Convex and non-convex algorithms for phase retrieval

Irène Waldspurger

CNRS and CEREMADE (Université Paris Dauphine)
Équipe MOKAPLAN (INRIA)

December 4, 2019

Séminaire Cosmostat

Saclay

Phase retrieval problems : definition

Reconstruct $x \in E$ from $(|L_i(x)|)_{i \in I}$?

Here,

- E is a complex vector space ;
- $(L_i)_{i \in I}$ is a known family of linear forms $(E \to \mathbb{C})$;
- $|.|$ is the standard complex modulus.

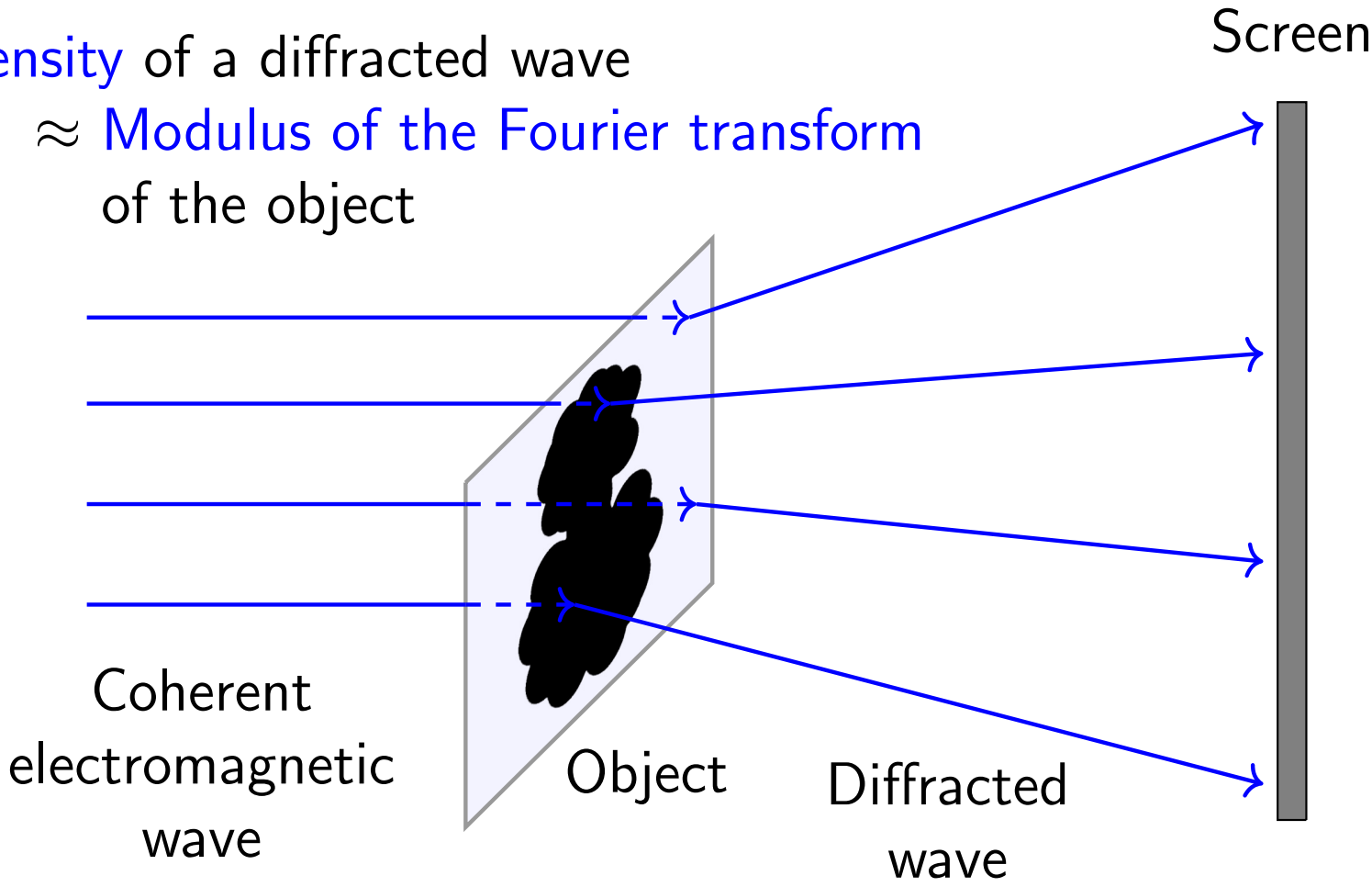
Remark : reconstruction is only up to a global phase,

$$\forall \phi \in \mathbb{R}, \quad |L_i(x)| = |L_i(e^{i\phi} x)|.$$

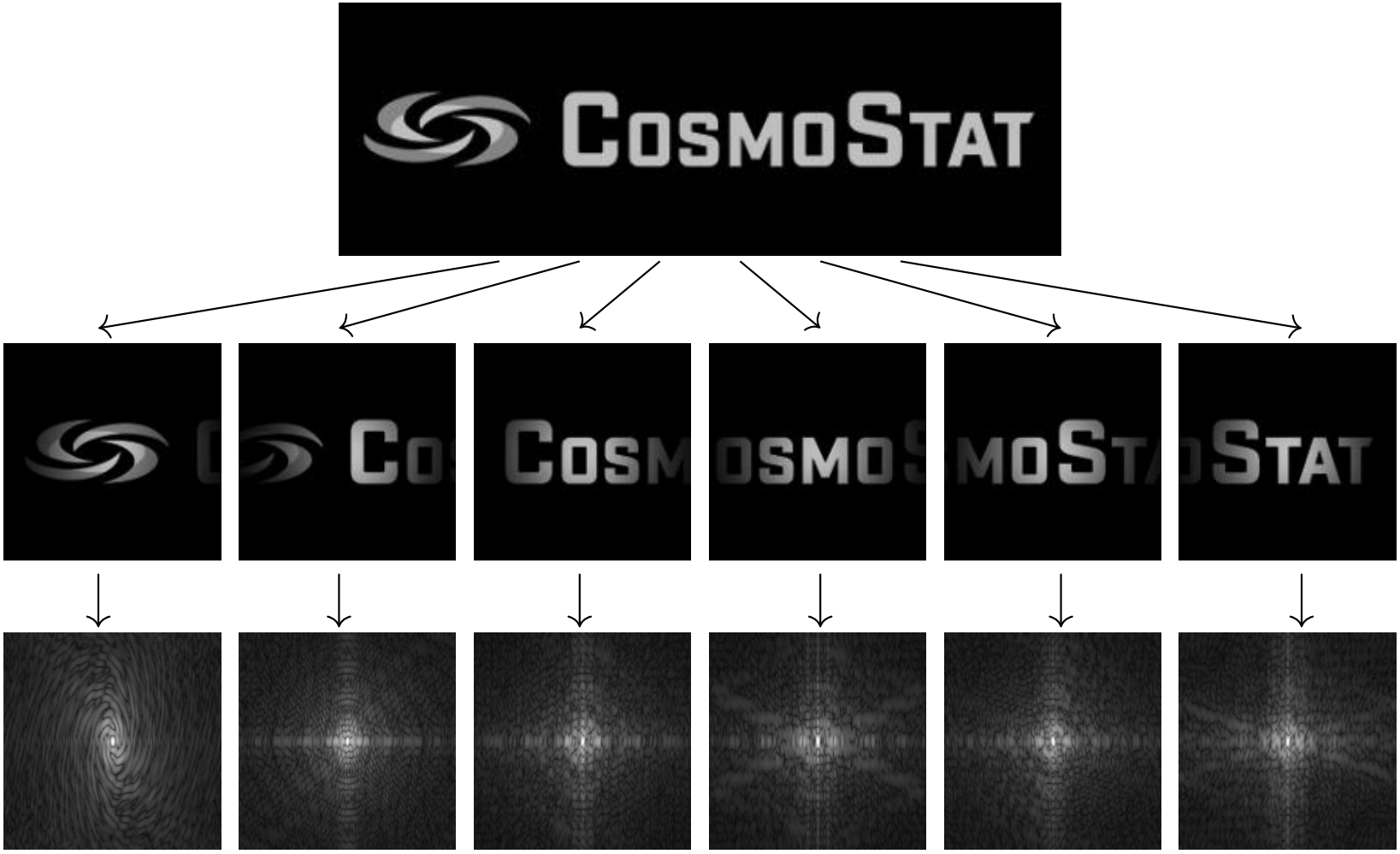
Applications in imaging

[Schechtman, Eldar, Cohen, Chapman, Miao, and Segev, 2015]

Intensity of a diffracted wave
$\approx$ Modulus of the Fourier transform
of the object



Masked Fourier transform / ptychography



Outline of the talk

- **Non-convex algorithms ('50s $\rightarrow$ today)**
  - Description of the most classical one, alternating projections
  - Numerical example
  - Theoretical correctness guarantees in a random setting

Outline of the talk

- **Non-convex algorithms ('50s → today)**
  - Description of the most classical one, alternating projections
  - Numerical example
  - Theoretical correctness guarantees in a random setting
- **Convexified algorithms (2011 → today)**
  - Definition
  - Main advantage : better reconstruction quality, especially in more structured settings
  - Main drawback : high computational cost
    → Possible improvements ?

Definition of alternating projections

Gerchberg and Saxton [1972]

Reconstruct $x \in E$ from $|\mathcal{L}(x)| \stackrel{def}{=} (|L_i(x)|)_{i \in I}$?

$\Longleftrightarrow$

Find y such that $y \in \mathrm{Range}\,(\mathcal{L})$
and $|y| = |\mathcal{L}(x_{true})|$.

Definition of alternating projections

Gerchberg and Saxton [1972]

Reconstruct $x \in E$ from $|\mathcal{L}(x)| \stackrel{def}{=} (|L_i(x)|)_{i \in I}$?

$\Longleftrightarrow$

Find y such that $y \in \mathrm{Range}\,(\mathcal{L})$
and $|y| = |\mathcal{L}(x_{true})|$.

- Choose an initial guess for y.

Definition of alternating projections

Gerchberg and Saxton [1972]

Reconstruct $x \in E$ from $|\mathcal{L}(x)| \stackrel{def}{=} (|L_i(x)|)_{i \in I}$?

$\Longleftrightarrow$

Find y such that $y \in \mathrm{Range}\,(\mathcal{L})$
and $|y| = |\mathcal{L}(x_{true})|$.

- Choose an initial guess for y.
- Project onto $\mathrm{Range}\,(\mathcal{L})$.

Definition of alternating projections

Gerchberg and Saxton [1972]

Reconstruct $x \in E$ from $|\mathcal{L}(x)| \stackrel{def}{=} (|L_i(x)|)_{i \in I}$?

$\Longleftrightarrow$

Find y such that $y \in \mathrm{Range}\,(\mathcal{L})$
and $|y| = |\mathcal{L}(x_{true})|$.

- Choose an initial guess for y.
- Project onto $\mathrm{Range}\,(\mathcal{L})$.
- Project onto $\{z \text{ s.t. } |z| = |\mathcal{L}(x_{true})|\}$.

Definition of alternating projections

Gerchberg and Saxton [1972]

Reconstruct $x \in E$ from $|\mathcal{L}(x)| \stackrel{def}{=} (|L_i(x)|)_{i \in I}$?

$\Longleftrightarrow$

Find y such that $y \in \mathrm{Range}\,(\mathcal{L})$
and $|y| = |\mathcal{L}(x_{true})|$.

- Choose an initial guess for y.
- Project onto $\mathrm{Range}\,(\mathcal{L})$.
- Project onto $\{z \text{ s.t. } |z| = |\mathcal{L}(x_{true})|\}$.

Repeat the double projection

Definition of alternating projections

Gerchberg and Saxton [1972]

Reconstruct $x \in E$ from $|\mathcal{L}(x)| \stackrel{def}{=} (|L_i(x)|)_{i \in I}$?

$\Longleftrightarrow$

Find y such that $y \in \mathrm{Range}\,(\mathcal{L})$
and $|y| = |\mathcal{L}(x_{true})|$.

- Choose an initial guess for y.
- Project onto $\mathrm{Range}\,(\mathcal{L})$.
- Project onto $\{z \text{ s.t. } |z| = |\mathcal{L}(x_{true})|\}$.
- Hope it converges to $\mathcal{L}(x_{true})$.

Repeat the double projection

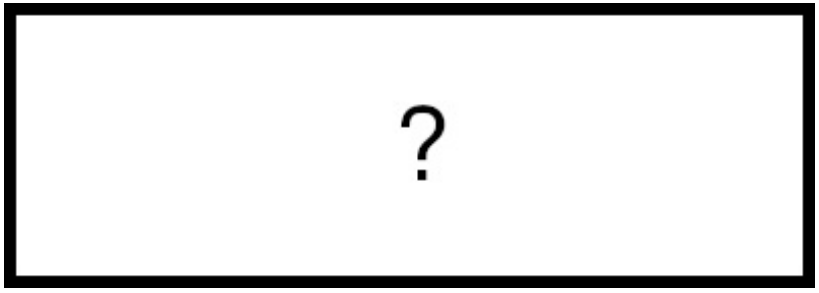
Advantages :

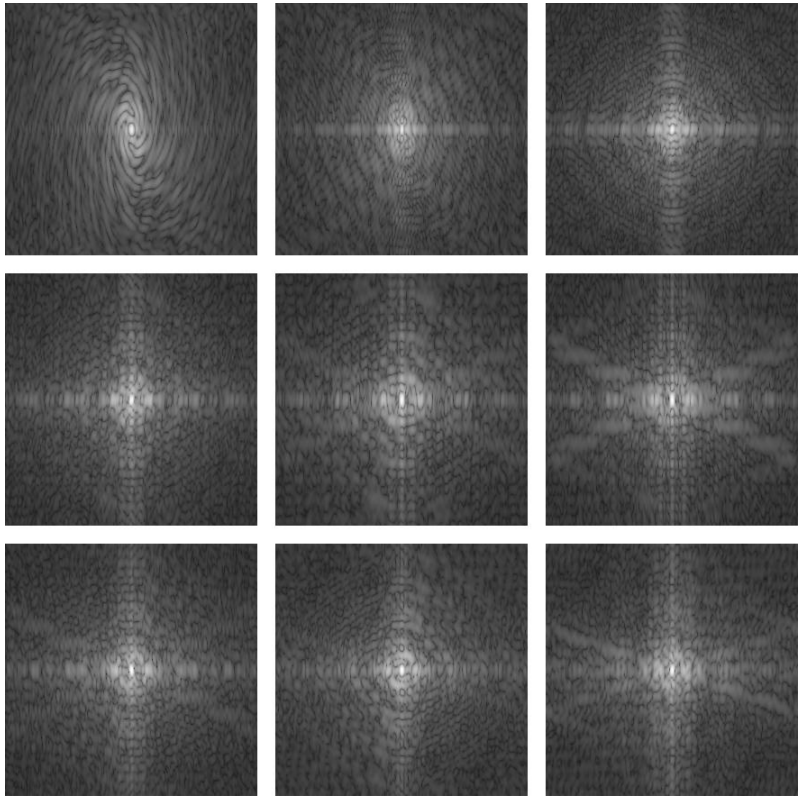
- Fast.
- Extremely easy to implement.
- Easily incorporates additional constraints.

Issue :

- May fail by getting stuck at a “critical point”.

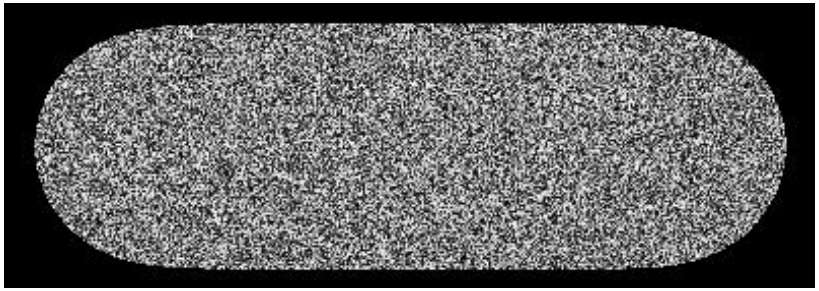
Numerical example (1)





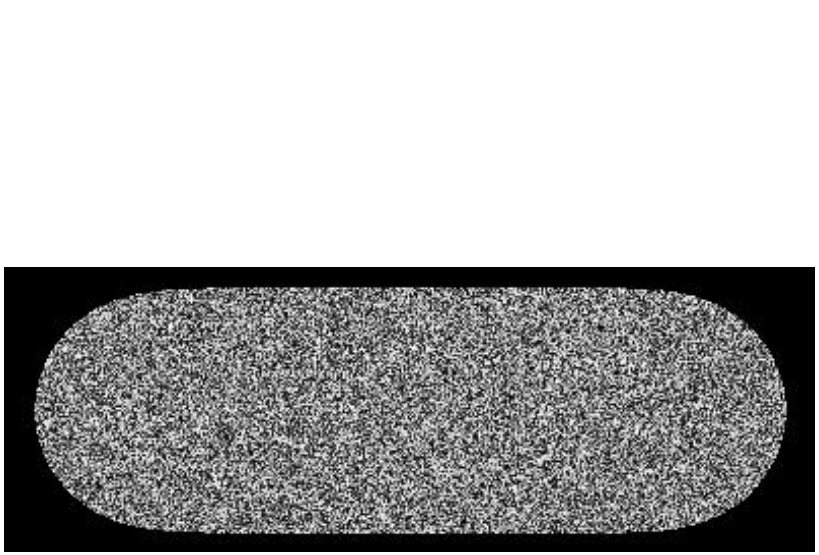
Target modulus of the masked Fourier transform

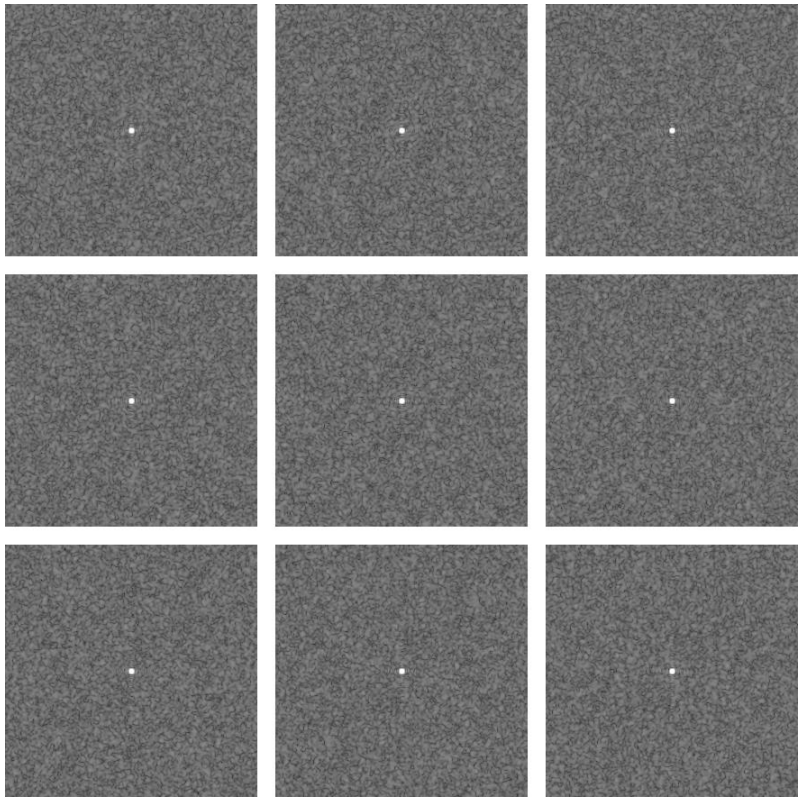
Numerical example (1)



Random initial guess

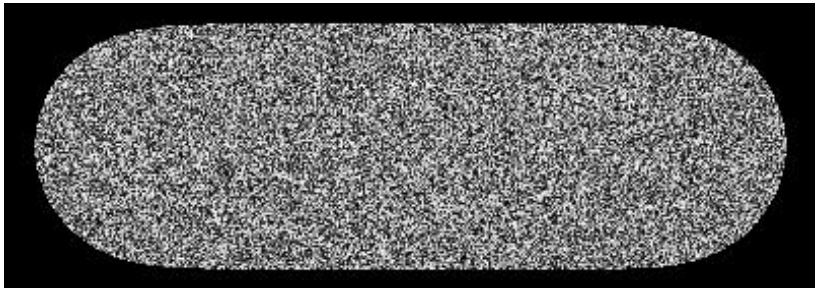
Numerical example (1)

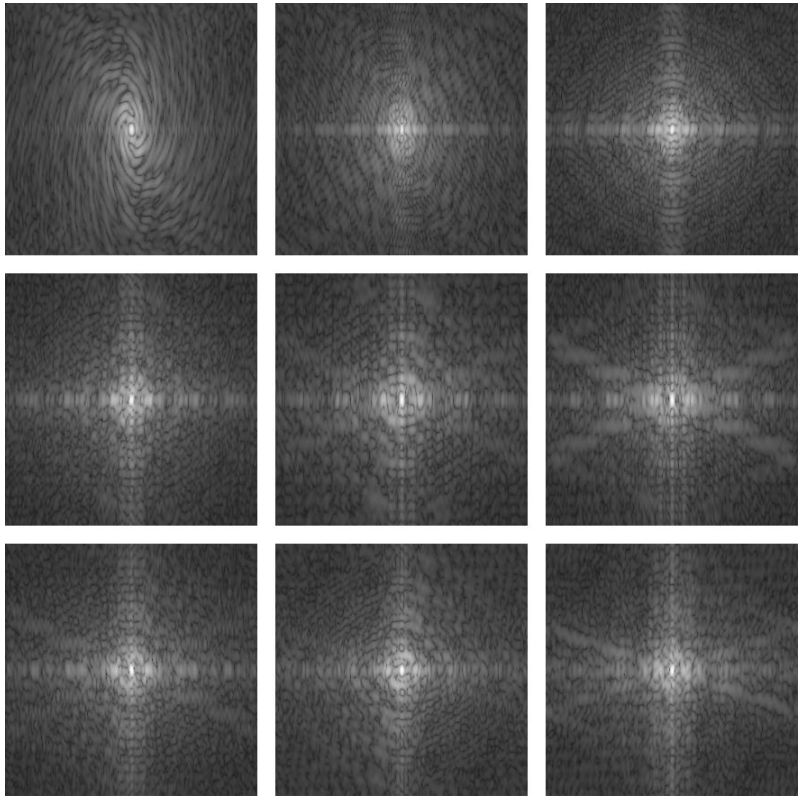




Masked Fourier transform

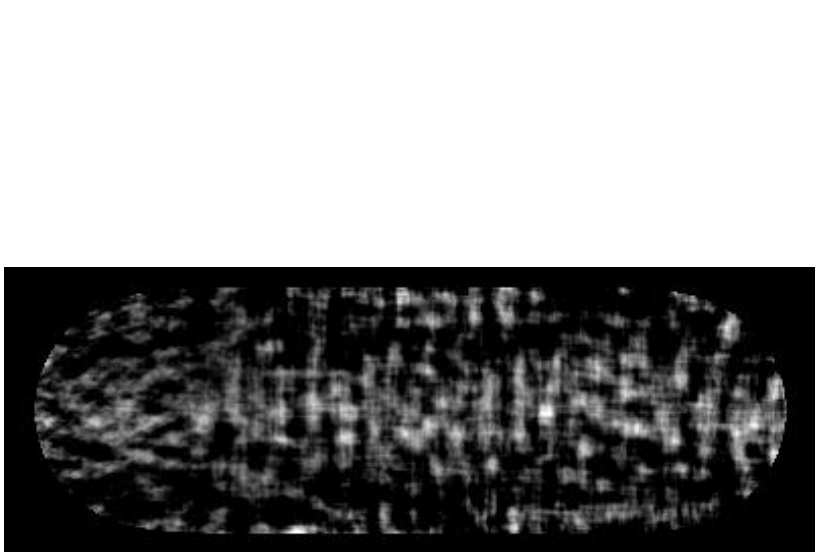
Numerical example (1)

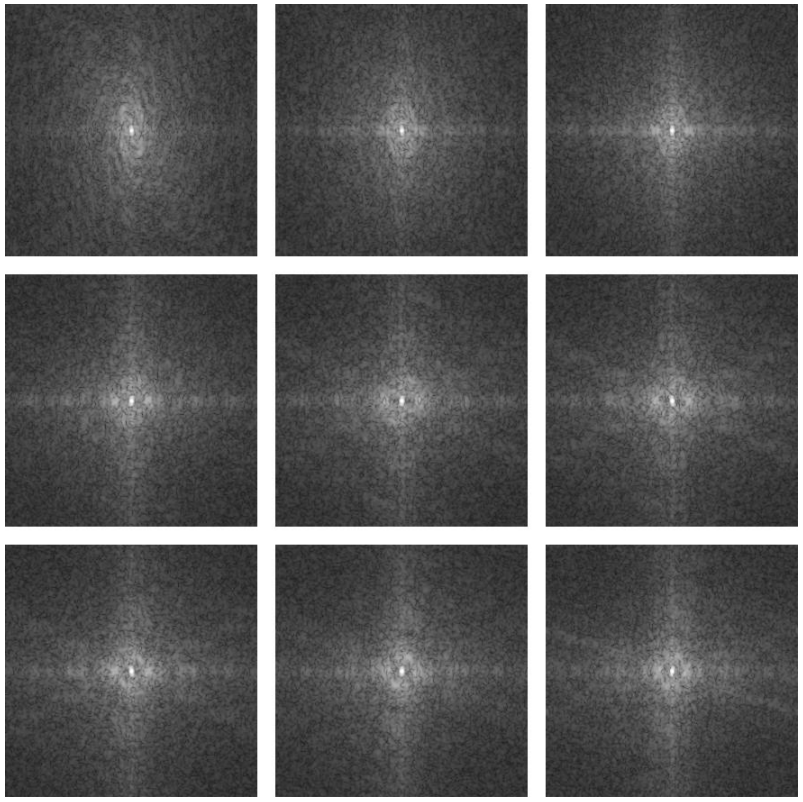




Projection on the modulus constraint

Numerical example (1)

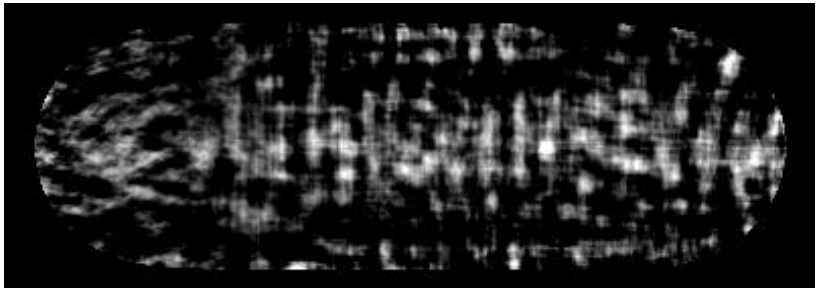


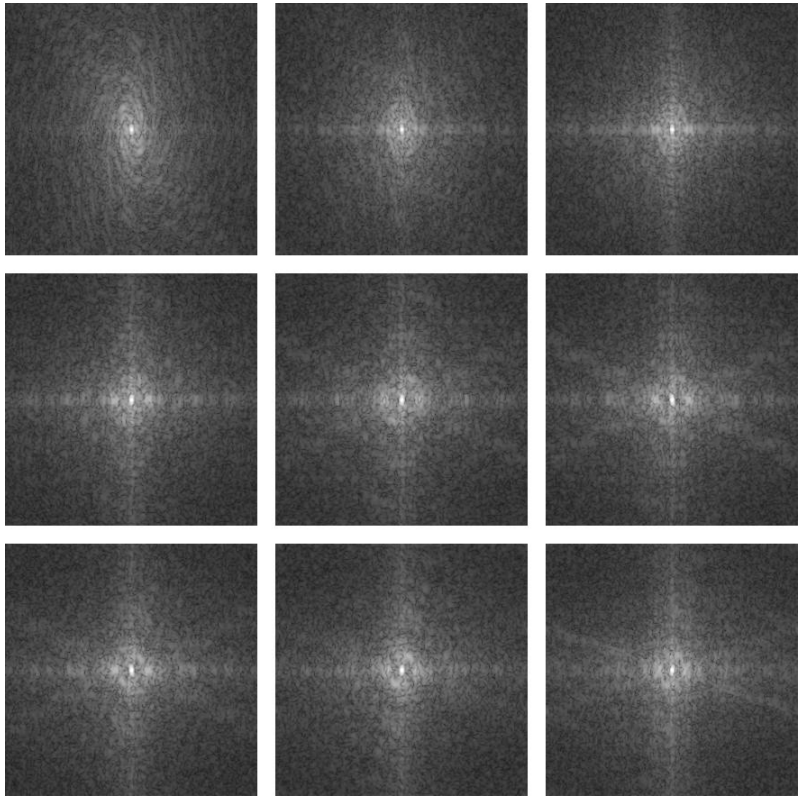


Projection on the range

Iteration 1

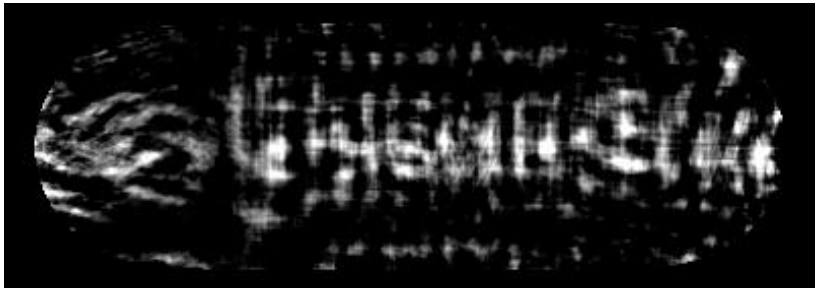
Numerical example (1)

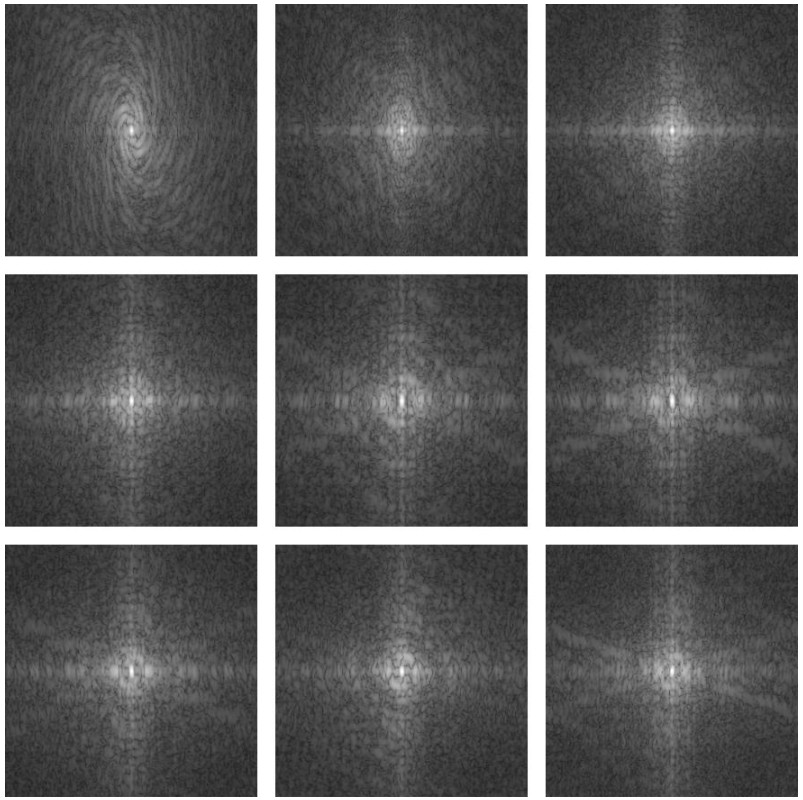




Iteration 2

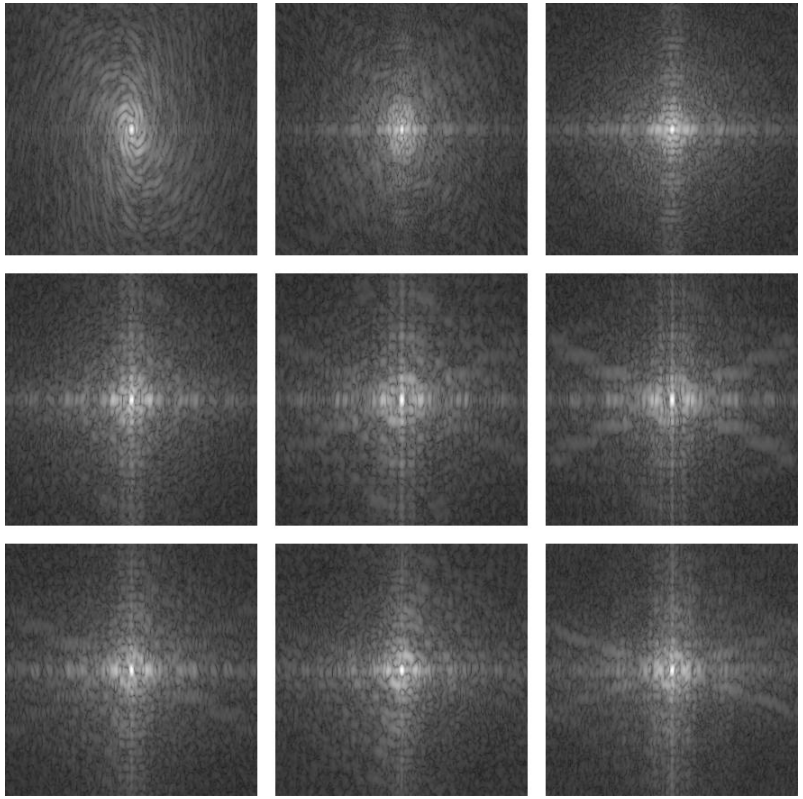
Numerical example (1)





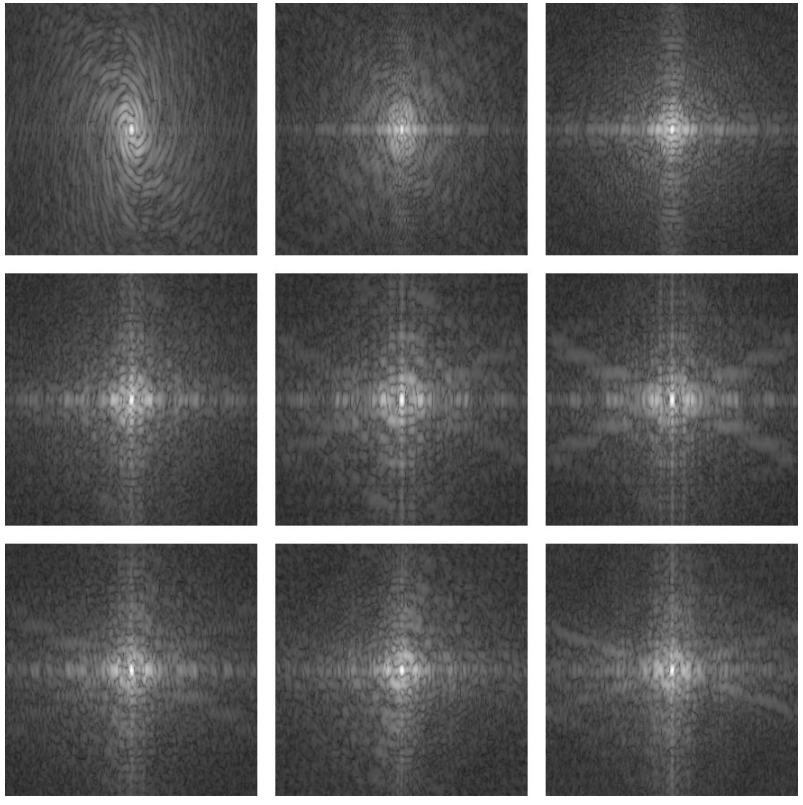
Iteration 10

Numerical example (1)



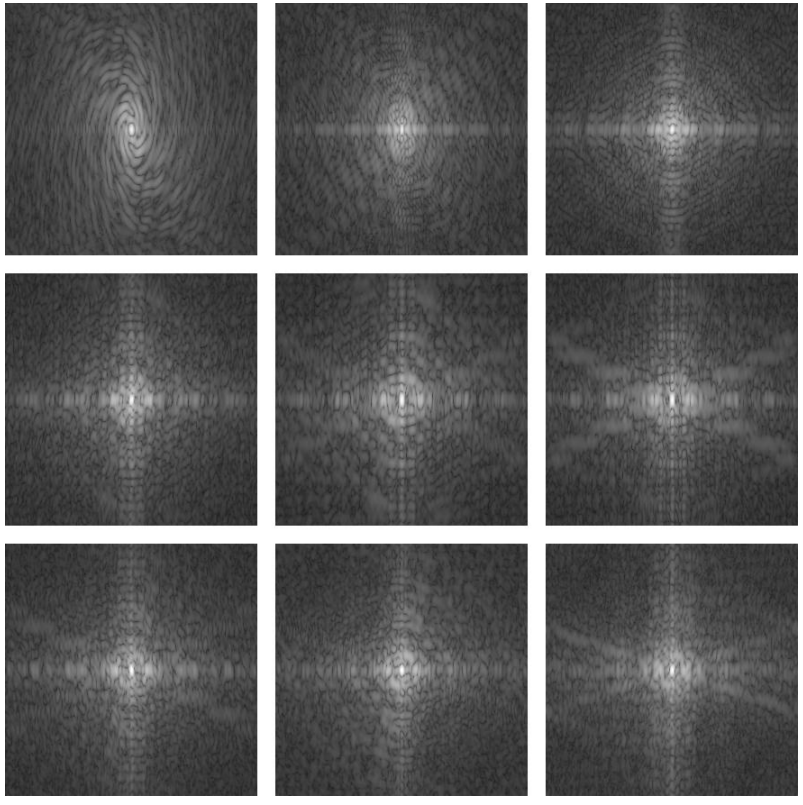
Iteration 50

Numerical example (1)



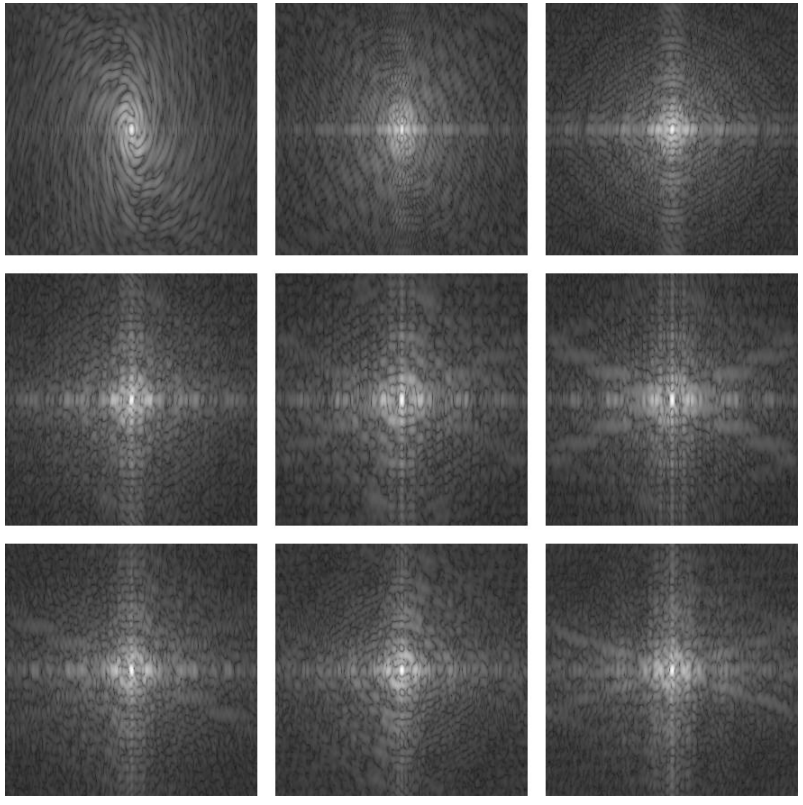
Iteration 100

Numerical example (1)



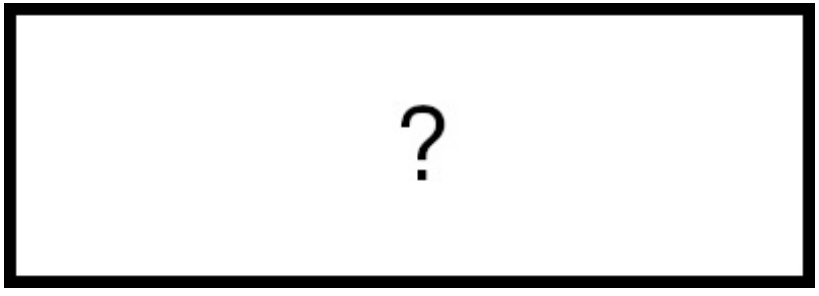
Iteration 200

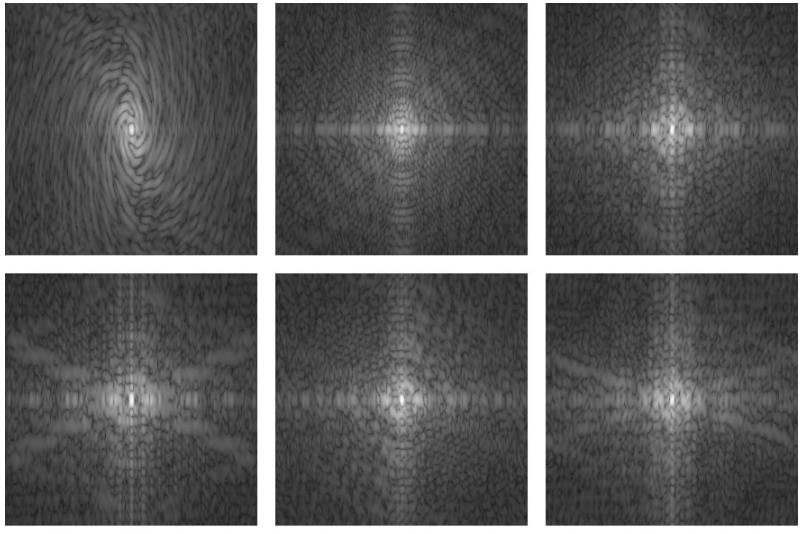
Numerical example (1)



Iteration 400

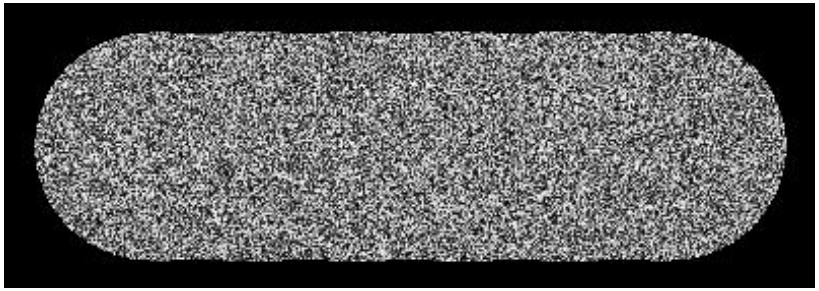
Numerical example (2)

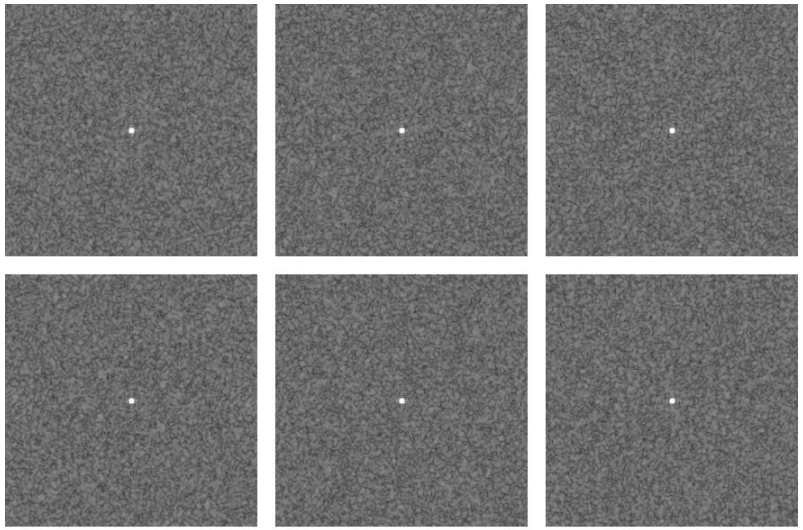




Target modulus of the masked Fourier transform

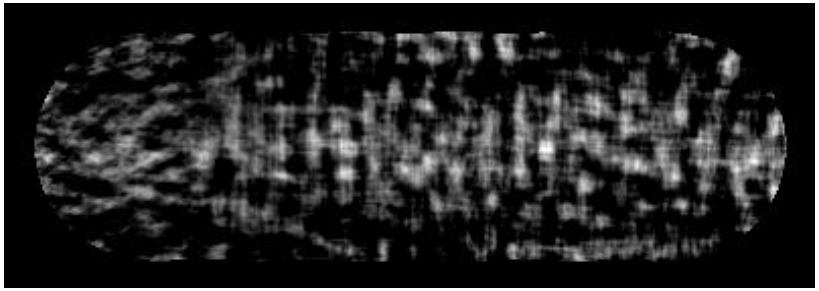
Numerical example (2)

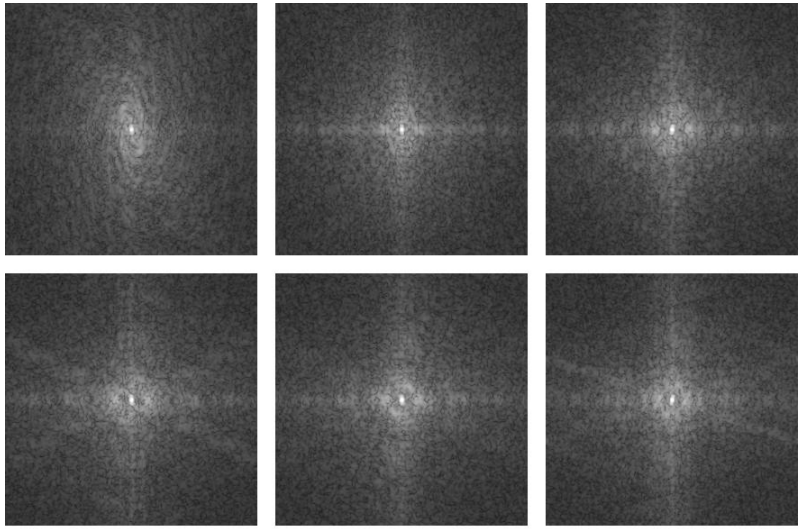




Random initial guess

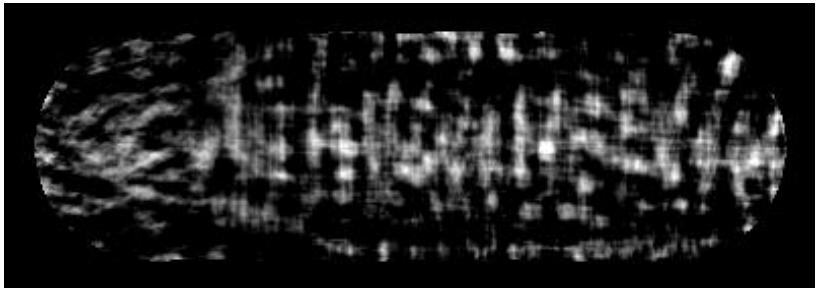
Numerical example (2)

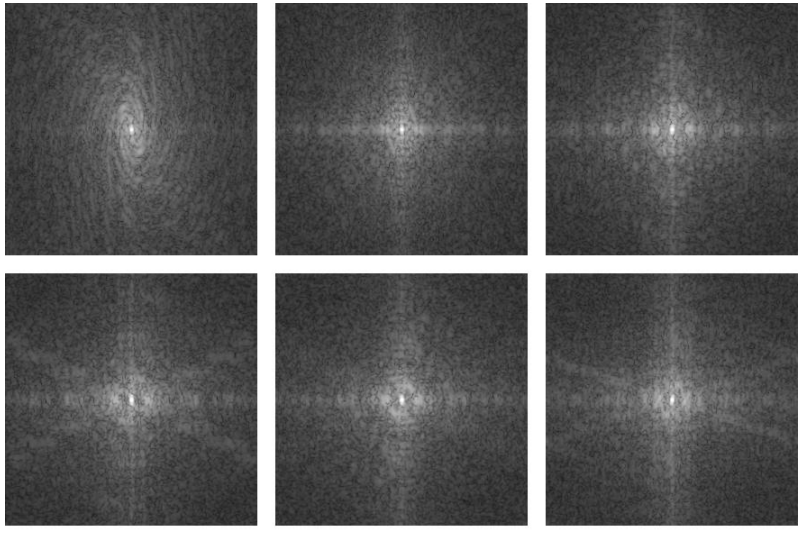




Iteration 1

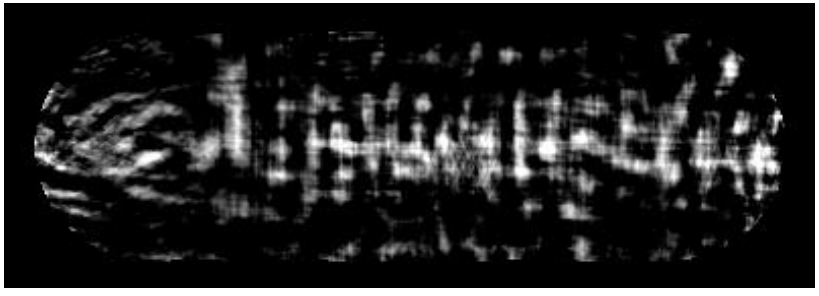
Numerical example (2)

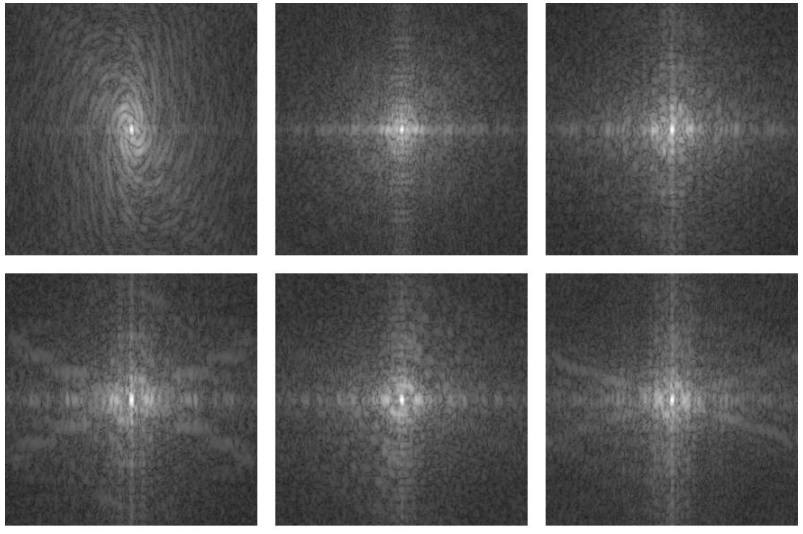




Iteration 2

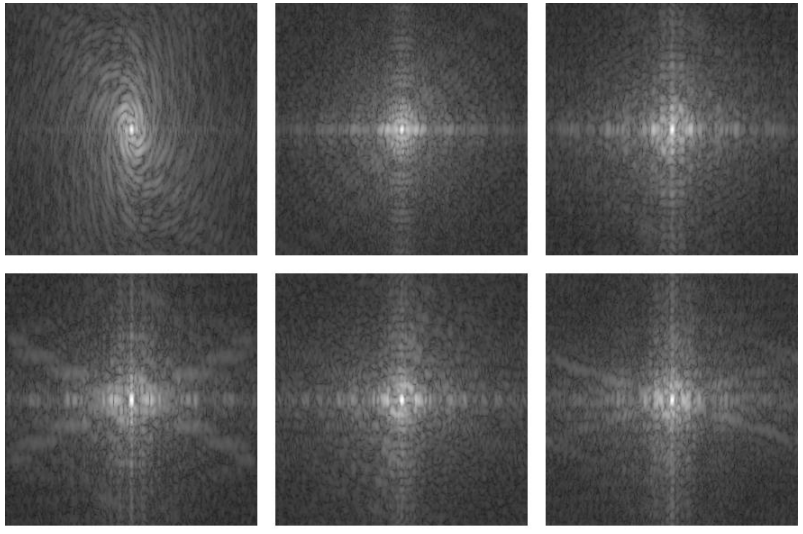
Numerical example (2)





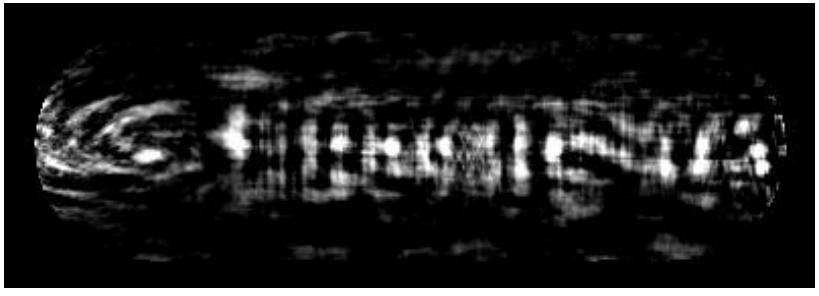
Iteration 10

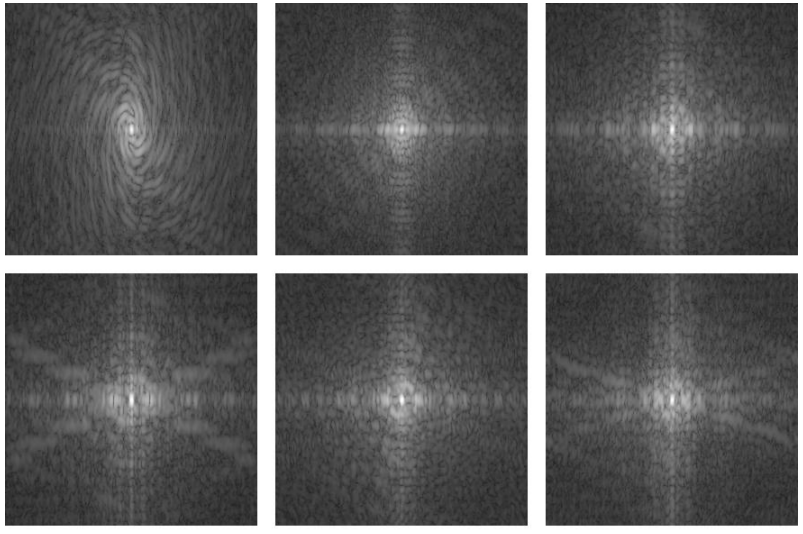
Numerical example (2)



Iteration 50

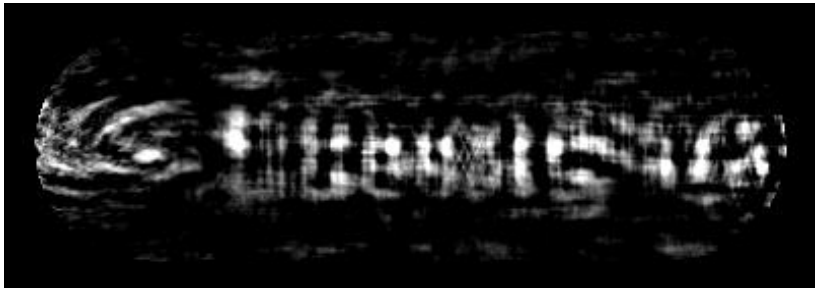
Numerical example (2)

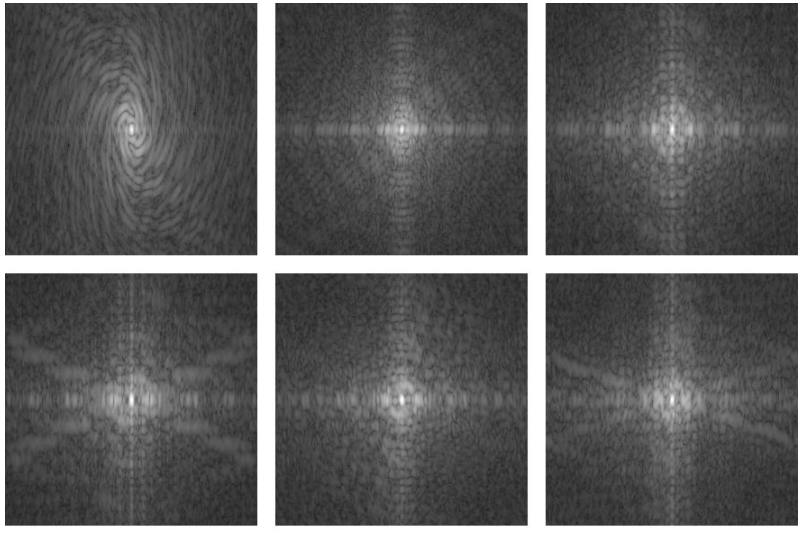




Iteration 100

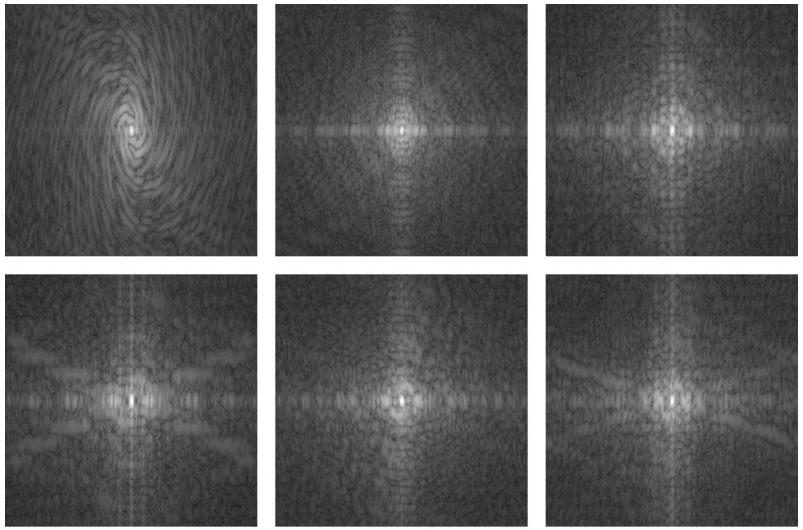
Numerical example (2)





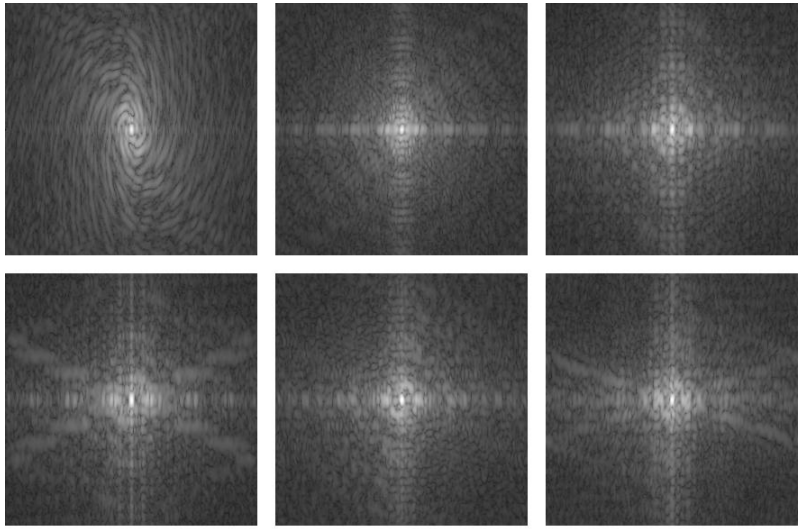
Iteration 200

Numerical example (2)



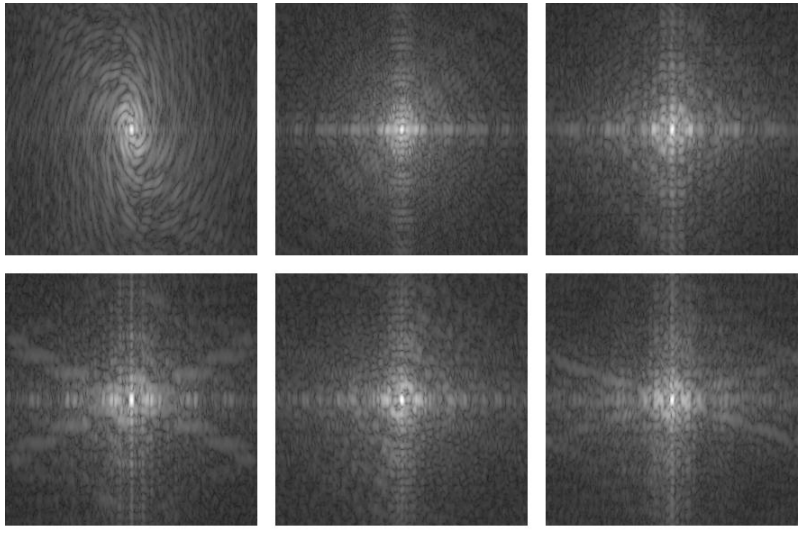
Iteration 400

Numerical example (2)



Iteration 1000

Numerical example (2)



Iteration 5000

Failure situations exist, but there are also many situations where the algorithm performs well.

We are completely unable to characterize success / failure situations.

Very vague intuition : When there is enough "redundancy" in the measurements, it more or less works.

Can we make this statement formal, at least in a simple setting ?

Simple theoretical setting

We consider the finite-dimensional setting : $E = \mathbb{C}^n$.

There are m linear forms $L_i = \langle v_i, . \rangle$, with

$$\forall 1 \leq i \leq m, \quad v_i \overset{iid}{\sim} \mathcal{N}(0, I_n).$$

The "interesting" regime is when m is proportional to n.

Theorem (Waldspurger [2017])

We assume that

- $m \geq Cn$, with C large enough ;
- the initial guess for y is chosen as in [Chen and Candès, 2015].

Then, with high probability, the sequence $(y_k)_{k\in\mathbb{N}}$ generated by alternating projections satisfies

$$y_k \overset{k\to+\infty}{\to} \mathcal{L}(x_{true})$$

(and the convergence speed is exponential).

Related work

[Netrapalli, Jain, and Sanghavi, 2013] : same theorem for a resampled (hence unrealistic) version of the algorithm.

Similar results for other non-convex algorithms (gradient descent over various cost functions).
[Chen and Candès, 2015]
[Zhang and Liang, 2016]
[Wang, Giannakis, and Eldar, 2017] ...

Specific interest of the previous theorem :

- Alternating projections (and variants) is the most widely-used phase retrieval algorithm ;
- Different behavior from other algorithms.
  (Because it is non-smooth ?)

We have described a situation where alternating projections provably works.

But recall that there are situations where it fails.

$\rightarrow$ Other algorithms to handle these situations ?

We have described a situation where alternating projections provably works.

But recall that there are situations where it fails.
$\rightarrow$ Other algorithms to handle these situations ?

In the rest of the talk, we describe another family of phase retrieval methods : convexified algorithms.

[Recht, Fazel, and Parrilo, 2010]
[Candès, Strohmer, and Voroninski, 2013]

$$\text{Find } x \in \mathbb{C}^n \quad \text{s.t. } \forall i,\ |\langle v_i, x\rangle| = b_i \iff \text{Find } x \in \mathbb{C}^n \quad \text{s.t. } \forall i,\ x^* v_i v_i^* x = b_i^2$$

$$\Updownarrow$$

$$\text{Find } X \in \mathbb{C}^{n\times n} \quad \text{s.t. } \forall i,\ \mathrm{Tr}(v_i v_i^* X) = b_i^2,\ \mathrm{rank}(X) = 1 \iff \text{Find } x \in \mathbb{C}^n \quad \text{s.t. } \forall i,\ \mathrm{Tr}(v_i v_i^* x x^*) = b_i^2$$

Find $x \in \mathbb{C}^n$
s.t. $\forall i,\ |\langle v_i, x\rangle| = b_i$

$\Longleftrightarrow$

Find $x \in \mathbb{C}^n$
s.t. $\forall i,\ x^* v_i v_i^* x = b_i^2$

$\Updownarrow$

Find $X \in \mathbb{C}^{n\times n}$
s.t. $\forall i,\ \mathrm{Tr}(v_i v_i^* X) = b_i^2$
$\mathrm{rank}(X) = 1$

$\Longleftrightarrow$

Find $x \in \mathbb{C}^n$
s.t. $\forall i,\ \mathrm{Tr}(v_i v_i^* x x^*) = b_i^2$

Not convex

$$\text{Find } x \in \mathbb{C}^n \quad \text{s.t. } \forall i,\ |\langle v_i, x\rangle| = b_i \iff \text{Find } x \in \mathbb{C}^n \quad \text{s.t. } \forall i,\ x^* v_i v_i^* x = b_i^2$$

$$\Updownarrow$$

$$\text{Find } X \in \mathbb{C}^{n\times n} \quad \text{s.t. } \forall i,\ \mathrm{Tr}(v_i v_i^* X) = b_i^2,\ \cancel{\mathrm{rank}(X) = 1} \iff \text{Find } x \in \mathbb{C}^n \quad \text{s.t. } \forall i,\ \mathrm{Tr}(v_i v_i^* x x^*) = b_i^2$$

$$\xrightarrow{\text{Approximation}} \quad \min\ \mathrm{Tr}\, X \quad \text{s.t. } \forall i,\ \mathrm{Tr}(v_i v_i^* X) = b_i^2,\ X \succeq 0$$

$$\text{Find } x \in \mathbb{C}^n \quad \text{s.t. } \forall i,\ |\langle v_i, x\rangle| = b_i \iff \text{Find } x \in \mathbb{C}^n \quad \text{s.t. } \forall i,\ x^* v_i v_i^* x = b_i^2$$

$$\Updownarrow$$

$$\text{Find } X \in \mathbb{C}^{n\times n} \quad \text{s.t. } \forall i,\ \mathrm{Tr}(v_i v_i^* X) = b_i^2,\ \cancel{\mathrm{rank}(X) = 1} \iff \text{Find } x \in \mathbb{C}^n \quad \text{s.t. } \forall i,\ \mathrm{Tr}(v_i v_i^* x x^*) = b_i^2$$

$$\xrightarrow{\text{Approximation}} \quad \min\ \mathrm{Tr}\, X \quad \text{s.t. } \forall i,\ \mathrm{Tr}(v_i v_i^* X) = b_i^2,\ X \succeq 0 \qquad \text{(Convex)}$$

This is the *PhaseLift* algorithm.
[Candès, Strohmer, and Voroninski, 2013]

Correctness guarantees

Theorem (Candès and Li [2014])

We assume that $m \geq Cn$ with C large enough.

Then, with high probability,

$$X_{PhaseLift} = x_{true} x_{true}^*,$$

hence *PhaseLift* allows to recover x_{true} .

Another convexified algorithm : *PhaseCut*

[Waldspurger, d'Aspremont, and Mallat, 2015]

Difference with *PhaseLift* : change of variable,

$$\text{recover } \mathcal{L}(x) \quad \text{vs} \quad \text{recover } x.$$

Correctness guarantees : essentially the same as *PhaseLift*.

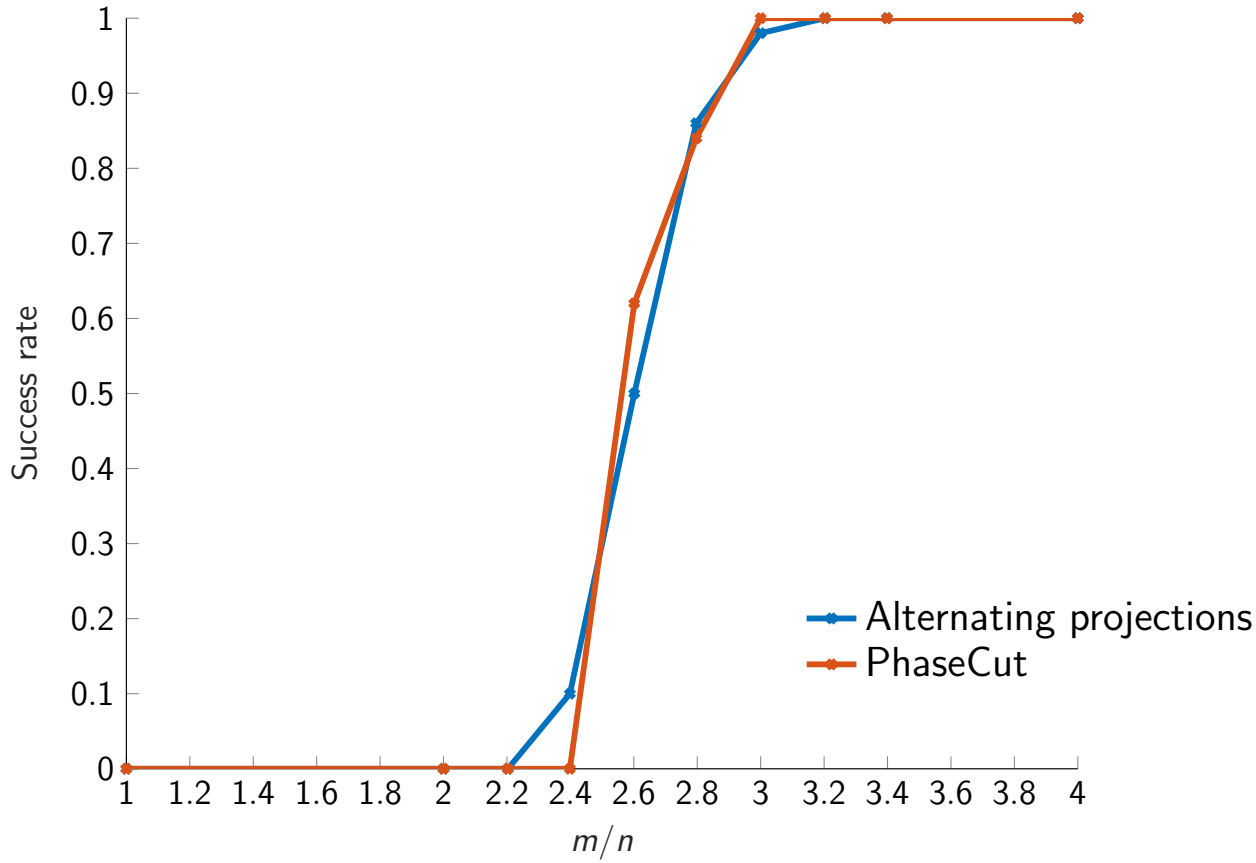
Computational advantage : *PhaseCut* is an instance of a *MaxCut* problem.

$\rightarrow$ Particular structure, which can be used to speed up the solver.

Numerical results for $n = 128$ (1)

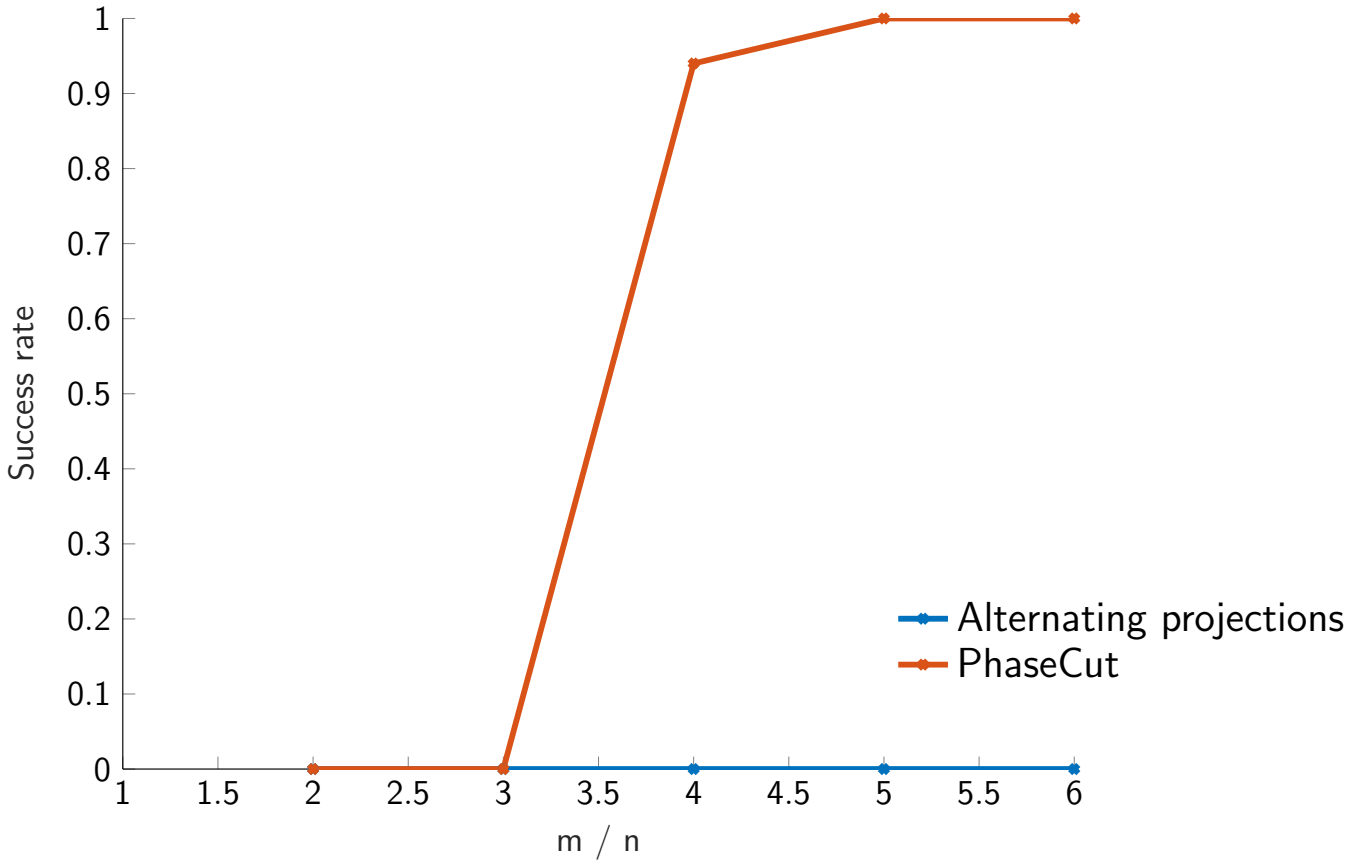
Independent, normally distributed measurement vectors.

"Smart" initialization for alternating projections.



Numerical results for $n = 128$ (2)

(Variant of) masked Fourier transform.



Computational cost

It numerically seems that, in some settings, convexified algorithms can solve problems on which non-convex ones fail.

Main issue : the non-convex problem has dimension n, while the convex problem has dimension n^2.

$$\begin{aligned} \min \ & \operatorname{Tr} X \\ \text{s.t. } \forall i, \ & \operatorname{Tr}(v_i v_i^* X) = b_i^2 \\ & X \succeq 0 \end{aligned}$$

Future work : Burer-Monteiro heuristic

Faster solvers, using the specific properties of the problem ?

$$\begin{aligned} \min \ & \operatorname{Tr} X \\ \text{s.t. } \forall i, \ & \operatorname{Tr}(v_i v_i^* X) = b_i^2 \\ & X \succeq 0 \end{aligned}$$

Specific property : the minimizer has rank 1.

$\rightarrow$ Burer-Monteiro heuristic :

$$\text{write } X = UU^*,$$

with $U \in \mathbb{C}^{n \times p}$, $p \geq 1$.

Optimize over U instead of X.

$\rightarrow$ $O(np)$ variables instead of $O(n^2)$, but not convex.

Future work : Burer-Monteiro heuristic

If $p \gtrsim \sqrt{n}$, the Burer-Monteiro formulation is solvable, despite being non-convex.

$\rightarrow$ $O(n^{1.5})$ variables instead of $O(n^2)$
(better but still too much).

[Boumal, Voroninski, and Bandeira, 2018]

Future work : Burer-Monteiro heuristic

If $p \gtrsim \sqrt{n}$, the Burer-Monteiro formulation is solvable, despite being non-convex.

$\rightarrow$ $O(n^{1.5})$ variables instead of $O(n^2)$ (better but still too much).

[Boumal, Voroninski, and Bandeira, 2018]

Can we take $p = O(1)$?

- For some pathological problems, $p \gtrsim n^{0.5}$ is necessary. [Waldspurger and Waters, 2018]
- According to preliminary numerical experiments, $p = 2$ works fine in most situations.

Future work : Burer-Monteiro heuristic

If $p \gtrsim \sqrt{n}$, the Burer-Monteiro formulation is solvable, despite being non-convex.

$\rightarrow$ $O(n^{1.5})$ variables instead of $O(n^2)$
(better but still too much).

[Boumal, Voroninski, and Bandeira, 2018]

Can we take $p = O(1)$?

- For some pathological problems, $p \gtrsim n^{0.5}$ is necessary.
  [Waldspurger and Waters, 2018]
- According to preliminary numerical experiments, $p = 2$ works fine in most situations.
  Why ?
  Turn it to a practical algorithm ?

Summary

- **Non-convex algorithms**
  - They work well in various situations.
  - Their correctness can be proved in simple settings.
  - But their are situations where they fail.
- **Convexified algorithms**
  - They work in situations where non-convex methods fail.
  - But their computational cost is prohibitive.
  - Future work : lower the complexity with the Burer-Monteiro heuristic ?

Thank you !