{R}^{\lfloor m/2 \rfloor}.$$

Here, for any k, $\mathcal{A}(X)_k$ denotes the k-th coordinate of $\mathcal{A}(X)$. Defining $\tilde{\mathcal{A}}$ is necessary to make the proof correct but, in order to grasp the main ideas of the sketch, one can simply imagine that $\tilde{\mathcal{A}} = \mathcal{A}$.

Proposition 2.3. *There exist $c, C, d, D > 0$ such that, for any $r \in \{1, \dots, n\}$, the following property is true: if $m \geq Cnr$, then, with probability at leat $1 - e^{-cm}$,*

$$d||X||_F \leq \frac{1}{m}||\tilde{\mathcal{A}}(X)||_{\ell^1} \leq D||X||_F. \tag{RIP}$$

for all matrices $X \in \mathcal{H}_n(\mathbb{C})$ with rank at most r.

(Here, $||X||_F$ is the Frobenius norm of X: $||X||_F = \left(\sum_{1 \leq i,j \leq n} |X_{i,j}|^2\right)^{1/2}$.)

When (RIP) holds, we say that $\tilde{\mathcal{A}}$ satisfies a *Restricted Isometry Property* in ℓ^1 / Frobenius norms (notion which echoes the theory of compressed sensing again). When it holds, this property implies that $X = x^s(x^s)^*$ is the unique solution of Problem (PhaseLift). Indeed, if we denote $\hat{X}$ the solution of the problem, we must have

$$\mathrm{Tr}(\hat{X}) \leq \mathrm{Tr}(x^s(x^s)^*),$$
$$\left\langle \hat{X}, v_k v_k^* \right\rangle = |\langle x^s, v_k \rangle|^2 = \langle x^s(x^s)^*, v_k v_k^* \rangle, \quad \forall k = 1, \dots, m,$$
$$\hat{X} \succeq 0,$$

which implies, for $H \overset{def}{=} \hat{X} - x^s(x^s)^*$:

$$\mathrm{Tr}(H) \leq 0, \tag{2a}$$
$$\tilde{\mathcal{A}}(H) = 0, \tag{2b}$$
$$x^s(x^s)^* + H \succeq 0. \tag{2c}$$

The following proposition concludes.

Proposition 2.4. *If Property (RIP) holds for some $r > \frac{4D^2}{d^2} + 2$, then, whatever $x^s \in \mathbb{C}^n$, there does not exist $H \in \mathcal{H}_n(\mathbb{C})$ a non-zero matrix for which Properties (2a), (2b) and (2c) are true.*

A very vague expanation of this proposition is that, if H is such that $\mathrm{Tr}(H) \leq 0$ and $x^s(x^s)^* + H \succeq 0$, then H is "somewhat close to $-\lambda x^s(x^s)^*$" for some $\lambda > 0$. As a consequence, even if the rank of H may be larger than r, Property (RIP) allows to show that

$$\frac{1}{m}||\tilde{\mathcal{A}}(H)||_{\ell^1} \geq d'||H||_F$$

for some constant $d' > 0$. The equality $\tilde{\mathcal{A}}(H) = 0$ therefore implies that $H = 0$. □

Many extensions of this result exist: it is possible to analyze the stability of (PhaseLift) to noise, to consider other random distributions for the measurement vectors than a normal law, wonder how to exploit additional knowledge we might have on x^s ... A non-exhaustive list of references is [Candès, Li, and Soltanolkotabi, 2015; Gross, Krahmer, and Kueng, 2015; Li and Voroninski, 2013].

Other convex phase retrieval methods can also be designed. In particular, it turns out that the requirement, in (PhaseLift), that the trace of X is minimal, is not stricly necessary. Surprisingly, the following problem

$$\begin{aligned} &\text{find } X \in \mathcal{H}_n(\mathbb{C}) \\ &\text{such that } \langle X, v_k v_k^* \rangle = |\langle x^s, v_k \rangle|^2, \quad \forall k \leq m, \\ &\quad X \succeq 0 \end{aligned} \tag{Weak-PhaseLift}$$

satisfies almost the same correctness guarantees as the ones stated in Theorem 2.2 [Demanet and Hand, 2012]. With a suitable change of variable, Problem (Weak-PhaseLift) can be transformed into another problem, called (PhaseCut),

$$\begin{aligned} &\text{minimize } \mathrm{Tr}(MU) \\ &\quad \text{with } U \in \mathcal{H}_m(\mathbb{C}), \\ &\qquad U_{k,k} = 1, \quad \forall k \leq m, \\ &\qquad U \succeq 0, \end{aligned} \tag{PhaseCut}$$

where M is an explicit matrix which depends on the v_k and $|\langle x^s, v_k \rangle|$. The (PhaseCut) formulation has some advantages over the (PhaseLift) one, from an algorithmic point of view. It also seems to be a bit more robust to noise in some settings [Waldspurger, d'Aspremont, and Mallat, 2015].

2.4 Limits of convex methods

To summarize, convex methods allow solving low-rank matrix recovery problems with an essentialy minimal number of measurements. Their algorithmic side is relatively well understood; many numerical solvers have been developed for problems of the form (Convex relaxation).

This idyllic picture must however be toned down: convex methods suffer from an intrinsic "dimensionality issue", which severely limits the computational efficiency of numerical solvers, and hence the practical applicability of these methods.

A matrix $X^s \in \mathbb{K}^{n_1 \times n_2}$, with rank r, can be written as

$$X^s = UV^T$$

for some $U \in \mathbb{K}^{n_1 \times r}, V \in \mathbb{K}^{n_2 \times r}$. When the rank r of the solution is known, the number of "degrees of freedom" of a low-rank recovery problem is therefore of order $(n_1 + n_2)r$[5]. But to solve the convex problem (Convex relaxation), one must reconstruct each of the $n_1 n_2$ coefficients of X^s. Thus, the complexity of numerical solvers tends to be polynomial in $n_1 n_2$, which is much larger than what we could have hoped for:

$$n_1 n_2 \gg (n_1 + n_2)r.$$

Let us illustrate this phenomenon with two examples of algorithms.

1. Interior-point methods: these algorithms apply to convex problems of the form

$$\begin{aligned} &\text{minimize } \mathrm{Tr}(X) \\ &\text{with } \mathcal{A}(X) = 0, \\ &X \succeq 0, \end{aligned}$$

   where $\mathcal{A} : \mathbb{K}^{n \times n} \to \mathbb{R}^m$ is an affine map. When $\mathcal{E}$ is affine, Problem (Convex relaxation) can be rewritten under this form for $n = n_1 + n_2$, with the introduction of properly chosen additional variables.

   Interior-point methods are iterative. The number of iterations necessary to reach a good approximation of the solution is in general moderate but, in full generality, the number of arithmetic operations performed at each iteration [Borchers and Young, 2007, Page 357] is of order

$$(m + n)mn^2.$$

2. FISTA [Beck and Teboulle, 2009]: this algorithm approximately solves problems of the form (Convex relaxation), also in the case where $\mathcal{E}$ is affine, by replacing them with a "regularized" version:

$$\underset{X \in \mathbb{K}^{n_1 \times n_2}}{\text{minimize}} \frac{1}{2}||\mathcal{A}(X)||^2 + \lambda||X||_*,$$

   for $\mathcal{A} : \mathbb{K}^{n_1 \times n_2} \to \mathbb{R}^m$ an affine map such that $\mathcal{E} = \{X, \mathcal{A}(X) = 0\}$, and $\lambda > 0$ a parameter.

[5] slightly smaller actually, since the matrices U, V in the decomposition $X^s = UV^T$ are not unique

This algorithm is also iterative. It requires more iterations than an interior-point method. Each iteration consists in one step of gradient descent on the map $X \to ||\mathcal{A}(X)||^2$ and one application of the so-called *proximal operator* of the nuclear norm. The proximal operator is the most costly part. Heuristics exist to accelerate its computation by exploiting the structure of the problem but, in the most general case, it requires to perform a singular value decomposition, which represents $O(n_1^3)$ operations if n_1 and n_2 are of the same order.

It is clear that algorithms with these complexities are unapplicable when n_1, n_2, m grow large.

To overcome this dimensionality issue, an option is to improve numerical solvers further, by taking advantage of the fact that convex approximations of low-rank recovery problems are not "generic" problems of the form (Convex relaxation) but (in principle) problems of the form (Convex relaxation) *with a low-rank solution* [Ding, Yurtsever, Cevher, Tropp, and Udell, 2019; Yurtsever, Tropp, Fercoq, Udell, and Cevher, 2019]

Another option is simply to come back to non-convex methods, either to the traditional heuristics mentioned in Subsection 1.2 or to more modern ones. Precisely understanding when non-convex methods succeed and when they fail is still out of reach. But in the last years, a flurry of works has at least allowed to find some situations where non-convex methods succeed and it is moreover possible to rigorously explain it.

3 Non-convex methods: definition and example

3.1 Informal definition of non-convex methods

A rank r matrix $X \in \mathbb{K}^{n_1 \times n_2}$ can always be written under the form

$$X = UV^T, \quad \text{for some } U \in \mathbb{K}^{n_1 \times r}, V \in \mathbb{K}^{n_2 \times r}.$$

When X is symmetric (respectively Hermitian if $\mathbb{K} = \mathbb{C}$), it can actually be written as $X = UU^T$ (respectively $X = UU^*$) for some $U \in \mathbb{K}^{n_1 \times n_1}$. Reconstructing the factors U and V suffices to recover X.

Remark 3.1. *In the case of low-rank matrix recovery problems coming from phase retrieval or phase synchronization, this property is obvious, since the low-rank matrix X^s (or Z^s in the case of phase synchronization) to recover is precisely defined as $X^s = x^s(x^s)^*$ (respectively $Z^s = z^s(z^s)^*$), with $x^s \in \mathbb{C}^{n \times 1}$ (respectively $z^s \in \mathbb{C}^{n \times 1}$) the solution of the original problem.*

Non-convex algorithms are generally based on the following scheme:

1. Choice of an initial point (U_0, V_0) (or simply U_0 in the symmetric/hermitian case), which can be an estimation of the solution if available or an arbitrary point.

2. Iterative application of some simple heuristics aiming at making (U_0, V_0) closer and closer to a solution (that is, a pair (U, V) such that $X = UV^T$ is the sought-after low-rank matrix X^s). This yields a sequence of iterates $(U_t, V_t)_{t\in\mathbb{N}}$ which should ideally converge.

3. Output of $U_T V_T^T$ for some $T \in \mathbb{N}$ large enough.

These methods are widely-used in practice. As we have said, they can fail, but it is empirically observed that they work well in various situations.

3.2 Example: alternating projections for phase retrieval

In this subsection, we present an example of non-convex method: the phase retrieval algorithm called *alternating projections*. It is the oldest and most well-known phase retrieval algorithm, sometimes named *error reduction* or *Gerchberg-Saxton* (from the name of the researchers who introduced it [Gerchberg and Saxton, 1972]).

We consider a general phase retrieval problem, in "vector" form: we aim at reconstructing $x^s \in \mathbb{C}^n$ from

$$b_k \overset{def}{=} |\langle x^s, v_k\rangle|, \quad \forall k = 1, \dots, m.$$

We define $\mathcal{A} : x \in \mathbb{C}^n \to \mathcal{A}(x) = (\langle x, v_1\rangle, \dots, \langle x, v_m\rangle) \in \mathbb{C}^m$. If $m \geq n$, $\mathcal{A}$ is injective for generic measurement vectors. Under this assumption, recovering x^s (up to a global phase) is equivalent to recovering $y^s \overset{def}{=} \mathcal{A}(x^s)$ (also up to a global phase). Our phase retrieval problem can thus be reformulated as

$$\begin{aligned} &\text{find } y \in \mathbb{C}^m, \\ &\text{such that } y \in \text{Range}(\mathcal{A}), \\ &\qquad |y_k| = b_k, \quad \forall k \leq m. \end{aligned}$$

If we introduce the notation $E = \{y \in \mathbb{C}^m, |y_k| = b_k, \forall k \leq m\}$, this problem can be concisely rewritten as

$$\begin{aligned} &\text{find } y \in \mathbb{C}^m, \\ &\text{such that } y \in \text{Range}(\mathcal{A}) \cap E. \end{aligned}$$

The alternating projections algorithm is a natural heuristic inspired by this formulation.

Definition 3.2. *Let $T \in \mathbb{N}$ be fixed. The alternating projections method is composed of the following steps.*

1. *We choose an arbitrary starting point (for our numerical experiments, we will choose $y_0 \sim \mathcal{N}(0, y_m)$).*

2. *For every* $t = 1, \dots, T$, *we define*

$$y_t = P_{\mathrm{Im}(\mathcal{A})}(P_E(y_{t-1})),$$

where, for any non-empty closed subset of $\mathbb{C}^m$, P_S *denotes the projection onto* S[6].

3. *We return* y_T.

Remark 3.3. *Although the "vectorial" definition we just gave does not make it fully apparent, the alternating projections algorithm can be seen as an instance of the scheme described in Subsection 3.1, up to the change of variable*

$$U_t = \mathcal{A}^{-1}(y_t), \quad \forall t \geq 1.$$

Here, $\mathcal{A}^{-1}$ *denotes the inverse of* $\mathcal{A}$, *when seen as an operator from* $\mathbb{C}^n$ *to* $\mathrm{Range}(\mathcal{A})$.

We illustrate the behavior of this method with a numerical experiment in the setting where $v_1, \dots, v_m$ are independently chosen according to a normal distribution:

$$v_1, \dots, v_m \overset{iid}{\sim} \mathcal{N}(0, I_m).$$

Figure 1[7] displays the performance curve of the alternating projections method (that is, the probability of exact recovery), as a function of m/n, for $n = 40$. From this figure, it is tempting to conjecture that, when measurement vectors are normally distributed, alternating projections succeed with high probability as soon as $m \geq Cn$, which would be the same correctness guarantee as the one we have seen for (PhaseLift) (Theorem 2.2). Can we prove it?

We will not answer this question: the alternating projections method is difficult to analyze and the question is still open. However, in the next sections, we will explain how to establish similar guarantees for some other non-convex low-rank recovery algorithms.

4 When there are no bad critical points

The main obstacle to the convergence of non-convex methods is the possible existence of *bad critical points*, at which the heuristic used in the method can get stuck. But, for some heuristics, it is possible to show that such points do not exist, which implies that the algorithms succeed. The goal of this section is to present this proof technique through examples.

[6] that is to say a (possibly non uniquely defined) function $P_S : \mathbb{C}^m \to S$ such that, for any y, $||P_S(y) - y|| = \min_{z \in S} ||z - y||$

[7] Figures have been generated with the code available at `https://www.ceremade.dauphine.fr/~waldspurger/code/non_convex_lecture_notes_figures.zip`, except the red curve on Figure 1, generated with *PhasePack* [Chandra, Zhong, Hontz, McCulloch, Studer, and Goldstein, 2017].

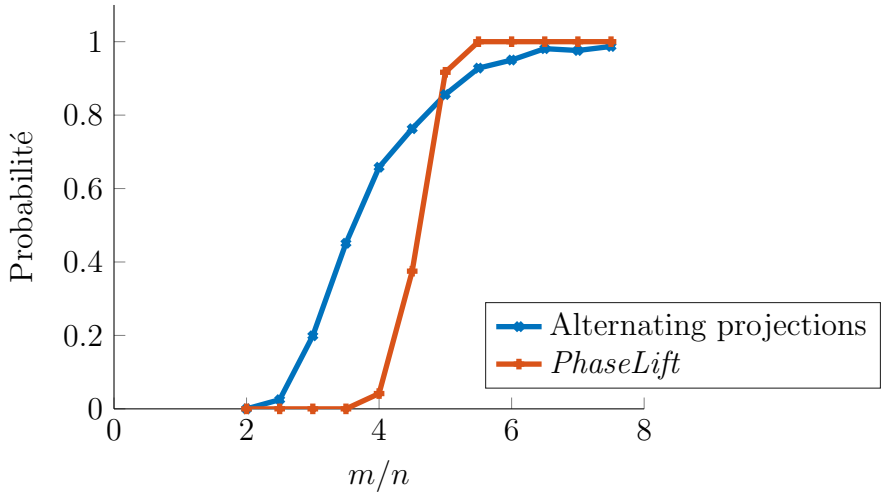


Figure 1: Success probability for two phase retrieval algorithms, as a function of m/n, for $n = 40$.

Let us first underline that this technique cannot be applied to all non-convex algorithms: most classical heuristics possess bad critical points. It numerically seems, for instance, that the alternating projections method we just discussed possesses bad critical points even when m/n is very large. This does not prevent alternating projections from working well, but makes them out of reach of the proof technique of this section. Actually, among the algorithms to which this technique has been successfully applied, most are not traditional methods used by practioners for decades, but algorithms explicitely designed in order to make them amenable to the proof technique.

The technique can be applied in two slightly different ways:

- One can

  1. first prove that the heuristic used in the algorithm has no bad critical point *in some neighborhood of the solution*;
  2. then show that all iterates belong to this neighborhood (which is only possible if the algorithm uses a careful initialization strategy: if the initial point is arbitrary, it has no reason to belong to the neighborhood).

  This strategy is described in Subsection 4.1.

- One can show that the heuristic has no bad critical point at all. This strategy is described in Subsection 4.2.

We describe the strategies through examples of phase retrieval algorithms, under the assumption that measurement vectors are normally distributed. However, they apply to many other problems and algorithms. References will be given at the end of Subsections 4.1 and 4.2.

4.1 No bad critical point close to the solution

We illustrate the first method with the study of *Wirtinger Flow*, a phase retrieval algorithm introduced in [Candès, Li, and Soltanolkotabi, 2015], which consists of the following steps:

1. Initialization: choice of x_0 according to a so-called *spectral method*, which we will describe later.

2. Refinement step: let us define

$$f : x \in \mathbb{C}^n \to \frac{1}{2m} \sum_{k=1}^{m} (|\langle x, v_k \rangle|^2 - b_k^2)^2.$$

   It is a $\mathcal{C}^\infty$ non-convex function, whose minima are exactly the solutions of the phase retrieval problem[8].
   For all $t \in \mathbb{N}$, we define x_{t+1} by applying to x_t a gradient descent step over f:

$$x_{t+1} = x_t - \mu \nabla f(x_t),$$

   for some constant $\mu > 0$.

3. Output of x_T for some T large enough.

This algorithm obeys the correctness guarantees stated in the following theorem[9].

Theorem 4.1 ([Candès, Li, and Soltanolkotabi, 2015, Thm 3.3])**.** *Let the solution x^s of the phase retrieval problem be arbitrary. Let the measurement vectors $v_1, \dots, v_m$ be generated according to independent normal distributions.*

There exist constants $C, c > 0$ such that, if

$$Cn \log(n) \leq m$$

and if $\mu \leq \frac{c}{n}$, then, with probability $1 - O\left(\frac{1}{n^2}\right)$,

$$\text{dist}(x^s, x_t) \leq \frac{1}{8} \left(1 - \frac{\mu}{4}\right)^{t/2} ||x^s||$$

for all $t \in \mathbb{N}$. In particular, $x_t \overset{t \to +\infty}{\to} x^s$.

[8] For all x, $f(x) \geq 0$ and equality is reached if and only if $|\langle x, v_k \rangle| = b_k$ for all k.

[9] In this theorem, the notation "dist" is defined as $\text{dist}(x, y) = \min_{\phi \in \mathbb{R}} ||x - e^{i\phi} y||$. In the rest of the subsection, we will do as if $\text{dist}(x, y) = ||x - y||$, to simplify the explanations, but this is not rigorous.

As said at the beginning of the section, the proof of this theorem proceeds in two steps, whose principles will be described in Paragraphs 4.1.1 and 4.1.2.

1. We show that the heuristic used in the refinement step has no bad critical point in the ball $B(x^s, ||x^s||/8)$. More precisely, we establish that, for any $x \in B(x^s, ||x^s||/8)$,

$$\text{dist}(x^s, x - \mu\nabla f(x)) \leq \rho\, \text{dist}(x^s, x) \tag{3}$$

   for some $\rho = \sqrt{1 - \frac{\mu}{4}} \in]0; 1[$.

2. We analyze the spectral initialization method and show that

$$x_0 \in B(x^s, ||x^s||/8). \tag{4}$$

These two steps suffice to establish the theorem: starting from Property (4) and iteratively applying Equation (3) immediately proves the main inequality of Theorem 4.1.

4.1.1 First step

We assume to simplify that $||x^s|| = 1$.

In this paragraph, we explain how to prove Property (3), but for some $\rho \in]0; 1[$ larger than $\sqrt{1 - \frac{\mu}{4}}$. This modification degrades the convergence rate guaranteed by Theorem 4.1 but allows for a slightly simpler proof.

It is enough to prove the following properties:

$$\forall x \in B(x^s, 1/8), \quad \text{Re}(\langle x - x^s, \nabla f(x)\rangle) \geq \alpha \text{dist}(x, x^s)^2, \tag{5}$$

$$\forall x \in B(x^s, 1/8), \quad ||\nabla f(x)|| \leq \beta \text{dist}(x, x^s), \tag{6}$$

for $\alpha, \beta > 0$ some well-chosen values. Indeed, if these inequalities are true, we have for all $x \in B(x^s, 1/8)$ that

$$\begin{aligned} ||x^s - (x - \mu\nabla f(x))||^2 &= ||x^s - x||^2 - 2\mu \text{Re}(\langle x - x^s, \nabla f(x)\rangle) + \mu^2 ||\nabla f(x)||^2 \\ &\leq ||x^s - x||^2 - 2\mu\alpha ||x - x^s||^2 + \mu^2\beta^2 ||x - x^s||^2 \\ &\leq (1 - \mu\alpha)||x^s - x||^2 \quad \text{if } \mu < \frac{\alpha}{\beta^2}. \end{aligned}$$

The general principle for proving Properties (5) and (6) is to compute the explicit expression of ∇f and use it to write $\text{Re}(\langle x - x^s, \nabla f(x)\rangle)$ and $||\nabla f(x)||^2$ as a sum of realizations of independent

random variables, which can be analyzed with the help of a classical statistical tool: concentration inequalities[10].

Let us focus on Property (5). Let us consider some x of the form $x = x^s + h$ with $||h|| < 1/8$. We have

$$\begin{aligned}\nabla f(x) &= \frac{1}{m}\sum_{r=1}^{m}(|\langle v_r, x\rangle|^2 - b_r^2)v_r v_r^* x \\ &= \frac{1}{m}\sum_{r=1}^{m}\left(2\mathrm{Re}(\overline{\langle v_r, x^s\rangle}\langle v_r, h\rangle + |\langle v_r, h\rangle|^2)\right)\langle v_r, x^s + h\rangle v_r,\end{aligned}$$

which implies that

$$\begin{aligned}&\mathrm{Re}(\langle x - x^s, \nabla f(x)\rangle) \\ &= \frac{1}{m}\sum_{r=1}^{m}\left(2\mathrm{Re}^2(\overline{\langle v_r, x^s\rangle}\langle v_r, h\rangle) + 3\mathrm{Re}(\overline{\langle v_r, x^s\rangle}\langle v_r, h\rangle)|\langle v_r, h\rangle|^2 + |\langle v_r, h\rangle|^4\right) \\ &\stackrel{\text{def}}{=} \frac{1}{m}\sum_{r=1}^{m} Y_r(h).\end{aligned}$$

For any fixed h, the random variables $Y_1(h), \ldots, Y_r(h)$ are independent and identically distributed. One can check that, for all r, if we assume (to simplify) that $\langle x^s, h\rangle$ belongs to $\mathbb{R}$,

$$\begin{aligned}\mathbb{E}(Y_r(h)) &= 3\langle x^s, h\rangle^2 + ||h||^2 + 6\langle x^s, h\rangle ||h||^2 + 2||h||^4 \\ &\geq \frac{||h||^2}{2} \text{ if } ||h|| \leq \frac{1}{8}.\end{aligned}$$

The above-mentioned concentration inequalities allow to show that

$$\frac{1}{m}\sum_{r=1}^{m} Y_r(h) \geq \mathbb{E}\left(\frac{1}{m}\sum_{r=1}^{m} Y_r(h)\right) - \frac{||h||^2}{4}$$

[10]Here is a brief definition of concentration inequalities, for readers who are not familiar with them. In their most basic version, concentration inequalities aim at studying the behavior of a sum of independent random variables

$$Y_1 + Y_2 + \cdots + Y_K.$$

Under reasonable assumptions, the sum is close to its expectation with high probability when K is large enough. Concentration inequalities allow to precisely control this closeness, by providing upper bounds for

$$\mathrm{Prob}(Y_1 + \cdots + Y_K \geq \mathbb{E}(Y_1 + \cdots + Y_K) + \epsilon)$$

for all $\epsilon > 0$. The exact form of the upper bound depends on the hypotheses available for the Y_k.

with probability at least $1 - e^{-\gamma m}$ for some constant $\gamma > 0$. From this we deduce

$$\mathrm{Re}(\langle x - x^s, \nabla f(x) \rangle) \geq \frac{||h||^2}{4} = \frac{||x - x^s||^2}{4},$$

which is the inequality in Property (5), with $\alpha = \frac{1}{4}$.

This proof only shows that, for some fixed $x \in B(x^s, 1/8)$, the inequality of Property (5) holds with high probability. It does not show that the inequality holds with high probability for all $x \in B(x^s, 1/8)$ at the same time. However, the proof can be extended to all elements $x \in B(x^s, 1/8)$ using a very classical probabilistic argument called *ϵ-net*.

4.1.2 Second step

Let us describe the spectral initialization method. As far as I know, this method was first proposed for phase retrieval in [Netrapalli, Jain, and Sanghavi, 2013], before being used in [Candès, Li, and Soltanolkotabi, 2015]. A similar idea could already be found in [Keshavan, Montanari, and Oh, 2010], but applied to matrix completion problems.

Let us define the matrix

$$M = \frac{1}{m} \sum_{r=1}^{m} b_r^2 v_r v_r^* = \frac{1}{m} \sum_{r=1}^{m} |\langle x^s, v_r \rangle|^2 v_r v_r^* \in \mathbb{C}^{n \times n}.$$

Informally, in this definition, for each r, the rank-1 matrix $v_r v_r^*$ appears with a weight proportional to $|\langle x^s, v_r \rangle|^2$. Consequently, the more it is aligned with $x^s (x^s)^*$, the more it contributes to M, so that M is "biased" in the direction of $x^s (x^s)^*$. This reasoning justifies using as an initial point for *Wirtinger Flow*

$$x_0 = \text{main eigenvector of } M.$$

The following lemma establishes precision guarantees for the spectral initialization method.

Lemma 4.2. *There exists a constant $C > 0$ such that, when $m \geq Cn \log(n)$,*

$$\mathrm{dist}(x_0, x^s) \leq \frac{||x^s||}{8}$$

with probability $1 - O\left(\frac{1}{n^2}\right)$.

The proof of the lemma relies on the following property, valid with probability $1 - O\left(\frac{1}{n^2}\right)$:

$$|||M - \mathbb{E}(M)||| \leq \delta,$$

where δ is a constant which can be arbitrarily small if the constant C in the lemma is large enough. This property is proved with the help of concentration inequalities and allows to establish the lemma thanks to the equality

$$\mathbb{E}(M) = I_n + x^s (x^s)^*.$$

4.1.3 Related work

Many other algorithms than *Wirtinger Flow* have been designed, which rely on the same two-step scheme "spectral initialization + refinement" and are amenable to a similar theoretical analysis.

Several phase retrieval algorithms notably use this principle, replacing the basic spectral initialization method of *Wirtinger Flow* with a more sophisticated one [Chen and Candès, 2017; Mondelli and Montanari, 2019] and using another refinement heuristic like "truncated" gradient descent [Chen and Candès, 2017], gradient descent on a different cost function than the Wirtinger one (possibly non-smooth, which raises some technical difficulties) [Zhang and Liang, 2016; Wang, Giannakis, and Eldar, 2018] or alternating projections [Waldspurger, 2018]. These algorithms have correction guarantees similar to the ones described in Theorem 4.1, actually a bit better

For other low-rank matrix recovery problems, we can cite for instance [Jain, Netrapalli, and Sanghavi, 2013] for matrix completion and *RIP matrix sensing* (matrix recovery from linear measurements when the measurement operator satisfies a restricted isometry property), [Zhao, Wang, and Liu, 2015] for RIP matrix sensing, with a wider range of refinement heuristics than the previous article, [Zheng and Lafferty, 2016] for matrix completion again and [Chen and Wainwright, 2015] for more general results, applicable to several matrix recovery problems.

4.2 No bad critical point at all

The second proof technique we describe is a variant of the previous one. It consists in showing that the refinement heuristic has no critical point at all (instead of having no critical point *in a neighborhood of the solution*). From a technical point of view, it is in general more involved than the first one: it necessitates an even finer analysis of the functions which come into play. On the positive side, it can apply to conceptually simpler algorithms, closer to practice: with this technique, no sophisticated initialization technique is necessary.

The algorithm we use to illustrate this technique has been proposed in [Sun, Qu, and Wright, 2018]. It is identical to *Wirtinger Flow*, except for the initialization:

1. Initialization: random choice of x_0; we may for instance pick $x_0 \in B(0,1)$ with uniform probability.

2. Refinement step: gradient descent[11] on the function

$$f : x \in \mathbb{C}^n \quad \to \quad \frac{1}{2m} \sum_{k=1}^{m} (|\langle x, v_k \rangle|^2 - b_k^2)^2.$$

[11] Sun, Qu, and Wright [2018] actually propose trying to minimize f with another local optimization method than gradient descent, called *Trust-Region*. As the convergence theorem we are going to state holds for both gradient descent and Trust-Region, we use gradient descent.

3. Output of x_T for some T large enough.

Here are the correction guarantees.

Theorem 4.3 ([Sun, Qu, and Wright, 2018])**.** *Let the solution x^s of the phase retrieval problem be arbitrary. Let the measurement vectors $v_1, \ldots, v_m$ be generated according to independent normal distributions.*

There exists a constant $C > 0$ such that, if

$$Cn \log^3(n) \leq m,$$

then, with probability $1 - O\left(\frac{1}{n}\right)$, the sequence of iterates $(x_t)_{t\in\mathbb{N}}$ obtained at the refinement step of the algorithm satisfies

$$\operatorname{dist}(x^s, x_t) \overset{t\to+\infty}{\to} 0,$$

provided that the gradient descent step is small enough.

4.2.1 What are the properties of critical points?

In order to analyze the algorithm, we first need to understand the properties of critical points. Then we can show that there exists no point with these properties other than the solution x^s; in particular, there is no bad critical point.

Let us recall that we call *critical points* the points at which the refinement heuristic (here, gradient descent over f) can stagnate. From the definition of gradient descent, a critical point x_* necessarily satisfies

$$\nabla f(x_*) = 0.$$

This is called a *first-order optimality condition.*

In addition, if we assume that x_0 does not belong to some set of "problematic" initial points which has zero Lebesgue measure, one can show (but it is much more difficult) that critical points must also verify a *second-order optimality condition*:

$$\nabla^2 f(x_*) \succeq 0.$$

These properties, which are not specific to our objective function f, are rigorously stated in the following theorem.

Theorem 4.4. *Let $\mathcal{L} : \mathbb{R}^n$ (or $\mathbb{C}^n$) $\to \mathbb{R}$ be analytic. We assume that*

$$\mathcal{L}(x) \to +\infty \quad \text{when } ||x|| \to +\infty.$$

We run gradient descent over $\mathcal{L}$ with constant stepsize $\mu > 0$. This yields a sequence of iterates $(x_t)_{t\in\mathbb{N}}$. If μ is small enough, then, for all $x_0 \in B(0, 1)$, the sequence $(x_t)_{t\in\mathbb{N}}$ is convergent. Moreover,

- *for all* x_0, $\nabla \mathcal{L}(\lim_{t\to+\infty} x_t) = 0$*;*
- *for almost all* x_0, $\nabla^2 \mathcal{L}(\lim_{t\to+\infty} x_t) \succeq 0$.

The first part of this theorem can be deduced from [Absil, Mahony, and Andrews, 2005, Thm 4.1] and the second one from [Panageas and Piliouras, 2017, Thm 3], which is a generalization of [Lee, Simchowitz, Jordan, and Recht, 2016, Cor 9]. We have stated this theorem for gradient descent with constant stepsize, but it holds for various other local optimization algorithms.

For the algorithm of [Sun, Qu, and Wright, 2018], it implies that:

- the sequence of iterates $(x_t)_{t\in\mathbb{N}}$ converges to some limite $x_* \in \mathbb{C}^n$;
- with probability 1, x_* satisfies

$$\nabla f(x_*) = 0; \qquad \text{(First-order-optimality)}$$
$$\nabla^2 f(x_*) \succeq 0.. \qquad \text{(Second-order-optimality)}$$

To prove that the algorithm succeeds (Theorem 4.3), it thus suffices to show that, except for the solution x^s, no point satisfies Equations (First-order-optimality) and (Second-order-optimality). In the next paragraph, we explain how this can be shown.

4.2.2 Idea of proof

To gain some intuition of the proof, let us first determine which points satisfy Equations (First-order-optimality) and (Second-order-optimality) when f is replaced with its expectation (which simplifies the computation a log). For any fixed $x \in \mathbb{C}^n$, we can check that the expectation of $f(x)$ over $v_1, \dots, v_m$ is

$$\mathbb{E}(f(x)) = ||x||^4 - ||x||^2||x^s||^2 - |\langle x, x^s\rangle|^2 + ||x^s||^4.$$

Consequently,

$$\nabla(\mathbb{E}f)(x) = 2((2||x||^2 - ||x^s||^2)x - \langle x^s, x\rangle x^s).$$

With this formula, we can explicitely compute which x satisfy Equation (First-order-optimality) (that is, $\nabla(\mathbb{E}f)(x) = 0$). They are the elements of the following three sets:

- $E_1 = \{e^{i\theta}x^s, \theta \in \mathbb{R}\}$, which is the set of solutions of the phase retrieval problem;
- $E_2 = \{0\}$;
- $E_3 = \left\{x \in \mathbb{C}^n, \langle x^s, x\rangle = 0, ||x|| = \frac{||x^s||}{\sqrt{2}}\right\}$.

Among these points, which ones satisfy Equation (Second-order-optimality)? Elements of E_1 do, since they are global minimizers of $\mathbb{E}(f)$. For elements of E_2, E_3, we need the explicit expression of $\nabla^2(\mathbb{E}f)$: for any $x, h \in \mathbb{C}^n$,

$$\nabla^2(\mathbb{E}f)(x) \cdot (h,h) = 2\left((2||x||^2 - ||x^s||^2)||h||^2 + 4\mathrm{Re}^2(\langle x, h\rangle) - |\langle x^s, h\rangle|^2\right).$$

From this expression, one easily checks that $x = 0$ does not satisfy the second-order optimality condition (actually, $\nabla^2(\mathbb{E}f)(0) \prec 0$). And for any $x \in E_3$, we observe that

$$\nabla^2(\mathbb{E}f)(x) \cdot (x^s, x^s) = -2||x^s||^4 < 0. \tag{7}$$

As a consequence, Equation (Second-order-optimality) cannot hold.

We have thus shown that, replacing f with $\mathbb{E}f$, the only points which satisfy Equations (First-order-optimality) and (Second-order-optimality) are the solutions of the phase retrieval problem. In order to extend this result from $\mathbb{E}f$ to f itself, a first idea would be to show that, with high probability, for all $x \in \mathbb{C}^n$,

$$||\nabla f(x) - \nabla \mathbb{E}f(x)|| \text{ is very small;} \tag{8a}$$
$$|||\nabla^2 f(x) - \nabla^2 \mathbb{E}f(x)||| \text{ is very small,} \tag{8b}$$

for some notion of "smallness" which would require a proper definition. We could then try to deduce the properties for f from the computations we have done for $\mathbb{E}f$.

This precise approach does not work: whatever way we define "smallness", properties (8a) and (8b) are not true for all $x \in \mathbb{C}^n$ at the same time, with high probability. However, one can still show that ∇f and $\nabla^2 f$ share some properties with their expectations. For instance, with arguments similar to Subsection 4.1, one can establish the following proposition.

Proposition 4.5. *We define, for some explicit constant $\alpha > 0$ which is not explicitely given here,*

$$Z_1 = \{x \in \mathbb{C}^n, |\langle x, x^s\rangle| \leq \alpha||x^s||^2, ||x|| \leq (1-\alpha)||x^s||\}.$$

If $m \geq Cn\log^3(n)$ for some constant $C > 0$ large enough, it holds with probability $1 - O\left(\frac{1}{m}\right)$ that

$$\nabla^2 f(x) \cdot (x^s, x^s) \leq -\alpha||x^s||^4$$

for all $x \in Z_1$.

This property, analogous to Equation (7), for f instead of $\mathbb{E}f$, implies that f has no point in Z_1 satisfies Equation (Second-order-optimality). We can define other sets $Z_2, Z_3, \ldots$ whose union is equal to $\mathbb{C}^n - E_1$ (recall that E_1 is the set of solutions) and establish over $Z_2, Z_3, \ldots$ similar properties as the one stated for Z_1 in Proposition 4.5. This proves that f has no point outside E_1 satisfies Equations (First-order-optimality) and (Second-order-optimality), and concludes the proof.

4.2.3 Related work

Let us emphasize that this proof technique cannot be applied to all algorithms: the non-existence of critical points is a strong property, which many refinement heuristics do not satisfy. In phase retrieval, as far as I know, the algorithm of [Sun, Qu, and Wright, 2018] which we have studied is the only one for which the property is known to hold. For alternating projections, for instance, numerical experiments suggest that bad critical points almost always exist[12]. This does not prevent alternating projections from working well with high probability, for normally distributed measurement vectors, but it makes our proof technique useless.

Outside phase retrieval, this technique has notably been used in [Ge, Lee, and Ma, 2016] for matrix completion, in [Bhojanapalli, Neyshabur, and Srebro, 2016] for RIP matrix sensing, in [Sun, Qu, and Wright, 2017] for dictionary learning and in [Kawaguchi, 2016] for linear neural networks. Many other examples can be found in the review article [Zhang, Qu, and Wright, 2020]. Although they have a common proof structure, these results are quite specific to the considered problem and algorithm. Some generalization attempts have been done, notably in [Li and Tang, 2017] and [Ge, Jin, and Zheng, 2017], but they are still rudimentary.

5 When there are critical points (leave-one-out)

In the previous section, we have seen that the success of some non-convex algorithms could be explained by the non-existence of critical points, which removes the stagnation risk for the refinement. We have seen that this phenomenon allows to prove correctness guarantees for various low-rank matrix recovery methods. However, we have also underlined that this proof strategy does not apply to all algorithms, because many succeed *despite the presence of critical points*.

An illustration is the alternating projections method for phase retrieval. Figure 2 depicts the attraction basins of the various critical points of this algorithm, in a setting where $n = 20, m = 400$ and the measurement vectors are realizations of independent normal distributions with real coordinates. More precisely, the figure represents a two-dimensional submanifold of $\mathbb{R}^n$, projected onto a square. To each critical point is associated a color; all points in the figure which belong to the attraction basin of this critical point are colored with the corresponding color. The solution of the phase retrieval problem is attributed the color black. Therefore, the black set in Figure 2 contains all points starting from which the alternating projections algorithm converges towards the correct solution.

From this figure, we can see that many bad critical points exist. However, the total volume of their attraction basins is tiny: when randomly initialized, alternating projections succeed in finding the correct solution with high probability.

[12] except if $m \geq O(n^2)$ but this regime is not very interesting

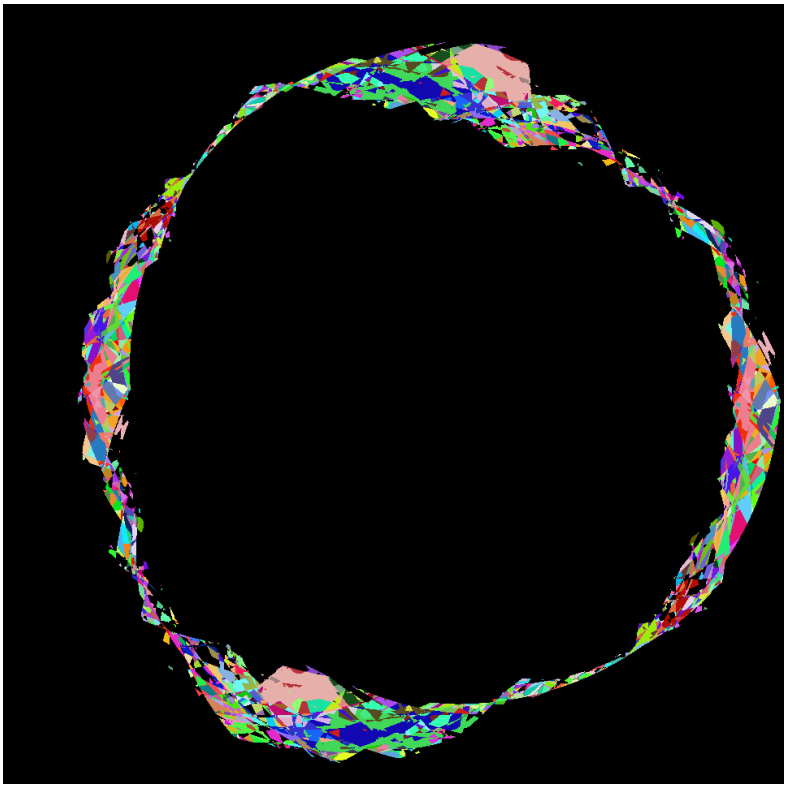


Figure 2: Attraction basins of alternating projections for a phase retrieval problem with $n = 20, m = 400$

Unfortunately, estimating the size of attraction basins for a given algorithm is in general difficult. If we denote T the operator applied by the refinement heuristic at each iteration, estimating the size of the basins requires to understand the behavior of $(T^n(x_0))_{n\in\mathbb{N}}$ as a function of x_0. Exploiting the randomness of T and its independence with x_0, it is often possible to understand the properties of $T(x_0)$. But understanding $T^2(x_0) = T(T(x_0))$ (and a fortiori $T^k(x_0)$ for general k) is much more difficult: T and $T(x_0)$ are both random variables, but they are correlated and the relation between them is complex.

A proof technique called *leave-one-out* has recently been proposed to overcome this issue. In this section, we present this proof technique through the example of a phase synchronization algorithm, following the article [Zhong and Boumal, 2018]. At the end, we give a preliminary overview of its possible extensions and limitations, although the leave-one-out technique is too recent so that these are well-understood.

5.1 Prologue: leave-one-out for machine learning

This subsection is relatively independent from the rest of the section and can harmlessly be skipped.

[Zhong and Boumal, 2018] is, as far as I know, the first article in which leave-one-out was used to study a non-convex optimization algorithm. It had however been used in other fields before. In particular, in statistical problems where parameters of a random law must be estimated from samples, it allows to understand some subtle properties of possible estimators [El Karoui, Bean, Bickel, Lim, and Yu, 2013]. It relies on a relatively simple idea, used for a long time in machine learning, which we briefly describe.

In a classical machine learning problem, the goal is to predict, when given a data point $x \in \mathbb{R}^n$, some property of x, modeled by a real number $f_*(x)$. Data are generated according to an unknown probability law $\mathbb{P}$. To learn how to perform the prediction, one is given m "training" data points $x_1, \dots, x_m$ (independently generated according to $\mathbb{P}$) and their associated properties $f_*(x_1), \dots, f_*(x_m)$.

Assuming we have designed an algorithm Alg, which takes as input $x_1, \dots, x_m$, $f_*(x_1), \dots, f_*(x_m)$ and outputs a prediction function $f_{\text{Alg}} : \mathbb{R}^n \to \mathbb{R}$, how do we measure its quality? Ideally, we want f_{Alg} to precisely mimic f_* at all data points x which could be generated by the probability law $\mathbb{P}$. This motivates the definition of the *generalization error* of f_{Alg}:

$$\text{Err}(f_{\text{Alg}}) = \mathbb{E}_{x\sim\mathbb{P}}(\ell(f_{\text{Alg}}(x), f_*(x))),$$

for some *loss function* $\ell : \mathbb{R}^2 \to \mathbb{R}^+$, well-suited to the problem at hand. Unfortunately, exactly computing the generalization error requires the knowledge of $\mathbb{P}$ and f_*. How can we compute an approximate value of this error from only the knowledge of $x_1, \dots, x_m, f_*(x_1), \dots, f_*(x_m)$?

One possibility is to compute, for all $i = 1, \dots, m$, the fonction $f_{\mathrm{Alg},-i}$ output by Alg when Alg is only given as input$x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_m$ and $f_*(x_1), \dots, f_*(x_{i-1}), f_*(x_{i+1}), \dots, f_*(x_m)$. For any i, $f_{\mathrm{Alg},-i}$ should not be very different from f_{Alg}, but it is independent from $(x_i, f_*(x_i))$. One can therefore say that

$$\ell(f_{\mathrm{Alg},-i}(x_i), f_*(x_i))$$

has almost the same probability distribution as $\ell(f_{\mathrm{Alg}}(x), f_*(x))$ for $x \sim \mathbb{P}$. This leads to the following approximation, called *leave-one-out* approximation:

$$\mathrm{Err}(f_{\mathrm{Alg}}) \approx \frac{1}{m} \sum_{i=1}^{m} \ell(f_{\mathrm{Alg},-i}(x_i), f_*(x_i)).$$

If the algorithm satisfies some "stability" assumptions, this approximation can be rigorously justified [Elisseeff and Pontil, 2003].

5.2 Definition of the generalized power method

In this section, we describe and motivate the phase synchronization algorithm which we will study in the rest of the section.

Let us recall what we have said in Paragraph 1.1.3: a phase synchronization problem consists in identifying (up to a global phase) n unitary complex numbers $z_1^s, \dots, z_n^s$ from

$$C_{k,l} = z_k^s \overline{z_l^s} + w_{k,l}, \quad \forall k, l \in \{1, \dots, n\},$$

where, for any k, l, $w_{k,l}$ is an unknown noise.

We assume to simplify that the $w_{k,l}$ are independent realizations of a complex normal law with variance σ^2. More precisely, we assume

$$\begin{aligned} &w_{k,l} \sim \mathcal{N}_{\mathbb{C}}(0, \sigma^2) \text{ independently, for all } k < l; \\ &w_{k,l} = \overline{w_{l,k}} \text{ for all } k > l; \\ &w_{k,k} = 0 \text{ for all } k. \end{aligned}$$

From the noisy observations $(C_{k,l})_{1 \le k,l \le n}$ only, it is impossible to exactly recover $(z_k^s)_{1 \le k \le n}$. Instead, we redefine our goal to finding unitary complex numbers $(z_k^{obj})_{1 \le k \le n}$ which are minimizers of the following problem:

$$\min_{|z_1|=\dots=|z_n|=1} \sum_{k,l} |C_{k,l} - z_k \overline{z_l}|^2 \tag{9}$$

(which corresponds to defining $(z_k^{obj})_{1 \le k \le n}$ as the so-called *maximal likelihood* estimator of $(z_k^s)_{1 \le k \le n}$).

Let us note that Definition (9) is equivalent to z^{obj} being a solution of the following maximization problem:

$$z^{obj*} C z^{obj} = \max_{|z_1|=\cdots=|z_n|=1} z^* C z, \tag{10}$$

which is similar to the definition of the main eigenvector of C:

$$z^{ppal*} C z^{ppal} = \max_{||z||=1} z^* C z. \tag{11}$$

The main eigenvector can be computed with the *power method*: starting from a random $z^{(0)}$, one iteratively defines $z^{(t+1)}$ as the projection of $Cz^{(t)}$ onto the unit sphere, that is

$$z^{(t+1)} = \frac{Cz^{(t)}}{||Cz^{(t)}||}, \quad \forall t \in \mathbb{N}.$$

The analogy between Problems (10) and (11) suggests the following phase synchronization algorithm, similar to the power method.

1. We set $z^{(0)} = z^{ppal}$.
2. For all $t \geq 0$, we set
$$z^{(t+1)} = \mathcal{P}(Cz^{(t)}),$$
where $\mathcal{P}$ is the projection onto $\{z \in \mathbb{C}^n, |z_1| = \cdots = |z_n| = 1\}$, that is to say, for all z,
$$\mathcal{P}(z)_k = \frac{z_k}{|z_k|}, \quad \forall k = 1, \ldots, n.$$
(We use the convention $\frac{0}{0} = 1$.)

This algorithm has been introduced in [Boumal, 2016] and is named *generalized power method*.

The best known correctness guarantees for this algorithm are stated in the following theorem, which comes from [Zhong and Boumal, 2018].

Theorem 5.1. *There exist constants $\delta > 0, \rho \in]0;1[$ such that, if $\sigma \leq \delta\sqrt{\frac{n}{\log(n)}}$, then, with probability $1 - O\left(\frac{1}{n^2}\right)$, the sequence of iterates output by the generalized power method satisfies*

$$\frac{\text{dist}(z^{(t)}, z^{obj})}{||z^{obj}||} \leq \rho^t, \quad \forall t \geq 0,$$

where dist *is defined by* $\text{dist}(u, v) = \inf_{\alpha \in \mathbb{R}} ||u - e^{i\alpha} v||$.

[Gao and Zhang, 2020] shows that, when $\sigma \geq \sqrt{n}$, z^{obj} is not a significantly better approximation of z^s than a random guess: for some constant $c > 0$,

$$\mathbb{E}\left(\text{dist}(z^{obj}, z^s)^2\right) \geq cn,$$

while a random vector $\hat{z}$ chosen with uniform probability in $\{z, |z_1| = \cdots = |z_n| = 1\}$ satisfies $\mathbb{E}\left(\text{dist}(\hat{z}, z^s)^2\right) \leq 2n$. This means that, when $\sigma \geq \sqrt{n}$, our phase synchronization problem is rather uninteresting. Consequently, the condition $\sigma \leq \delta\sqrt{\frac{n}{\log(n)}}$ in Theorem 5.1 is not very restrictive.

5.3 Proof of Theorem 5.1

5.3.1 Difficulties

The goal of this paragraph is to explain why proving Theorem 5.1 is difficult and requires a different strategy from Section 4. Pointing out the main difficulties will also help us understand the intuition which led Zhong and Boumal to a correct proof, presented in Paragraph 5.3.2. Towards this goal, let us try to apply the strategy of Section 4 (that is to show that the generalized power method has no critical point) and see why it fails.

We start with a basic but necessary property (whose proof is in Appendix A).

Proposition 5.2. *The vector z^{obj} is a fixed point of the operator $(z \to \mathcal{P}(Cz))$.*

To follow the strategy of Section 4, let us try to prove that $(z \to \mathcal{P}(Cz))$ has no bad critical point in some neighborhood of z^s. As neighborhood, let us take the simplest choice $B(z^s, \epsilon||z^s||)$, for some $\epsilon > 0$ which can be much smaller than 1 but must not go to 0 when n goes to infinity. We could try to establish the following properties:

1. The operator $(z \to \mathcal{P}(Cz))$ is ρ-Lipschitz on $B(z^s, \epsilon||z^s||)$ for some $\rho < 1$: for all $y, y' \in B(z^s, \epsilon||z^s||)$,
$$\text{dist}(\mathcal{P}(Cy), \mathcal{P}(Cy')) \leq \rho \text{dist}(y, y'). \tag{12}$$
2. Vectors z^{obj}, z^0 belong to $B\left(z^s, \frac{\epsilon}{3}||z^s||\right)$.

Combined with Proposition 5.2, these two properties would imply that, for all t,

$$\text{dist}(z^{(t)}, z^{obj}) \leq \rho^t \text{dist}(z^{(0)}, z^{obj}),$$

and thus prove the result.

The second property turns out to be true. Unfortunately, the first one is false with high probability. Indeed, $\mathcal{P}$ is discontinuous at any point $z \in \mathbb{C}^n$ with at least one zero coordinate. If

we want Equation (12) to be true for all $y, y' \in B(z^s, \epsilon||z^s||)$, it is therefore necessary that, for any $y \in B(z^s, \epsilon||z^s||)$, no coordinate of Cy is equal to zero. Can this happen?

Let us first consider a vector $y = z^s + u$, for some random vector u chosen in $B(0, \epsilon||z^s||)$ with uniform probability, independently from the noise $(w_{k,l})_{1\leq k,l\leq n}$. We denote W the matrix whose (k,l)-th coefficient is $w_{k,l}$. Let us recall that it is a Hermitian matrix and that its off-diagonal coefficients are realizations of centered Gaussian variables with variance σ^2. The matrix C is

$$C = z^s z^{s*} + W.$$

For all k, we thus have

$$\begin{aligned}(C(z^s+u))_k &= z^s_k \left(||z^s||^2 + \langle z^s, u\rangle\right) + \langle W_{:,k}, z^s + u\rangle \\ &= z^s_k \left(||z^s||^2 + O(\epsilon||z^s||^2)\right) + O(\sigma||z^s+u||) \\ &= nz^s_k + O(\epsilon n) + O(\sigma\sqrt{n}). \end{aligned} \tag{13}$$

In these equations, $W_{:,k}$ is the k-th column of W. The second equality is due to the fact that $||u|| \leq \epsilon||z^s||$ on the one hand and, on the other hand, to the fact that, since u and W are independent, $\langle W_{:,k}, z^s + u\rangle$ is, conditioning on u, a centered Gaussian variable with variance $||z^s + u||^2\sigma^2$; its values are therefore of order $\sigma||z^s + u||$. The last equality is true because $||z^s|| = \sqrt{n}$.

As $|z^s_k| = 1$ and $\sigma \ll \sqrt{n}$, Equation (13) yields

$$|C(z^s + u)|_k = n(1 + O(\epsilon)),$$

which is far away from 0. This informal reasoning suggests that, if we pick a vector $y \in B(z^s, \epsilon||z^s||)$ at random with uniform probability, independently from W, then Cy has no coordinate equal or close to zero, with high probability.

Nevertheless, if we consider $y = z^s - \eta\frac{W_{:,k}}{||W_{:,k}||}$ (for some complex number η such that $|\eta| < \epsilon||z^s||$ whose value will be given later),

$$\begin{aligned}(Cy)_k &= z^s_k \left(||z^s||^2 - \eta\left\langle z^s, \frac{W_{:,k}}{||W_{:,k}||}\right\rangle\right) + \langle W_{:,k}, z^s\rangle - \eta||W_{:,k}|| \\ &= z^s_k n(1 + O(\epsilon)) - \eta||W_{:,k}|| \\ &= z^s_k n(1 + O(\epsilon)) - \eta\sigma\sqrt{n}(1 + o(1)).\end{aligned}$$

We choose $\eta = \frac{\sqrt{n}(1+O(\epsilon))}{\sigma(1+o(1))}z^s_k$. This leads to

$$(Cy)_k = 0.$$

As $||z^s|| = \sqrt{n}$, this definition of η satisfies the constraint $|\eta| < \epsilon||z^s||$ as soon as $\sigma > \frac{2}{\epsilon}$, so y does belong to $B(z^s, \epsilon||z^s||)$ and one of the coordinates of Cy is zero. And since the theorem must be proved for σ of order up to $\sqrt{\frac{n}{\log(n)}}$, we cannot assume $\sigma \leq \frac{2}{\epsilon}$.

To summarize, Property (12) is plausible for vectors y, y' chosen at random in $B(z^s, \epsilon||z^s||)$ independently from W, but false when y or y' is unnaturally correlated with a column of W.

5.3.2 Proof

Hiding some technical aspects under the carpet, we can say that the proof of Theorem 5.1 decomposes in the following four steps:

1. For some $\rho \in]0;1[$, we show that the map $(z \to \mathcal{P}(Cz))$ is ρ-Lipschitz over
$$\mathcal{N} \stackrel{def}{=} \{z \in B(z^s, \epsilon||z^s||) \text{ such that } \forall k, |\langle W_{:,k}, z\rangle| < \kappa n\}. \tag{14}$$
(In this definition, $\epsilon, \kappa > 0$ are well-chosen constants.)

2. We show that, with high probability, $z^{(0)}$ belongs to
$$\mathcal{N}' \stackrel{def}{=} \left\{z \in B\left(z^s, \frac{\epsilon}{3}||z^s||\right) \text{ such that } \forall k, |\langle W_{:,k}, z\rangle| < \frac{\kappa n}{2}\right\} \subset \mathcal{N}.$$

3. We show that, with high probability, $z^{(t)}$ belongs to $\mathcal{N}'$ for all $t \in \mathbb{N}^*$.

4. We conclude: from the previous three steps, $(z^{(t)})_{t\in\mathbb{N}}$ is a Cauchy sequence. It is therefore convergent, and its limit belongs to $\mathcal{N}$. To establish the theorem, we only have to prove that the limit is z^{obj}.

The first of these steps is the least difficult one: $\mathcal{N}$ is precisely defined as (more or less) the largest set over which the classical probabilistic arguments informally used in Paragraph 5.3.1 allow to show that $(z \to \mathcal{P}(Cz))$ is ρ-Lipschitz for some $\rho \in]0;1[$. The first step therefore only requires to make rigorous the arguments of Paragraph 5.3.1. The fourth step relies on a relation between the definition of z^{obj} and some semidefinite optimization problem. We will not describe it in more detail. The second and third steps are the ones for which the leave-one-out technique is necessary. As the main objective of this section is to present the leave-one-out principle and not to provide a complete proof of the theorem, we assume that the property of the second step is true and focus on proving the property of the third step.

We proceed iteratively. Assuming that $z^{(0)}, \ldots, z^{(t-1)}$ belong to $\mathcal{N}'$, we must show that $z^{(t)}$ belongs to $\mathcal{N}'$. We can relatively easily show that
$$z^{(t)} \in B\left(z^s, \frac{\epsilon}{3}||z^s||\right).$$

The key difficulty is to show that, for all $k \leq n$,

$$|\left\langle W_{:,k}, z^{(t)} \right\rangle| < \frac{\kappa n}{2}. \tag{15}$$

Following the reasoning of Paragraph 5.3.1, it would be easy if, for any k, $z^{(t)}$ and $W_{:,k}$ were independent random variables. However, since the construction of $z^{(t)}$ involves the matrix C, $z^{(t)}$ depends on C, and therefore on W.

To overcome this issue, we introduce, for each $k \leq n$, an auxiliary sequence $(z^{(k,t)})_{t\in\mathbb{N}}$, whose definition is exactly the same as $(z^{(t)})_{t\in\mathbb{N}}$, except that we replace the matrix

$$C = z^s z^{s*} + W$$

with the matrix

$$C^{(k)} = z^s z^{s*} + W^{(k)},$$

where $W^{(k)}$ is the same matrix as W, with the coefficients in the k-th row and column replaced by zeroes. Let us underline that these auxiliary sequences are theoretical tools only: they cannot be computed in practice. Indeed, in a real phase synchronization instance, computing an auxiliary sequence $(z^{(k,t)})_{t\in\mathbb{N}}$ would require computing $W^{(k)}$ and thus exactly knowing W (which is never the case, otherwise the problem would be trivial).

For all k, the sequence $(z^{(k,t)})_{k\in\mathbb{N}}$ is independent from $W_{:,k}$, since the coefficients in $W_{:,k}$ appear nowhere in the definition of the sequence. As a consequence, it holds for all t, with high probability, that

$$|\left\langle W_{:,k}, z^{(k,t)} \right\rangle| = \tilde{O}(\sigma\sqrt{n}) < \frac{\kappa n}{4}. \tag{16}$$

(Compared to O, the notation $\tilde{O}$ hides an additional $\sqrt{\log(n)}$ factor.)

To deduce Property (15) from Property (16), it suffices to show that, for any k, sequences $(z^{(t)})_{t\in\mathbb{N}}$ and $(z^{(k,t)})_{t\in\mathbb{N}}$ are close with high probability. More precisely, we prove by iteration over t that the following properties are true with high probability[13]:

1. $\mathrm{dist}(z^{(t)}, z^{(k,t)}) \leq \frac{1}{60}$ for all k;
2. $|\left\langle W_{:,k}, z^{(t)} \right\rangle| < \frac{\kappa n}{2}$ for all k.

Let us give an idea of the proof of these properties. For all k,

$$\begin{aligned}\mathrm{dist}(z^{(t)}, z^{(k,t)}) &= \mathrm{dist}(\mathcal{P}(Cz^{(t-1)}), \mathcal{P}(C^{(k)}z^{(k,t-1)})) \\ &\leq \mathrm{dist}(\mathcal{P}(Cz^{(t-1)}), \mathcal{P}(Cz^{(k,t-1)})) + \mathrm{dist}(\mathcal{P}(Cz^{(k,t-1)}), \mathcal{P}(C^{(k)}z^{(k,t-1)})).\end{aligned} \tag{17}$$

[13]For technical reasons, Zhong and Boumal proceed by iteration only up to $t \approx 3n^2$. For larger values of t, they use a different, more elementary, argument, which we do not describe here.

Using our induction hypotheses,

$$z^{(t-1)} \in \mathcal{N}' \quad \text{and} \quad \text{dist}(z^{(k,t-1)}, z^{(t-1)}) \leq \frac{1}{60},$$

from which we deduce that $z^{(t-1)}, z^{(k,t-1)}$ belong to $\mathcal{N}$ and therefore that

$$\text{dist}(\mathcal{P}(Cz^{(t-1)}), \mathcal{P}(Cz^{(k,t-1)})) \leq \rho \text{dist}(z^{(t-1)}, z^{(k,t-1)}) \leq \frac{\rho}{60}. \tag{18}$$

On the other hand, with high probability

$$\begin{aligned}
\text{dist}(\mathcal{P}(Cz^{(k,t-1)}), \mathcal{P}(C^{(k)}z^{(k,t-1)})) &\overset{(*)}{\leq} \frac{2}{n}\text{dist}(Cz^{(k,t-1)}, C^{(k)}z^{(k,t-1)}) \\
&\leq \frac{2}{n}||(C - C^{(k)})z^{(k,t-1)}|| \\
&\overset{(\square)}{\leq} \frac{2}{n}\left(|\left\langle W_{:,k}, z^{(k,t-1)}\right\rangle| + ||W_{:,k}||\,|z_k^{(k,t-1)}|\right) \\
&\overset{(\circ)}{=} \tilde{O}\left(\frac{\sigma}{\sqrt{n}}\right).
\end{aligned} \tag{19}$$

Inequality $(*)$ is true because the coordinates of $Cz^{(k,t-1)}$ and $C^{(k)}z^{(k,t-1)}$ are all larger than $\frac{n}{2}$ in modulus (which can be derived from the fact that $z^{(t-1)}$ and $z^{(k,t-1)}$ belong to $\mathcal{N}$). Inequality $(\square)$ is true because all rows and columns of $C - C^{(k)}$ are zero, except the k-th column (which is $W_{:,k}$) and the k-th row (which is $W^*_{:,k}$). For inequality $(\circ)$, we have used the independence between $W_{:,k}$ and $z^{(k,t-1)}$ to upper bound $|\left\langle W_{:,k}, z^{(k,t-1)}\right\rangle|$.

Combining inequalities (17), (18) and (19) yields

$$\text{dist}(z^{(t)}, z^{(k,t)}) \leq \frac{\rho}{60} + \tilde{O}\left(\frac{\sigma}{\sqrt{n}}\right) \leq \frac{1}{60}.$$

This proves the Property 1. Property 2 is easier: for all k,

$$\begin{aligned}
|\left\langle W_{:,k}, z^{(t)}\right\rangle| &\leq |\left\langle W_{:,k}, z^{(k,t)}\right\rangle| + ||W_{:,k}||\text{dist}(z^{(t)}, z^{(k,t)}) \\
&= \tilde{O}(\sigma\sqrt{n}) + \tilde{O}(\sigma\sqrt{n}) \times \frac{1}{60} \\
&< \frac{\kappa n}{2}.
\end{aligned}$$

5.4 Extensions and limits

This subsection briefly describes how leave-one-out can be applied to other non-convex algorithms than the generalized power method and when its application fails. The reader must keep in mind

that, as said before, the introduction of leave-one-out in the field of non-convex optimization is recent. Therefore, the applicability and limits of this method have not been well explored yet, and the content of this subsection must be considered as preliminary ideas.

5.4.1 General principle

To understand how leave-one-out can be applied to other algorithms, we briefly summarize its principle in a general context. We consider a non-convex algorithm, whose sequence of iterates $(z^{(t)})_{t\in\mathbb{N}}$ is defined by

$$z^{(t+1)} = T(z^{(t)})$$

for some map $T : \mathbb{R}^n \to \mathbb{R}^n$. We want to show that it converges to the "correct solution" z^{obj}.

The principle of leave-one-out is as follows.

1. We define an appropriate set, of the form

   $$\mathcal{N} = \{z \in \Omega \text{ such that } F_1(z) < 1, \dots, F_K(z) < 1\},$$

   for some functions $F_1, \dots, F_K : \mathbb{R}^n \to \mathbb{R}$ and Ω a (deterministic) open set. In the case of the generalized power method, this corresponds to Definition (14), with $F_k : z \to \frac{1}{\kappa n} |\langle W_{:,k}, z\rangle|$ for $k = 1, \dots, n$ and $\Omega = B(z^s, \epsilon||z^s||)$.
   We show that

   - T is contractive over $\mathcal{N}$ (or some other property which implies that $z^{(t)} \overset{t\to+\infty}{\longrightarrow} z^{obj}$ if $z^{(t)}$ belongs to $\mathcal{N}$ for all t);
   - $F_1, \dots, F_K$ are Lipschitz in a neighborhood of any point of $\mathcal{N}$.

2. We show that $z^{(0)} \in \mathcal{N}$.

3. We introduce auxiliary iterates $(z^{(k,t)})_{1\le k\le K, t\in\mathbb{N}}$, obeying a similar definition as $(z^{(t)})_{t\in\mathbb{N}}$ except that T is replaced by variants T_k: for any k, t,

   $$z^{(k,t+1)} = T_k(z^{(k,t)}).$$

   The definition of these auxiliary sequences must be chosen in such a way that, for any k, t, it is not difficult to upper bound

   $$F_k(z^{(k,t)}).$$

4. We prove by iteration over $t \in \mathbb{N}$ that, for well-chosen real numbers $(\epsilon_{k,t})_{1\le k\le K}$,

   $$\forall k, \quad ||z^{(t)} - z^{(k,t)}|| \le \epsilon_{k,t};$$
   $$z^{(t)} \in \mathcal{N}.$$

For the first of these properties, the principle of the proof is broadly the following sequence of inequalities:

$$\begin{aligned}||z^{(t)} - z^{(k,t)}|| &= ||T(z^{(t-1)}) - T_k(z^{(k,t-1)})|| \\ &\leq ||T(z^{(t-1)}) - T(z^{(k,t-1)})|| + ||T(z^{(k,t-1)}) - T_k(z^{(k,t-1)})|| \\ &\leq \rho_T \epsilon_{k,t-1} + ||(T - T_k)(z^{(k,t-1)})||,\end{aligned}$$

where ρ_T is the Lipschitz constant of T over $\mathcal{N}$. As T_k is a variant of T, one can hope to be able to upper bound $||(T - T_k)(z^{(k,t-1)})||$ by something "small".
For the second property, the difficulty is to show that $F_k(z^{(t)}) < 1$ for all k. We have

$$\begin{aligned}F_k(z^{(t)}) &\leq F_k(z^{(k,t)}) + \left|F_k(z^{(t)}) - F_k(z^{(k,t)})\right| \\ &\leq F_k(z^{(k,t)}) + \rho_{F_k} \epsilon_{k,t},\end{aligned}$$

where ρ_{F_k} is the Lipschitz constant of F_k in the neighborhood of $z^{(t)}$. From the well-chosen definition of $(z^{(k,t)})_{t\in\mathbb{N}}$, we have a good upper bound for $F_k(z^{(k,t)})$ at our disposal, which allows to conclude.

This is the general principle applied, for instance, in [Ma, Wang, Chi, and Chen, 2018; Ding and Chen, 2020; Chen, Chi, Fan, and Ma, 2019; Chen, Chi, Fan, Ma, and Yan, 2020]. It can be further refined. For instance, in [Chen, Chi, Fan, and Ma, 2019], the map T is not exactly contractive, which makes it difficult to upper bound $||T(z^{(t-1)}) - T(z^{(k,t-1)})||$; the authors achieve this through a second leave-one-out argument, nested in the first one.

[Zhang, 2020] can also be seen as a (quite extreme) instance of this principle, applied to alternating projections for phase retrieval in a setting where the number of measurement vectors is much larger than the dimension of the unknown signal. In this article, T_k is not a variant of T: it is a random map, with the same distribution as T but independent from it.

5.4.2 Limits

A first drawback of this method is that it requires a precise understanding of the local Lipschitz properties of T, possibly also of dT. This makes the proof rather technical, and very specific to one problem and one algorithm.

In my opinion, a second drawback is that it strongly relies on the fact that T is continuous, and even contractive, on a relatively large set $\mathcal{N}$. This is necessary if want to guarantee that sequences $(z^{(t)})_{t\in\mathbb{N}}$ and $(z^{(k,t)})_{t\in\mathbb{N}}$ stay close to each other. But some algorithms do not satisfy this contraction property.

An example is (again) the alternating projections method for phase retrieval, applied to signals and measurement vectors with real coordinates[14], which follow independent normal laws.

[14] We emphasize the realness. It seems that the behaviour is quite different when coordinates are complex.

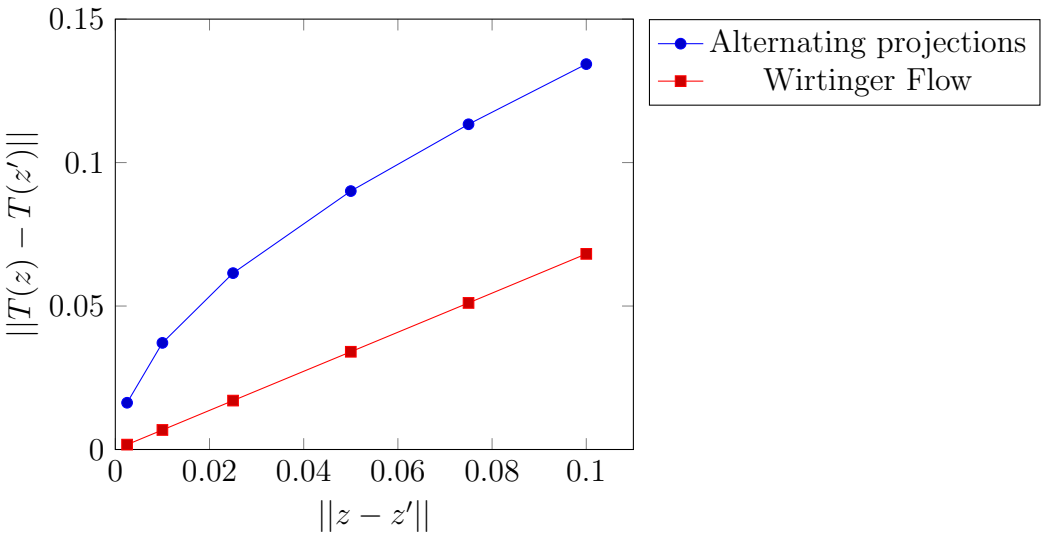


Figure 3: $\mathbb{E}||T(z) - T(z')||$ as a function of $||z - z'||$ for two phase retrieval algorithms, *Wirtinger Flow* and alternating projections, for the reconstruction of signals with dimension $n = 400$ from $m = 4000$ phaseless measurements. For any distance $||z - z'||$, $\mathbb{E}||T(z) - T(z')||$ is computed by averaging over 1000 random pairs (z, z').

Indeed, a rough computation suggests that, for this algorithm, for any points z, z' on the unit sphere,

$$\mathbb{E}(||T(z) - T(z')||^2) \text{ is of order } \frac{n}{m}||z - z'||,$$

which suggests that T is far from being locally Lipschitz on a relatively large set (otherwise the expectation should be of order $||z - z'||^2$). Figure 3 supports this assertion: it shows that, for the *Wirtinger Flow* algorithm described in Subsection 4.1 (to which leave-one-out can be applied [Chen, Chi, Fan, and Ma, 2019]), $||T(z) - T(z')||$ is essentially proportional to $||z - z'||$ and the proportionality constant is smaller than 1. For alternating projections, $||T(z) - T(z')||$ grows much faster at small values of $||z - z'||$.

6 Burer-Monteiro methods

In this final section, we study a family of non-convex algorithms, the so-called *Burer-Monteiro methods*, introduced in [Burer and Monteiro, 2003]. They are applicable to all low-rank matrix recovery problems satisfying reasonably general assumptions. Compared to the previous two sections, where we described techniques to study algorithms applied to specific low-rank problems

satisfying restrictive statistical assumptions, the correction guarantees we present in this section have a much larger application field.

6.1 Definition of Burer-Monteiro methods

In this subsection, we first describe the class of low-rank matrix recovery problems to which Burer-Monteiro methods are applicable (Paragraph 6.1.1). We then define the Burer-Monteiro methods themselves (Paragraph 6.1.2) and give a first overview of their behavior through a basic numerical experiment (Paragraph 6.1.3).

6.1.1 Problems

As previously, we consider problems where one must recover a low-rank matrix X^s from simple information, modeled by the fact that X^s belongs to some set $\mathcal{E}$:

$$\text{minimize } \operatorname{rank}(X) \text{ for } X \in \mathcal{E}. \tag{min-rank}$$

Burer-Monteiro methods can be seen as a "deconvexification" of the convexified techniques presented in Section 2. Therefore, they only apply to problems which admit a *convex relaxation* satisfying precise properties. The first of these properties is that the convex relaxation must be *exact*: Problem (min-rank) and its convex approximation must have the same solution X^s, so that solving the convex version suffices to solve Problem (min-rank).

Assumption 1. *Problem* (min-rank) *has an exact convex relaxation.*

The second property is as follows.

Assumption 2. *The convex relaxation can be written under the form*

$$\textit{minimize } \operatorname{Tr}(CX) \textit{ for } X \in \mathcal{E}_{SDP}, \tag{20}$$

for some $C \in \mathcal{S}_n(\mathbb{R})$ (called the cost matrix) and $\mathcal{E}_{SDP}$ a subset of $\mathcal{S}_n(\mathbb{R})$ which is compact and is defined as the intersection between $\mathcal{S}_n^+(\mathbb{R})$ and an affine space of dimension $m \geq 1$.

When this assumption is satisfied, Problem (20) is a semidefinite program:

$$\begin{aligned} &\text{minimize } \operatorname{Tr}(CX), \\ &\quad \text{with } \mathcal{A}(X) = b, \\ &\quad\quad X \succeq 0, \end{aligned} \tag{SDP}$$

for some linear map $\mathcal{A} : \mathcal{S}_n(\mathbb{R}) \to \mathbb{R}^m$ and some vector $b \in \mathbb{R}^m$ such that

$$\mathcal{E}_{SDP} = \{X \in \mathcal{S}_n(\mathbb{R}), \mathcal{A}(X) = b, X \succeq 0\}.$$

Let us note that, in this section, we constrain all matrices which come into play to have real coefficients. This restriction simply aims at simplifying the proofs: up to minor modifications, our main results also hold true for complex coefficients.

All low-rank recovery problems described in the introduction satisfy Assumptions 1 and 2.

1. We have seen (Equation (1)) that matrix completion problems could be approximated by the following convex problem:

$$\begin{aligned} &\text{minimize } ||X||_* \\ &\quad \text{with } X_{i,j} = X^s_{ij}, \quad \forall (i,j) \in \Omega. \end{aligned}$$

   This problem can be reformulated under the form (SDP)[15] because of the following equality, which holds for all $X \in \mathbb{R}^{n_1 \times n_2}$:

$$||X||_* = \min \left\{ \frac{\mathrm{Tr}(Y) + \mathrm{Tr}(Z)}{2}, Y \in \mathcal{S}_{n_1}(\mathbb{R}), Z \in \mathcal{S}_{n_2}(\mathbb{R}), \left(\begin{smallmatrix} Y & X \\ X^T & Z \end{smallmatrix} \right) \succeq 0 \right\}.$$

   In addition, we have seen in Subsection 2.2 that, under appropriate assumptions, the convex relaxation is exact with high probability.

2. We have already seen that phase retrieval problems admit several convex relaxations of the form (SDP):

(PhaseLift)	(PhaseCut)
$\begin{aligned} &\text{minimize } \mathrm{Tr}(X), \\ &\quad \text{with } \langle X, v_k v_k^* \rangle = \lvert \langle x^s, v_k \rangle \rvert^2, \forall k \leq m, \\ &\quad X \succeq 0. \end{aligned}$	$\begin{aligned} &\text{minimize } \mathrm{Tr}(MU), \\ &\quad \text{with } U_{k,k} = 1, \forall k \leq m, \\ &\quad U \succeq 0. \end{aligned}$

   We have seen that, at least in a specific statistical setting, (PhaseLift) is exact with high probability. This is also true for (PhaseCut).

3. Phase synchronization problems admit a convex relaxation of the form

$$\begin{aligned} &\text{minimize } -\mathrm{Tr}(CU), \\ &\quad \text{with } U_{k,k} = 1, \quad \forall k \leq m, \\ &\quad U \succeq 0. \end{aligned}$$

[15] This guarantees that Assumption 2 holds, except possibly for the compactness requirement (but it turns out to be unnecessary in this case).

In the case where phase measures are contaminated with a Gaussian additive noise (which is the setting we have studied in Section 5), the relaxation is exact with high probability, provided that the noise level satisfies[16] $\sigma \leq c\sqrt{\frac{n}{\log(n)}}$ for some constant $c > 0$ [Zhong and Boumal, 2018].

6.1.2 Definition of Burer-Monteiro methods

Let us assume that Assumptions 1 and 2 hold and discuss (again) the question of how to solve Problem (SDP):

$$\begin{aligned} &\text{minimize } \mathrm{Tr}(CX) \text{ for } X \in \mathcal{E}_{SDP}, \\ &\text{where } \mathcal{E}_{SDP} = \{X \in \mathcal{S}_n(\mathbb{R}), \mathcal{A}(X) = b, X \succeq 0\}. \end{aligned} \tag{SDP}$$

As mentioned in Section 2, many general solvers with rigorous convergence guarantees exist for problems of this form, but they tend to be too slow for medium to high-dimensional applications.

Burer-Monteiro methods, on the other hand, are of heuristic nature. They sometimes fail at correctly identifying the desired minimizer but, when they succeed, they can provide significant speed-ups. Their principle is to "deconvexify" Problem (SDP). At first sight, this can appear counter-intuitive: since Problem (SDP) has been constructed by "convexifying" a low-rank matrix recovery problem, isn't there a risk that, when deconvexifying it, we fall back to a non-convex problem essentially equivalent to the initial one? Actually, no. After the deconvexification, we arrive at a family of non-convex problems, parametrized by an integer p, and some members of this family can be easier to solve than the initial formulation.

The "deconvexification" relies on the observation that, since the convex relaxation is exact (Assumption 1), the solution X^s of Problem (SDP) has low rank. For all $p \in \mathbb{N}^*$, we define

$$\mathcal{E}_{SDP,p} = \mathcal{E}_{SDP} \cap \{X \in \mathcal{S}_n(\mathbb{R}), \mathrm{rank}(X) \leq p\}.$$

For any $p \geq \mathrm{rank}(X^s)$, Problem (SDP) is equivalent to

$$\text{minimize } \mathrm{Tr}(CX) \text{ for } X \in \mathcal{E}_{SDP,p}.$$

Informally, if $p \ll n$, the set $\mathcal{E}_{SDP,p}$ has a much smaller dimension than $\mathcal{E}_{SDP}$. We can therefore hope that minimizing our cost function $X \to \mathrm{Tr}(CX)$ over $\mathcal{E}_{SDP,p}$ requires significantly less computational effort than minimizing it over $\mathcal{E}_{SDP}$.

[16] We observe that this assumption is the same as in Theorem 5.1. This is not a coincidence: the fact that the non-convex generalized power method succeeds in recovering z^{obj} is a crucial tool for establishing the exactness of the convex relaxation.

To take advantage of this dimensionality reduction, the simplest idea is to parametrize $\mathcal{E}_{SDP,p}$ by a low-dimensional manifold $\mathcal{M}_p$. Specifically, we consider the following mapping:

$$V \in \mathcal{M}_p \quad \to \quad VV^T \in \mathcal{E}_{SDP,p},$$

with

$$\mathcal{M}_p = \{V \in \mathbb{R}^{n\times p}, \mathcal{A}(VV^T) = b\}.$$

This mapping is onto: any element $X \in \mathcal{E}_{SDP,p}$ can be factorized as $X = VV^T$ for some $V \in \mathbb{R}^{n\times p}$, since it is a positive semidefinite matrix with rank at most p. And the equality $\mathcal{A}(X) = b$ implies that $\mathcal{A}(VV^T) = b$ and thus that V belongs to $\mathcal{M}_p$.

Using this parametrization, we can rewrite Problem (SDP) as

$$\text{minimize } f_C(V) \overset{def}{=} \text{Tr}(CVV^T) \text{ for } V \in \mathcal{M}_p. \tag{Factorized SDP}$$

In this version of the problem, the unknown V has np coefficients, which is much less than the n^2 coefficients of the original unknown X if $p \ll n$. Manipulating V is therefore less costly than manipulating X. However, compared to (SDP), Problem (Factorized SDP) has the major drawback of not being convex anymore. As a consequence, bad critical points may exist and simple algorithms applicable to Problem (Factorized SDP) are not guaranteed to succeed, although it is possible that they work beautifully.

We call *Burer-Monteiro method* any solver which attempts to solve Problem (min-rank) by applying any reasonable algorithm to the factorized problem (Factorized SDP). This algorithmic scheme is summarized in Figure 4. In these notes, we limit ourselves to Burer-Monteiro methods where Problem (Factorized SDP) is solved by *Riemannian optimization*. This requires the set $\mathcal{M}_p$ to be a Riemannian submanifold[17] of $\mathbb{R}^{n\times p}$. We actually need a slightly stronger assumption.

Assumption 3. *For all $V_0 \in \mathcal{M}_p$, the differential at V_0 of the mapping*

$$\tilde{\mathcal{A}} : V \in \mathbb{R}^{n\times p} \quad \to \quad \mathcal{A}(VV^T) \in \mathbb{R}^m$$

is onto.

Proposition 6.1. *If Assumption 3 is satisfied, $\mathcal{M}_p$ is a Riemannian submanifold of $\mathbb{R}^{n\times p}$, with dimension $np - m$.*

Going back to the examples mentioned at the end of Paragraph 6.1.1, the last two among them (that is phase retrieval, for the (PhaseCut) formulation, and phase synchronization) satisfy

[17] A reader unfamiliar with Riemannian geometry can imagine a Riemannian submanifold as a regular curve or surface inside $\mathbb{R}^{n\times p}$.

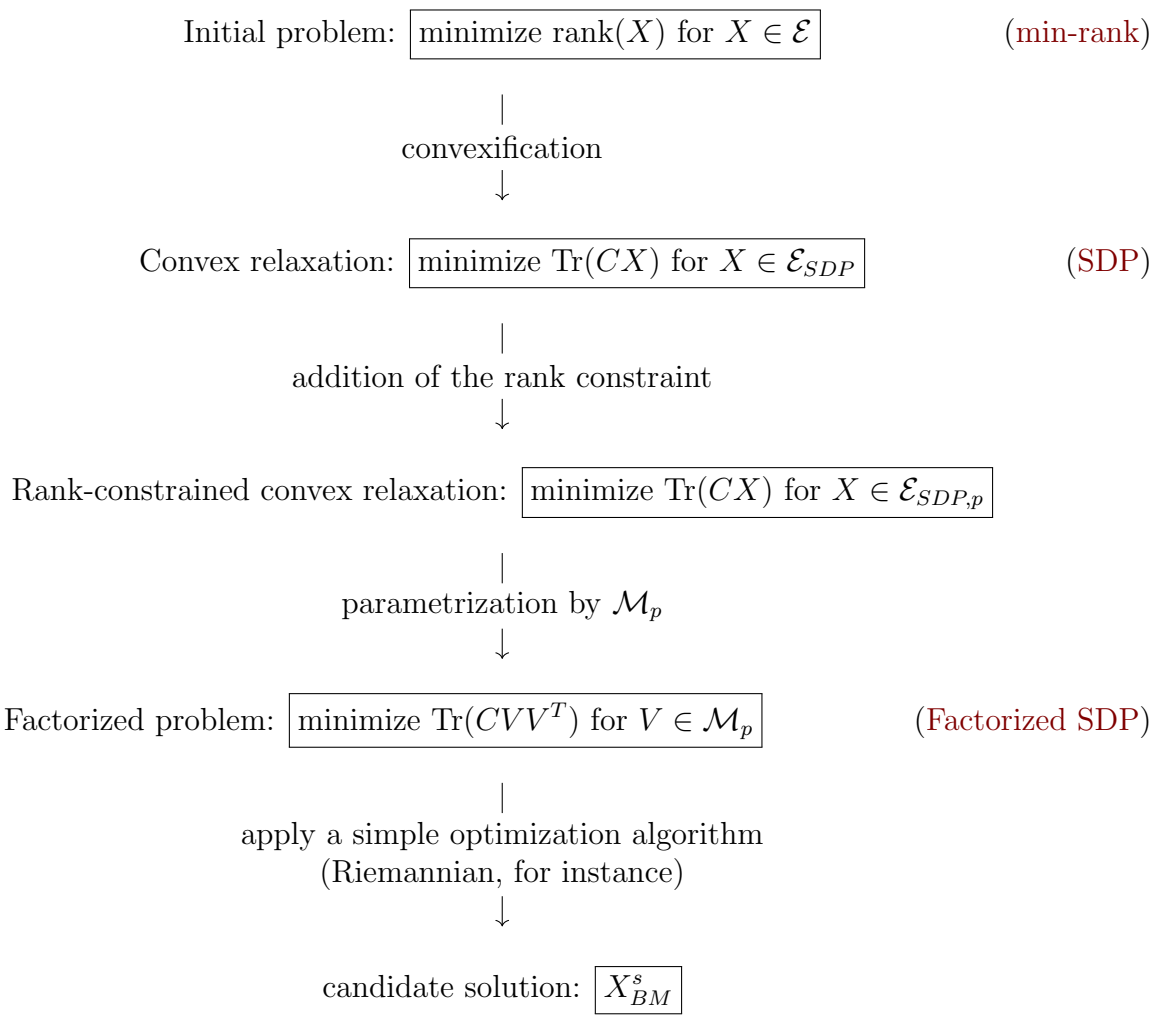


Figure 4: Schematic view of a Burer-Monteiro method

this assumption. For matrix completion, it is less clear; it probably depends on the ground truth X^s.

Riemannian optimization algorithms are in general local optimization methods, which start at some point of the considered manifold, and progressively "move", using the gradient (and possibly the Hessian) of the cost function to decide the movement direction. They oftentimes derive from a classical optimization algorithm over $\mathbb{R}^d$. For instance, two of the most prominent Riemannian algorithms are *Riemannian gradient descent* (which derives from gradient descent over $\mathbb{R}^d$) and *Riemannian Trust-Region method* (which derives from the Trust-Region method over $\mathbb{R}^d$). Many others exist [Absil, Mahony, and Sepulchre, 2009]. Each one of them can be more or less adapted to a given application. Here, we will try to keep our discussion general, and will not make any particular assumption on the Riemannian optimization algorithm.

6.1.3 Numerical experiment

Our goal, in the rest of this section, is to provide a partial answer to the following question:

When can we guarantee that a Burer-Monteiro method succeeds?

We focus on guarantees which require essentially no hypothesis besides Assumptions 1, 2 and 3, so that our results apply to as many (min-rank) problems as possible. In particular, $C, \mathcal{A}, b$ can be arbitrary.

Nevertheless, we cannot avoid making an hypothesis on p. We must choose this hypothesis with great care, since it determines the practical relevance of the correctness guarantees. Indeed, in Problem (Factorized SDP), the unknown V has dimension proportional to p so Burer-Monteiro methods run much faster when p is small. As far as I know, large values of p are rarely used in practice, so it is important for correctness guarantees to hold for as small values of p as possible. On the other hand, whether Problem (Factorized SDP) has bad critical points or not may strongly depend on p. Consequently, it may be that establishing guarantees for small values of p is very difficult or even impossible.

To get a preliminary intuition, let us study the numerical performance of a Burer-Monteiro method on toy phase retrieval problems. The specific method we have chosen for this experiment uses the (PhaseCut) relaxation of the phase retrieval problem, and solves the corresponding (Factorized SDP) by Riemannian gradient descent. In this case, when the relaxation is exact, its solution has rank 1, so p can a priori take any positive integer value. In our tests, we use either $p = 1$ or $p = 2$. Signals have dimension $n = 32$ and are chosen according to a complex normal distribution. We vary the number of measurements between $m = 0$ and $m = 8n$. Results are displayed on Figure 5. As a reference point, the figure also displays the success rate of the convex (PhaseCut) method (which consists in directly solving Problem (SDP) with an interior-point method, without considering Problem (Factorized SDP)).

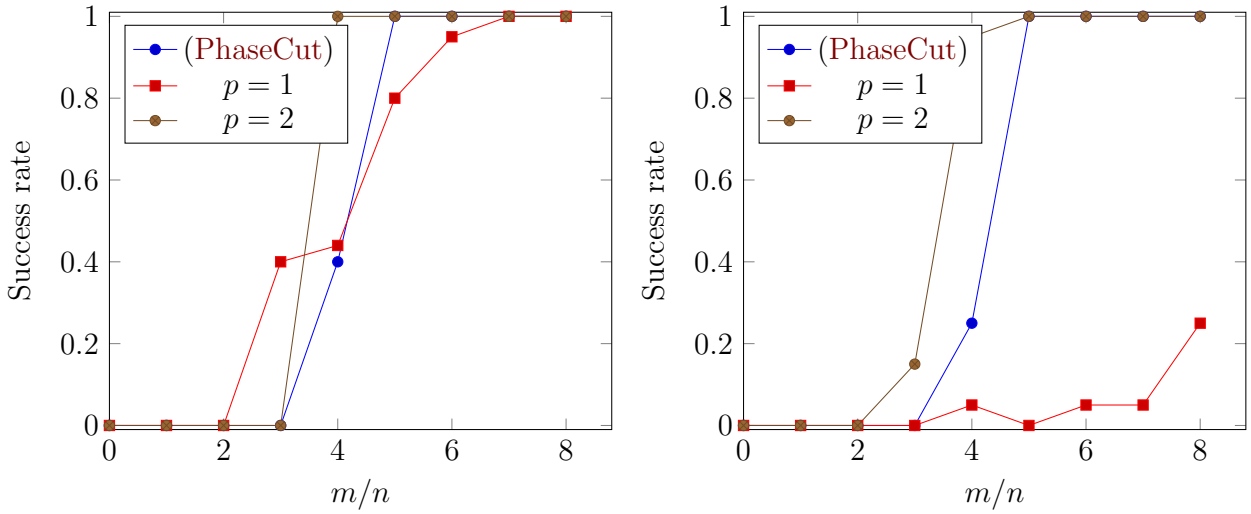


Figure 5: Success rate, as a function of m/n, for three algorithms: (PhaseCut), a Burer-Monteiro method with $p = 1$, a Burer-Monteiro method with $p = 2$. The signal dimension is $n = 32$. The figure on the left corresponds to measurement vectors independently chosen according to normal laws and the figure on the right to measurement vectors representing a "wavelet transform". Success rates have been computed by averaging the results of 20 reconstruction attempts.

The experiment covers two types of measurement vectors: realizations of independent normal vectors on the one hand and vectors describing a "wavelet transform" on the other hand. Vectors of the second type are more "structured" (in particular, some of them are very close to being orthogonal to each other while others are strongly correlated). It is known that phase retrieval problems with one or the other measurement type have significantly different intrinsic properties (regarding stability to noise, for instance); problems of the second type generally tend to be more difficult. Hence, it is possible that Burer-Monteiro methods do not behave the same for the two measurement types; this is the reason for including both in our experiment.

From Figure 5, we see that, when $p = 1$, the behavior of our Burer-Monteiro method indeed depends on the measurement type: it works fine (that is, essentially as well as (PhaseCut)) for normal vectors, and fails for the wavelet transform. On the contrary, when $p = 2$, it works as well as (PhaseCut) in both cases.

Much more extensive numerical experiments can be found in the literature, involving different Burer-Monteiro methods and other low-rank recovery problems than phase retrieval [Burer and Monteiro, 2003; Journée, Bach, Absil, and Sepulchre, 2010; Boumal, 2015]. Overall, they lead to a conclusion compatible with our toy experiment: some failure cases can be observed if p is equal or almost equal to the rank of the solution of Problem (SDP). However, it suffices for p to be "slightly larger" than the rank for these failure cases to disappear.

6.2 Success guarantees when $\frac{p(p+1)}{2} > m$

The observations of Paragraph 6.1.3 suggest that Burer-Monteiro have excellent empirical success rates, even for quite small values of p. Can we prove it?

In this subsection, we describe the correctness guarantees established in [Boumal, Voroninski, and Bandeira, 2020], which can informally be stated as follows: under Assumptions 1, 2, 3, provided that

$$\frac{p(p+1)}{2} > m,$$

Riemannian algorithms succeed at solving almost any problem of the form (Factorized SDP) (that is, Burer-Monteiro methods succeed). We recall that m is the number of affine constraints in Problem (Factorized SDP), that is, the dimension of the range of $\mathcal{A}$.

In the following paragraph, we precisely state these guarantees and in the next ones, we give an idea of how to prove them.

6.2.1 Precise statement

We first recall a result of Section 4, Theorem 4.4: if we run gradient descent over an analytic function $f : \mathbb{R}^d \to \mathbb{R}$, we obtain (under relatively weak assumptions) a converging sequence of

iterates, whose limit x_* is a second-order critical point of f:

$$\nabla f(x_*) = 0 \quad \text{and} \quad \nabla^2 f(x_*) \succeq 0.$$

It turns out that this property is also true for Riemannian algorithms, at least part of them [Boumal, Absil, and Cartis, 2016; Criscitiello and Boumal, 2019]: when applied to Problem (Factorized SDP), these algorithms are guaranteed to find a matrix V_* which is a second-order critical point of f_C:

$$\nabla_{\mathcal{M}_p} f_C(V_*) = 0 \quad \text{and} \quad \nabla^2_{\mathcal{M}_p} f_C(V_*) \succeq 0.$$

Here, $\nabla_{\mathcal{M}_p}$ and $\nabla^2_{\mathcal{M}_p}$ are respectively the gradient and Hessian of f_C, restricted to the manifold $\mathcal{M}_p$.

Therefore, we can establish correctness guarantees for Burer-Monteiro methods with the same argument as in Subsection 4.2: the minimizers[18] of Problem (Factorized SDP) are second-order critical points of f_C (it is a general property of minimizers). If they are the *only* second-order critical points, Burer-Monteiro methods are guaranteed to correctly solve Problem (Factorized SDP) and thus the initial problem (min-rank).

Theorem 6.2 ([Boumal, Voroninski, and Bandeira, 2020])**.** *Let $\mathcal{A} : \mathcal{S}_n(\mathbb{R}) \to \mathbb{R}^m, b \in \mathbb{R}^m$ be fixed. We assume that Assumptions 1, 2 and 3 are verified and that*

$$\frac{p(p+1)}{2} > m.$$

Then, for all cost matrices $C \in \mathcal{S}_n(\mathbb{R})$ outside some zero Lebesgue measure set, Problem (Factorized SDP) has no second-order critical point except its minimizers.

The following paragraphs contain an overview of the proof of this theorem. Parts of it are relatively technical and possibly difficult to understand, but I hope that they allow, even without being read in full detail, to get an idea of which tools are necessary to prove this result, and why the condition $\frac{p(p+1)}{2} > m$ appears.

6.2.2 Proof principle for Theorem 6.2

Let us assume that $\mathcal{A}, b$ are fixed and Assumptions 1, 2 and 3 are verified. Theorem 6.2 follows from the following two lemmas.

Lemma 6.3. *For all $V_* \in \mathcal{M}_p$, whatever the cost matrix C, if*

$$\nabla_{\mathcal{M}_p} f_C(V_*) = 0, \quad \nabla^2_{\mathcal{M}_p} f_C(V_*) \succeq 0 \quad \textit{and} \quad \operatorname{rank}(V_*) < p,$$

then V_* is a minimizer of Problem (Factorized SDP).

[18] I use plural since Problem (Factorized SDP) never has a single minimizer: if V_* is a minimizer, so is $V_* G$ for any orthogonal $p \times p$ matrix G.

Lemma 6.4. *We assume that* $\frac{p(p+1)}{2} > m$.

For any matrix $C \in \mathcal{S}_n(\mathbb{R})$ *outside some zero Lebesgue measure set, there does not exist* $V_* \in \mathcal{M}_p$ *such that*

$$\nabla_{\mathcal{M}_p} f_C(V_*) = 0 \quad \textit{and} \quad \operatorname{rank}(V_*) = p.$$

Indeed, if these two lemmas are true, it means that, when $\frac{p(p+1)}{2} > m$, for all cost matrices outside some zero Lebesgue measure set:

- Problem (Factorized SDP) has no second-order critical point (nor even a first-order one) with rank p (Lemma 6.4);
- Problem (Factorized SDP) has no second-order critical point with rank strictly smaller than p, except its solutions (Lemma 6.3).

Consequently, Problem (Factorized SDP) has no second-order critical point except its solutions (let us note that elements of $\mathcal{M}_p$ have p columns and are therefore of rank at most p).

We will not explain the proof of Lemma 6.3. We simply present a rough geometrical interpretation for it. Let us imagine that the second-order critical points of f_C are exactly its local minimizers (it is not true in full generality, but this hypothesis helps developing an intuition). If V_* is a local minimizer of f_C over $\mathcal{M}_p$, then $V_*V_*^T$ is a local minimizer of $X \to \langle C, X\rangle$ over $\mathcal{E}_{SDP,p}$ (since $V \in \mathcal{M}_p \to VV^T \in \mathcal{E}_{SDP,p}$ is a parametrization of $\mathcal{E}_{SDP,p}$).

Now, Lemma 6.3 is true because of the following (non obvious) property: if $\operatorname{rank}(V_*V_*^T) < p$, the set of possible "displacement directions" around $V_*V_*^T$ in $\mathcal{E}_{SDP,p}$ forms a cone of elements of $\mathcal{S}_n(\mathbb{R})$, whose convex envelope contains $\mathcal{E}_{SDP}$. When $V_*V_*^T$ is a local minimizer of $X \to \langle C, X\rangle$ over $\mathcal{E}_{SDP,p}$, we have $\left\langle C, X - V_*V_*^T\right\rangle \geq 0$ for all X in the "displacement cone", and thus also for all X in the convex envelope of the cone. Therefore, if the envelope contains $\mathcal{E}_{SDP}$, we have $\left\langle C, X - V_*V_*^T\right\rangle \geq 0$ for all $X \in \mathcal{E}_{SDP}$. In particular, $V_*V_*^T$ is a solution of Problem (SDP); this implies that V_* is a solution of Problem (Factorized SDP).

6.2.3 Idea of proof for Lemma 6.4

We want to show that, when $\frac{p(p+1)}{2} > m$, there exists no $V_* \in \mathcal{M}_p$ such that

$$\nabla_{\mathcal{M}_p} f_C(V_*) = 0 \quad \text{and} \quad \operatorname{rank}(V_*) = p, \tag{21}$$

except possibly if C belongs to some zero Lebesgue measure set.

Let us first give an explicit description of the set of cost matrices C for which, on the contrary, there exists V_* satisfying Equation (21). Then we will show that this set has Lebesgue measure zero.

For any full rank $V_* \in \mathcal{M}_p$, the map $V \in \mathcal{M}_p \to VV^T \in \mathcal{E}_{SDP,p}$ is essentially[19] a diffeomorphism in some neighborhood of V_* (because one can check that the rank of its differential is locally constant). The equality $\nabla_{\mathcal{M}_p} f_C(V_*) = 0$ is therefore equivalent to the fact that the gradient of $X \in \mathcal{E}_{SDP,p} \to \langle C, X \rangle$ is zero at $V_* V_*^T$. In other words, it is equivalent to C being orthogonal to the tangent space to $\mathcal{E}_{SDP,p}$ at $V_* V_*^T$.

Consequently, the set of cost matrices C for which there exists V_* satisfying Equation (21) is included in (actually, equal to)

$$\bigcup_{M \in \mathcal{E}_{SDP,p}, \text{rank}(M)=p} (T_M \mathcal{E}_{SDP,p})^\perp. \tag{22}$$

(For any $M \in \mathcal{E}_{SDP,p}$ with rank p, we denote by $T_M \mathcal{E}_{SDP,p}$ the tangent space to $\mathcal{E}_{SDP,p}$ at M.)

If N is the dimension of $\mathcal{S}_n(\mathbb{R})$ (that is, $\frac{n(n+1)}{2}$) and D the dimension of the manifold $\{M \in \mathcal{E}_{SDP,p}, \text{rank}(M) = p\}$, the vector space $(T_M \mathcal{E}_{SDP,p})^\perp$ has dimension $N - D$ for any $M \in \mathcal{E}_{SDP,p}$ with rank p. As a consequence, the set in Equation (22) is the union, parametrized by a D-dimensional manifold, of $(N-D)$-dimensional spaces. Intuitively, as illustrated by Figure 6a, it is a set with "dimension"[20] at most

$$(N - D) + D = N.$$

We have shown that the set of "problematic" cost matrices is an at most N-dimensional subset of the N-dimensional vector space $\mathcal{S}_n(\mathbb{R})$. An N-dimensional subset of an N-dimensional space has no reason to have zero Lebesgue measure: this conclusion does not seem very useful. Fortunately, it is possible to refine the computation of the dimension with the help of the following proposition.

Proposition 6.5. *Let us assume that $\frac{p(p+1)}{2} > m$. For any $M \in \mathcal{E}_{SDP,p}$ with rank p, there exists a segment in $\mathcal{E}_{SDP,p}$ such that M is in the interior of the segment and the tangent space to $\mathcal{E}_{SDP,p}$ is constant on the segment.*

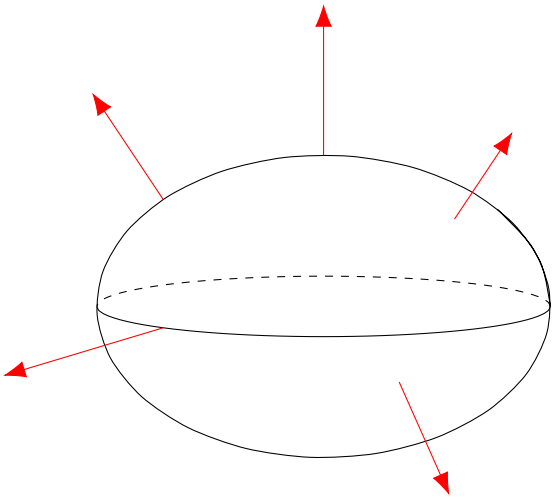
This proposition implies that, in Equation (22), the parametrization by rank p elements of $\mathcal{E}_{SDP,p}$ is redundant. The set can actually be written as a union of $(N-D)$-dimensional vector spaces, parametrized by a set with dimension only $D-1$. The "dimension" of this set is therefore at most

$$(N - D) + (D - 1) = N - 1,$$

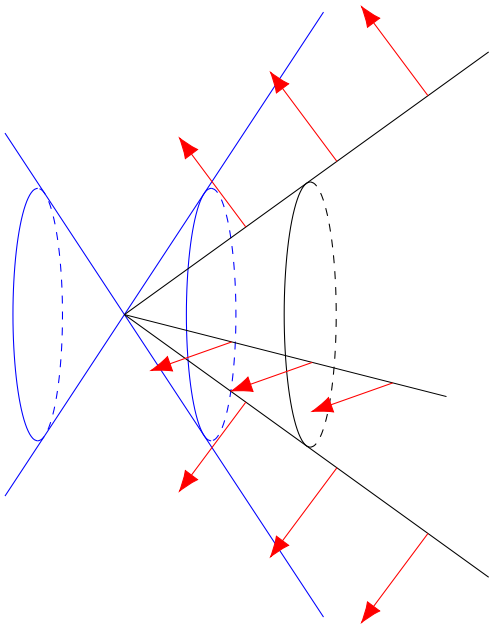
which means that it is a zero Lebesgue measure subset of $\mathcal{S}_n(\mathbb{R})$. Figure 6b illustrates this argument.

[19] It is not exactly a diffeomorphism since the map is invariant to right multiplication with any orthogonal matrix. To get an exact diffeomorphism, we must quotient $\mathcal{M}_p$ by the set of orthogonal matrices.

[20] I use quotes because the set is not a manifold and therefore has no "dimension" in the usual sense of this word.



(a) A 2-dimensional manifold included in $\mathbb{R}^3$. The union of lines which are orthogonal to one of its tangent spaces is the whole ambiant space $\mathbb{R}^3$; it has dimension 3.



(b) A manifold (here a cone without its vertex, in black) whose points are all inside some segment along which the tangent space is constant. The union of lines which are orthogonal to one of its tangent spaces has dimension 2 (it is the cone drawn in blue).

6.2.4 Idea of proof for Proposition 6.5

Let $M \in \mathcal{E}_{SDP,p}$ with rank p be fixed.

We want to show that there is a segment in $\mathcal{E}_{SDP,p}$, containing M, over which the tangent space to $\mathcal{E}_{SDP,p}$ does not vary. For this, we need an explicit expression for the tangent space.

Let us recall that

$$\mathcal{E}_{SDP,p} = \{X \succeq 0\} \cap \{X, \mathcal{A}(X) = b\} \cap \{X, \text{rank}(X) \leq p\}.$$

Since M is positive semidefinite and $\text{rank}(M) = p$, all symmetric matrices close enough to M have rank at least p and all rank p symmetric matrices close enough to M are positive semidefinite. Consequently, in some neighborhood of M, $\mathcal{E}_{SDP,p}$ coincides with

$$\{X, \mathcal{A}(X) = b\} \cap \{X, \text{rank}(X) = p\}.$$

From this remark (and Assumption 3), we can show that

$$\begin{aligned} T_M\mathcal{E}_{SDP,p} &= \mathrm{Ker}(\mathcal{A}) \cap T_M\{X, \mathrm{rank}(X) = p\} \\ &= \mathrm{Ker}(\mathcal{A}) \cap \{X \in \mathcal{S}_n(\mathbb{R}), \langle v, Xv\rangle = 0 \text{ for all } v \in \mathrm{Range}(M)^\perp\}. \end{aligned}$$

We see that the tangent space depends on M only through the range of M. Therefore, to show that the tangent space is constant on a segment, it suffices to show that there exists a segment of $\mathcal{M}_p$ (containing M) whose elements all have the same range.

The set $\{X \in \mathcal{S}_n(\mathbb{R}), \mathrm{Im}(X) \subset \mathrm{Im}(M)\}$ has dimension

$$\frac{\dim(M)(\dim(M)+1)}{2} = \frac{p(p+1)}{2}.$$

Since $\frac{p(p+1)}{2} > m = \dim(\mathrm{Range}(\mathcal{A}))$, the set must contain a matrix $H \neq 0$ such that

$$\mathcal{A}(H) = 0.$$

We fix such a matrix H.

For any $t \in \mathbb{R}$ close enough to 0,

- $\mathrm{rank}(M + tH) \geq \mathrm{rank}(M) = p$ and $\mathrm{Range}(M + tH) \subset \mathrm{Range}(M)$, from which we deduce

$$\mathrm{rank}(M + tH) = p \quad \text{and} \quad \mathrm{Range}(M + tH) = \mathrm{Range}(M);$$

- $M + tH \succeq 0$;
- $\mathcal{A}(M + tH) = \mathcal{A}(M) = b$.

In particular, for $\epsilon > 0$ small enough, $[M - \epsilon H; M + \epsilon H]$ is a segment included in $\mathcal{E}_{SDP,p}$ whose elements all have the same range. This concludes the proof.

6.3 Optimality of Theorem 6.2

The theorem we have presented in the previous subsection, Theorem 6.2, states that Burer-Monteiro methods succeed for almost all problems satisfying Assumptions 1, 2, 3, provided that

$$\frac{p(p+1)}{2} > m. \tag{23}$$

The assumptions can be modified so as to include more problems [Bhojanapalli, Boumal, Jain, and Netrapalli, 2018]. In addition, if one applies a slight perturbation to the cost matrix C before running a suitable algorithm, it is possible to guarantee that Burer-Monteiro methods succeed at

solving all (and not almost all) problems satisfying the assumptions with high probability, and with a running time polynomial in the precision [Pumir, Jelassi, and Boumal, 2018; Cifuentes and Moitra, 2019].

This means that, when Inequality (23) holds, Burer-Monteiro methods perform well, and are also relatively well understood from a theoretical point of view. Unfortunately, Inequality (23) is quite disappointing. Indeed, it can only be satisfied if $p \geq \sqrt{2m} + o(1)$. Let us recall that Burer-Monteiro methods are applicable as soon as $p \geq \text{rank}(X^s)$, and that they numerically seem to almost always succeed even if p is only slightly larger than $\text{rank}(X^s)$ (Paragraph 6.1.3). In practice, since the computational cost increases with p, the most frequently used values of p are therefore of order $\text{rank}(X^s)$, and not of order $\sqrt{2m}$[21] Consequently, Inequality (23) is rarely satisfied in concrete situations.

It is therefore of importance to determine whether Condition (23) is optimal (that is, necessary in order for Theorem 6.2 to hold) or whether it can be improved. This is the question considered in [Waldspurger and Waters, 2020], and the answer is that the condition is essentially optimal.

Theorem 6.6. *Let $\mathcal{A}, b$ be fixed. We assume that Assumptions 1, 2 and 3 are verified, as well as a "minimal intersection" assumption (see below).*

Let us define $r_0 = \min\{\text{rank}(X), X \in \mathcal{E}_{SDP}\}$.

For all p such that

$$\frac{p(p+1)}{2} + pr_0 \leq m, \tag{24}$$

there exists a set $\mathcal{C} \subset \mathcal{S}_n(\mathbb{R})$ with non-zero Lebesgue measure such that, for all $C \in \mathcal{C}$,

1. $\text{rank}(X^s) = r_0$;
2. *Problem* (Factorized SDP) *has a second-order critical point which is not a global minimizer.*

Why does Theorem 6.6 imply, as asserted above, that Condition (23) is essentially necessary for Theorem 6.2 to hold? Let us remember that, in most applications, the matrices to be recovered have very small rank, typically of order 1, hence $r_0 = O(1)$. Therefore, Inequality (24) is verified when $p \leq \sqrt{2m} + o(1)$: up to this $o(1)$, this is the exact negation of Inequality (23). As a consequence, we can rephrase Theorem 6.6 as: "When Inequality (23) does not hold, Theorem 6.2 is false: there exists a non-negligible set of cost matrices C for which Problem (Factorized SDP) has bad critical points, and this is even if we restrict ourselves to problem instances where the solution has the smallest possible rank r_0".

We do not provide the exact definition of the "minimal intersection" assumption here. We invite the curious reader to look for it in [Waldspurger and Waters, 2020]. It consists in requiring that the intersection between two specific subspaces of $\mathbb{R}^{n \times p}$ is as small as possible. I do not

[21] Typically, $\text{rank}(X^s) \ll \sqrt{2m}$. In many interesting applications, $\text{rank}(X^s) = O(1)$ while $\sqrt{2m} = O(\sqrt{n})$.

have a satisfactory geometric interpretation of this condition, but it is necessary for the proof of Theorem 6.6. Fortunately, there are good reasons to think that it is "generically" satisfied and it is at least satisfied in all applications discussed in [Waldspurger and Waters, 2020]. It is therefore not a significant restriction to the applicability of Theorem 6.6.

6.3.1 Idea of proof for Theorem 6.6

The proof is in two parts.

1. First part: we show that, when a cost matrix C satisfies Properties 1 and 2 of Theorem 6.6 and some additional "non-degeneracy" conditions, then all cost matrices in some neighborhood of C satisfy Properties 1 and 2 as well.

   As a consequence, in order to prove that there exists a set with non-zero Lebesgue measure of cost matrices with Properties 1 and 2, it suffices to show that there exists *one* cost matrix C satisfying these two properties as well as the non-degeneracy conditions.

2. Second part: we show the existence of this one cost matrix.

We do not describe the first part of the proof, which relies on standard arguments from differential geometry. Let us focus on the second part. We will present the main arguments of this part, and try to explain where the condition $\frac{p(p+1)}{2} + pr_0 \leq m$ comes from. For simplicity, we ignore the non-degeneracy conditions. We must therefore only explain how to construct C satisfying Properties 1 and 2.

Let us fix $X_0 \in \mathcal{E}_{SDP}$ with rank r_0 and $V \in \mathcal{M}_p$. We are going to construct C such that[22]

- the solution X^s of Problem (SDP) is X_0, hence $\operatorname{rank}(X^s) = r_0$ and Property 1 holds;
- V is a second-order critical point of Problem (Factorized SDP), but not a global minimizer, hence Property 2 also holds.

First step: analytic formulations We first rewrite these properties under a more analytic form.

A sufficient (and almost necessary) condition for the first property to hold is given by the classical duality theory of semidefinite programs. We state it in the following proposition.

[22] Slightly surprinsingly, a cost matrix C satisfying the desired properties exists for almost all choices of $X_0 \in \mathcal{E}_{SDP}, V \in \mathcal{M}_p$ such that $\operatorname{rank}(X_0) = r_0$.

Proposition 6.7. *If there exist $g_1 \in \mathbb{R}^m, C_1 \in \mathcal{S}_n(\mathbb{R})$ such that*

$$\begin{aligned} C &= \mathcal{A}^*(g_1) + C_1, \\ C_1 X_0 &= 0, \\ C_1 &\succeq 0, \\ \operatorname{rank}(C_1) &= n - r_0, \end{aligned}$$

then X_0 is the unique solution of Problem (SDP).

The second property can be reformulated using the explicit formulas of $\nabla_{\mathcal{M}_p} f_C$ and $\nabla^2_{\mathcal{M}_p} f_C$.

Proposition 6.8. *V is a second-order critical point of Problem (Factorized SDP) if and only if there exist $g_2 \in \mathbb{R}^m, C_2 \in \mathcal{S}_n(\mathbb{R})$ such that*

$$\begin{aligned} C &= \mathcal{A}^*(g_2) + C_2, \\ C_2 V &= 0, \\ \forall \dot{V} \in T_V \mathcal{M}_p, \quad &\left\langle C_2, \dot{V}\dot{V}^T \right\rangle \geq 0. \end{aligned}$$

From Propositions 6.7 and 6.8, in order to construct C as desired, we simply need to find g_1, g_2, C_1, C_2 such that

$$\mathcal{A}^*(g_1) + C_1 = \mathcal{A}^*(g_2) + C_2, \tag{25}$$
$$C_1 X_0 = 0, \tag{26}$$
$$C_1 \succeq 0, \tag{27}$$
$$\operatorname{rank}(C_1) = n - r_0, \tag{28}$$
$$C_2 V = 0, \tag{29}$$
$$\forall \dot{V} \in T_V \mathcal{M}_p, \quad \left\langle C_2, \dot{V}\dot{V}^T \right\rangle \geq 0. \tag{30}$$

Indeed, if we find such g_1, g_2, C_1, C_2, the matrix $C = \mathcal{A}^*(g_1) + C_1$ satisfies Properties 1 and 2.

Second step: construction of g_1, g_2, C_1, C_2 Without loss of generality, we can set $g_2 = 0$ and construct g_1, C_1, C_2 only.

One can also show that, if g_1, C_1, C_2 satisfy Properties (25) to (29), it is possible to define $\tilde{C}_1, \tilde{C}_2$, by adding a suitable semidefinite positive matrix to C_1 and C_2, such that $g_1, \tilde{C}_1, \tilde{C}_2$ also satisfy Properties (25) to (29), and Property (30) as well. Consequently, it is enough to construct g_1, C_1, C_2 with Properties (25) to (29).

Let us explain the construction in the simplified setting where

$$\begin{aligned}\mathrm{Range}(X_0) &= \{(x_1, \dots, x_{r_0}, 0, \dots, 0) \text{ with } x_1, \dots, x_{r_0} \in \mathbb{R}\},\\ \mathrm{Range}(V) &= \{(0, \dots, 0, x_{r_0+1}, \dots, x_{r_0+p}, 0, \dots, 0) \text{ with } x_{r_0+1}, \dots, x_{r_0+p} \in \mathbb{R}\}.\end{aligned}$$

In this setting, we have the following proposition.

Proposition 6.9. *For any $g_1 \in \mathbb{R}^m$, the following properties are equivalent.*

1. *There exist $C_1, C_2 \in \mathcal{S}_n(\mathbb{R})$ such that g_1, C_1, C_2 satisfy Properties (25) to (29).*
2. *$\mathcal{A}^*(g_1)$ admits a block-decomposition of the form*

$$\mathcal{A}^*(g_1) = \begin{pmatrix} G_1 & 0 & G_2 \\ 0 & G_3 & G_4 \\ G_2^T & G_4^T & G_5 \end{pmatrix}, \tag{31}$$

for some $G_1 \in \mathcal{S}_{r_0}(\mathbb{R})$, $G_2 \in \mathbb{R}^{r_0 \times (n-(r_0+p))}$, $G_3 \in \mathcal{S}_p(\mathbb{R})$, $G_4 \in \mathbb{R}^{p \times (n-(r_0+p))}$, $G_5 \in \mathcal{S}_{n-(r_0+p)}(\mathbb{R})$, such that $G_3 \prec 0$.

We observe that the mapping

$$\begin{array}{rccccc} \Phi: & \mathbb{R}^m & \to & \mathbb{R}^{r_0 \times p} & \times & \mathcal{S}_p(\mathbb{R}) \\ & g & \to & \Big((\mathcal{A}^*(g)_{k,l})_{\substack{1 \le k \le r_0 \\ r_0 < l \le r_0+p}} & , & (\mathcal{A}^*(g)_{k,l})_{r_0 < k,l \le r_0+p} \Big) \end{array}$$

is a linear map between a vector space of dimension m and a vector space of dimension

$$\frac{p(p+1)}{2} + r_0 p \le m.$$

We can thus expect that it is generically surjective. And actually, using the minimal intersection property, we can rigorously guarantee that it is surjective.

Let us now fix any $H \prec 0$ in $\mathcal{S}_p(\mathbb{R})$. Since Φ is surjective, there exists g_1 such that $\Phi(g_1) = (0, H)$. Let us fix one such g_1. Then the matrix $\mathcal{A}^*(g_1)$ admits a block-decomposition of the form (31) (with $G_3 = H$). From Proposition 6.9, this is enough to ensure the existence of C_1, C_2 such that g_1, C_1, C_2 satisfy Properties (25) to (29).

6.4 Summary and open questions

To summarize, we can distinguish three regimes regarding the behavior of the Burer-Monteiro heuristic.

- When the factorization rank p is equal to, or barely larger than $\operatorname{rank}(X^s)$, it is numerically observed (see Figure 5, for instance) that Burer-Monteiro methods succeed at solving some problems, but that there are also situations of practical interest in which they fail.
- When p is larger than $\operatorname{rank}(X^s)$ but smaller than $\sqrt{2m} + o(1)$, Burer-Monteiro methods seem to correctly find a global minimum in almost all situations of practical interest, but no theoretical explanation of this phenomenon exists.
- When p is larger than $\sqrt{2m} + o(1)$, Burer-Monteiro methods almost always converge to a global minimum and satisfactory theoretical guarantees exist (Theorem 6.2).

In practice, computational efficiency commands to choose p as small as possible among the values which allow to find a global minimum. The most interesting regime is therefore the second one, when $\operatorname{rank}(X^s) < p \leq \sqrt{2m} + o(1)$. Thus, a major open question is:

How can we explain the good numerical behavior of Burer-Monteiro heuristics in this regime?

More precisely, it follows from Theorem 6.6 that, when $p \leq \sqrt{2m} + o(1)$, there exists a non-zero Lebesgue measure set of cost matrices C for which Problem (Factorized SDP) has bad second-order critical points. How come that we do not seem to encounter these matrices in numerical experiments? Is it because problematic cost matrices form a subset of $\mathcal{S}_n(\mathbb{R})$ with an extremely small, although non-zero, volume, which makes it extremely unlikely to encounter one of them in an experiment? Is it that we encounter them, but that the attraction basin of bad critical points is very small, so that convergence to a global minimum occurs despite the existence of second-order critical points?

Other open questions have to do with more concrete algorithmic aspects of Burer-Monteiro methods. For instance, how to deal with problems for which Assumption 2 or 3 does not hold? For Assumption 2, this has already been studied in [Bhojanapalli, Boumal, Jain, and Netrapalli, 2018]. Independently, which Riemannian optimization algorithms are best suited to which problems? This is a crucial issue in applications, but probably a delicate one. A notable source of difficulties is the possible ill-conditioning of Problem (Factorized SDP) close to the solution. About this point, the interested reader can refer to [Tong, Ma, and Chi, 2020] and references therein.

7 Bibliography

P.-A. Absil, R. Mahony, and B. Andrews. Convergence of the iterates of descent methods for analytic cost functions. SIAM Journal on Optimization, 16(2):531–547, 2005.

P.-A. Absil, R. Mahony, and R. Sepulchre. Optimization algorithms on matrix manifolds. Princeton University Press, 2009.

R. Balan, B. G. Bodmann, P. G. Casazza, and D. Edidin. Painless reconstruction from magnitudes of frame coefficients. Journal of Fourier Analysis and Applications, 15(4):488–501, 2009.

A. S. Bandeira, N. Boumal, and A. Singer. Tightness of the maximum likelihood semidefinite relaxation for angular synchronization. Mathematical Programming, 163(1-2):145–167, 2017.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences, 2(1):183–202, 2009.

S. Bhojanapalli, B. Neyshabur, and N. Srebro. Global optimality of local search for low rank matrix recovery. In Advances in Neural Information Processing Systems 29, 2016.

S. Bhojanapalli, N. Boumal, P. Jain, and P. Netrapalli. Smoothed analysis for low-rank solutions to semidefinite programs in quadratic penalty form. In Proceedings of the 31st Conference On Learning Theory, pages 3243–3270, 2018.

B. Borchers and J. Young. Implementation of a primal–dual method for SDP on a shared memory parallel architecture. Computational Optimization and Applications, 37(3):355–369, 2007.

N. Boumal. A Riemannian low-rank method for optimization over semidefinite matrices with block-diagonal constraints. Technical report, Inria, 2015. http://arxiv.org/abs/1506.00575.

N. Boumal. Nonconvex phase synchronization. SIAM Journal on Optimization, 26(4):2355–2377, 2016.

N. Boumal, P.-A. Absil, and C. Cartis. Global rates of convergence for nonconvex optimization on manifolds. IMA Journal of Numerical Analysis, 2016.

N. Boumal, V. Voroninski, and A. S. Bandeira. Deterministic guarantees for Burer-Monteiro factorizations of smooth semidefinite programs. Communications on Pure and Applied Mathematics, 73(3):581–608, 2020.

S. Burer and R. D. C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. Mathematical Programming, 95(2):329–357, 2003.

E. J. Candès and X. Li. Solving quadratic equations via PhaseLift when there are about as many equations as unknowns. Foundations of Computational Mathematics, 14(5):1017–1026, 2014.

E. J. Candès and B. Recht. Exact matrix completion via convex optimization. Foundations of Computational Mathematics, 9(6):717–772, 2009.

E. J. Candès, T. Strohmer, and V. Voroninski. Phaselift: exact and stable signal recovery from magnitude measurements via convex programming. Communications on Pure and Applied Mathematics, 66(8):1241–1274, 2013.

E. J. Candès, X. Li, and M. Soltanolkotabi. Phase retrieval from coded diffraction patterns. Applied and Computational Harmonic Analysis, 39(2):277–299, 2015.

E. J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. IEEE Transactions on Information Theory, 56(5):2053–2080, 2010.

E. J. Candès, X. Li, and M. Soltanolkotabi. Phase retrieval via Wirtinger flow: theory and algorithms. IEEE Transactions of Information Theory, 61(4):1985–2007, 2015.

A. Chai, M. Moscoso, and G. Papanicolaou. Array imaging using intensity-only measurements. Inverse Problems, 27(1), 2011.

R. Chandra, Z. Zhong, J. Hontz, V. McCulloch, C. Studer, and T. Goldstein. Phasepack: A phase retrieval library. Asilomar Conference on Signals, Systems, and Computers, 2017.

Y. Chen. Incoherence-optimal matrix completion. IEEE Transactions on Information Theory, 61(5):2909–2923, 2015.

Y. Chen and E. J. Candès. Solving random quadratic systems of equations is nearly as easy as solving linear systems. Communications on Pure and Applied Mathematics, 70(5):2133–2150, 2017.

Y. Chen and Y. Chi. Harnessing structures in big data via guaranteed low-rank matrix estimation: Recent theory and fast algorithms via convex and nonconvex optimization. IEEE Signal Processing Magazine, 35(4):14–31, 2018.

Y. Chen and M. J. Wainwright. Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees. preprint, 2015. https://arxiv.org/abs/1509.03025.

Y. Chen, Y. Chi, and A. Goldsmith. Exact and stable covariance estimation from quadratic sampling via convex programming. IEEE Transactions on Information Theory, 61(7):4034–4059, 2015.

Y. Chen, Y. Chi, J. Fan, and C. Ma. Gradient descent with random initialization: Fast global convergence for nonconvex phase retrieval. Mathematical Programming, 176(1-2):5–37, 2019.

Y. Chen, Y. Chi, J. Fan, C. Ma, and Y. Yan. Noisy matrix completion: Understanding statistical guarantees for convex relaxation via nonconvex optimization. SIAM journal on optimization, 30(4):3098–3121, 2020.

D. Cifuentes and A. Moitra. Polynomial time guarantees for the Burer-Monteiro method. preprint, 2019. https://arxiv.org/abs/1912.01745.

C. Criscitiello and N. Boumal. Efficiently escaping saddle points on manifolds. In Advances in Neural Information Processing Systems 32, pages 5987–5997, 2019.

M. A. Davenport and J. Romberg. An overview of low-rank matrix recovery from incomplete observations. IEEE Journal of Selected Topics in Signal Processing, 10(4):608–622, 2016.

L. Demanet and P. Hand. Stable optimizationless recovery from phaseless linear measurements. Journal of Fourier Analysis and Applications, 20(1):199–221, 2012.

L. Ding and Y. Chen. Leave-one-out approach for matrix completion: Primal and dual analysis. IEEE Transactions on Information Theory, 2020.

L. Ding, A. Yurtsever, V. Cevher, J. A. Tropp, and M. Udell. An optimal-storage approach to semidefinite programming using approximate complementarity. preprint, 2019. https://arxiv.org/abs/1902.03373.

N. El Karoui, D. Bean, P. J. Bickel, C. Lim, and B. Yu. On robust regression with high-dimensional predictors. Proceedings of the National Academy of Sciences, 110(36):14557–14562, 2013.

A. Elisseeff and M. Pontil. Leave-one-out error and stability of learning algorithms with applications. In NATO science series, III: Computer and systems sciences, volume 190, pages 111–130. IOS Press, 2003.

M. Fickus, D. G. Mixon, A. A. Nelson, and Y. Wang. Phase retrieval from very few measurements. Linear Algebra and its Applications, 449:475–499, 2014.

C. Gao and A. Y. Zhang. Exact minimax estimation for phase synchronization. preprint, 2020. https://arxiv.org/abs/2010.04345.

R. Ge, J. D. Lee, and T. Ma. Matrix completion has no spurious local minimum. In Advances in Neural Information Processing Systems 29, pages 2973–2981, 2016.

R. Ge, C. Jin, and Y. Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In International Conference on Machine Learning, pages 1233–1242, 2017.

R. Gerchberg and W. Saxton. A practical algorithm for the determination of phase from image and diffraction plane pictures. Optik, 35(2):237–246, 1972.

D. Gross, F. Krahmer, and R. Kueng. A partial derandomization of PhaseLift using spherical designs. Journal of Fourier Analysis and Applications, 21(2):229–266, 2015.

M. Hardt, R. Meka, P. Raghavendra, and B. Weitz. Computational limits for matrix completion. In Conference on Learning Theory, pages 703–725, 2014.

P. Jain, P. Netrapalli, and S. Sanghavi. Low-rank matrix completion using alternating minimization. In Symposium on the Theory of Computing, pages 665–674, 2013.

M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. SIAM Journal on Optimization, 20(5):2327–2351, 2010.

K. Kawaguchi. Deep learning without poor local minima. In Advances in Neural Information Processing Systems 29, pages 586–594, 2016.

R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. IEEE transactions on information theory, 56(6):2980–2998, 2010.

J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht. Gradient descent converges to minimizers. In Proceedings of the Conference on Computational Learning Theory, 2016.

Q. Li and G. Tang. The nonconvex geometry of low-rank matrix optimizations with general objective functions. In 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pages 1235–1239, 2017.

X. Li and V. Voroninski. Sparse signal recovery from quadratic measurements via convex programming. SIAM Journal on Mathematical Analysis, 45(5):3019–3033, 2013.

C. Ma, K. Wang, Y. Chi, and Y. Chen. Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval and matrix completion. In International Conference on Machine Learning, pages 3345–3354, 2018.

M. Mondelli and A. Montanari. Fundamental limits of weak recovery with applications to phase retrieval. Foundations of Computational Mathematics, 19(3):703–773, 2019.

P. Netrapalli, P. Jain, and S. Sanghavi. Phase retrieval using alternating minimization. In Advances in Neural Information Processing Systems 26, pages 1796–2804, 2013.

I. Panageas and G. Piliouras. Gradient descent only converges to minimizers: Non-isolated critical points and invariant regions. In Innovations in Theoretical Computer Science, 2017.

T. Pumir, S. Jelassi, and N. Boumal. Smoothed analysis of the low-rank approach for smooth semidefinite programs. In Advances in Neural Information Processing Systems 31, pages 2283–2292, 2018.

B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM Review, 52(3):471–501, 2010.

Y. Schechtman, Y. C. Eldar, O. Cohen, H. N. Chapman, J. Miao, and M. Segev. Phase retrieval with application to optical imaging: a contemporary overview. IEEE Signal processing magazine, 32(3):87–109, 2015.

J. Sun, Q. Qu, and J. Wright. Complete dictionary recovery over the sphere I: Overview and geometric picture. IEEE Transactions on Information Theory, 63(2), 2017.

J. Sun, Q. Qu, and J. Wright. A geometric analysis of phase retrieval. Foundations of Computational Mathematics, 18(5):1131–1198, 2018.

T. Tong, C. Ma, and Y. Chi. Accelerating ill-conditioned low-rank matrix estimation via scaled gradient descent. to appear in Journal of Machine Learning Research, 2020. https://arxiv.org/abs/2005.08898.

I. Waldspurger. Phase retrieval with random gaussian sensing vectors by alternating projections. IEEE Transactions on Information Theory, 64(5):3301–3312, 2018.

I. Waldspurger and A. Waters. Rank optimality for the Burer-Monteiro factorization. SIAM journal on Optimization, 30(3):2577–2602, 2020.

I. Waldspurger, A. d'Aspremont, and S. Mallat. Phase recovery, maxcut and complex semidefinite programming. Mathematical Programming, 149(1-2):47–81, 2015.

G. Wang, G. B. Giannakis, and Y. C. Eldar. Solving random systems of quadratic equations via truncated amplitude flow. IEEE Transactions on Information Theory, 64(2):773–794, 2018.

A. Yurtsever, J. A. Tropp, O. Fercoq, M. Udell, and V. Cevher. Scalable semidefinite programming. preprint, 2019. https://arxiv.org/abs/1912.02949.

H. Zhang and Y. Liang. Reshaped Wirtinger flow for solving quadratic systems of equations. In Advances in Neural Information Processing Systems 29, 2016.

T. Zhang. Phase retrieval by alternating minimization with random initialization. IEEE Transactions on Information Theory, 2020.

Y. Zhang, Q. Qu, and J. Wright. From symmetry to geometry: Tractable nonconvex problems. preprint, 2020. https://arxiv.org/abs/2007.06753.

T. Zhao, Z. Wang, and H. Liu. A nonconvex optimization framework for low rank matrix estimation. In Advances in Neural Information Processing Systems 28, pages 559–567, 2015.

Q. Zheng and J. Lafferty. Convergence analysis for rectangular matrix completion using Burer-Monteiro factorization and gradient descent. preprint, 2016. https://arxiv.org/abs/1605.07051.

Y. Zhong and N. Boumal. Near-optimal bounds for phase synchronization. SIAM Journal on Optimization, 28(2):989–1016, 2018.

A Proof of Proposition 5.2

For all k,

$$z^{obj*}Cz^{obj} = \sum_{l,l'\neq k} C_{l,l'}\overline{z_l^{obj}}z_{l'}^{obj} + 2\Re\left(\overline{z_k^{obj}}\sum_{l'\neq k} C_{k,l'}z_{l'}^{obj}\right) + 1.$$

As $z^{obj*}Cz^{obj} = \max_{|z_1|=\cdots=|z_n|=1} z^*Cz$, we must have

$$\begin{aligned}\sum_{l,l'\neq k} C_{l,l'}\overline{z_l^{obj}}z_{l'}^{obj} + 2\Re\left(\overline{z_k^{obj}}\sum_{l'\neq k} C_{k,l'}z_{l'}^{obj}\right) + 1 \\ = \max_{\theta\in\mathbb{R}}\left(\sum_{l,l'\neq k} C_{l,l'}\overline{z_l^{obj}}z_{l'}^{obj} + 2\Re\left(e^{-i\theta}\sum_{l'\neq k} C_{k,l'}z_{l'}^{obj}\right) + 1\right),\end{aligned}$$

which is equivalent to the existence of some $\lambda_k \in \mathbb{R}^+$ such that

$$\sum_{l'\neq k} C_{k,l'}z_{l'}^{obj} = \lambda_k z_k^{obj}.$$

Therefore,

$$(Cz^{obj})_k = z_k^{obj} + \sum_{l'\neq k} C_{k,l'}z_{l'}^{obj} = (1+\lambda_k)z_k^{obj},$$

which implies

$$\begin{aligned}\mathcal{P}(Cz^{obj})_k &= \frac{(1+\lambda_k)z_k^{obj}}{\left|(1+\lambda_k)z_k^{obj}\right|} \\ &= z_k^{obj}.\end{aligned}$$