Contents

Dedication **1**

Abstract **3**

Acknowledgements **5**

Acronyms and List of Special Symbols **7**

1 Introduction **9**
1.1 Related work 9

2 Wirtinger Flow **11**
2.1 Problem Formulation 11
2.2 Difficulties 11

3 Deep Unfolding **13**

4 UNROLLED WIRTINGER FLOWS **15**

5 Results **17**

Bibliography **19**

TO THE MEMORY OF MOHAMMAD MAHDI ELYASI
A DOWN TO EARTH TRUE PHYSICIST AND A LEGENDARY PETROLHEAD

Abstract

Physical measurements in settings where light, electrons, and similar existences are involved will result in a phase loss in the mathematical formulation. The phenomenon is called *Phase Problem* and the methods developed to tackle the said unpleasantness are called *Phase Retrieval* methods. Due to its presence in a wide spectrum of applications ranging from X-ray crystallography, transmission electron microscopy to quantum mechanics, the retrieval methods are highly investigated and coveted. One of the contemporary breakthroughs are the *Wirtinger Flow* variants which are nice and relatively easy algorithms with small memory footprints equipped with nice guarantees on the solutions. Wirtinger Flows are derived from minimizing a certain functional and are of iterative nature. Like most iterative approaches they are certain parameters that need to be fixed that greatly influence the convergence rate and stability. We aimed to optimized these parameters using *Deep Unfolding* which is an emerging technic from the data-driven side of approaches. Deep Unfolding is basically unfolding an iterative algorithm finite times and putting it into a neural network to be trained. Anyone who has been exposed to machine learning would realize that *Hyperparameter* optimization must be done to close the study which is what we did.

Acknowledgements

1st Order Family Members: Fatemeh Darijani, Nabiollah Darijani, 2nd Order Family Members:

Acronyms and List of Special Symbols

$\in$	belongs to
$\notin$	does not belong to
$\subset, \supset$	inclusion signs
$\mathbb{Q}$	rational field
$<, \leq, >, \geq$	inequality signs
sup	least upper bound
inf	greatest lower bound
$\mathbb{R}$	real field
$+\infty, -\infty, \infty$	infinities
$\overline{z}$	complex conjugate
$\mathrm{Re}(z)$	real part
$\mathrm{Im}(z)$	imaginary part
$\sum$	summation sign
$\mathbb{R}^k$	euclidean k-space
$\mathbf{0}$	null vector
$\boldsymbol{x} \cdot \boldsymbol{y}$	inner product
$\lvert\boldsymbol{x}\rvert$	norm of vector $\boldsymbol{x}$
$\{x_n\}$	sequence
$\bigcup, \cup$	union
$\bigcap, \cap$	intersection
(a, b)	segment
$[a, b]$	interval
E^{c}	complement of E
$E^{'}$	limit points of E
$\overline{E}$	closure of E
lim	limit
$\rightarrow$	converges to
$\limsup$	lim sup
$\liminf$	lim inf
$g \circ f$	composition
$f(x+)$	right-hand limit
$f(x-)$	left-hand limit
$f', \boldsymbol{f}(\boldsymbol{x})'$	derivatives
$U(\boldsymbol{P}, f), U(\boldsymbol{P}, f, \alpha), L(\boldsymbol{P}, f), L(\boldsymbol{P}, f, \alpha)$	Riemann sums
$\mathcal{R}, \mathcal{R}(\alpha)$	classes of Riemann (Stieltjes) integrable functionas
$\mathcal{C}(X)$	space of continiuous functions
$\lVert\ \rVert$	norm
exp	exponential function
D_N	Dirichlet kernel

$\Gamma(x)$	gamma function
$\{\boldsymbol{e}_1, \cdots, \boldsymbol{e}_n\}$	standard basis
$L(X), L(X,Y)$	spaces of linear transformation
$[\boldsymbol{A}]$	matrix
$D_J f$	partial derivative
∇f	gradient
$\mathcal{C}', \mathcal{C}''$	classes of differentiable functions
$\det[\boldsymbol{A}]$	determinant
$\boldsymbol{J}_f(\boldsymbol{x})$	Jacobian
$\frac{\partial(y_1,\cdots,y_n)}{\partial(x_1,\cdots,x_n)}$	Jacobian
$\mathbb{I}^k$	k-cell
$\mathbb{Q}^k$	k-simplex
$d\boldsymbol{x}_{\boldsymbol{I}}$	basic k-form
$\wedge$	multiplication symbol
d	defferentiation operator
$\omega_{\boldsymbol{T}}$	transform of ω
∂	boundary operator
$\nabla \times \boldsymbol{F}$	curl
$\nabla \cdot \boldsymbol{F}$	divergence
$\mathcal{E}$	ring of elementary sets
m	Lebesgue measure
μ	measure
$\mathcal{M}_F, \mathcal{M}$	families of measurable sets
f^+, f^-	positive(negative) part of f
K_E	characteristic function
$\mathcal{L}, \mathcal{L}(\mu), \mathcal{L}^2, \mathcal{L}^2(\mu)$	classes of Lebesgue-integrable functions
WF	Wirtinger Flow
TWF	Truncated Wirtinger Flow
ITWF	Incrementally Truncated Wirtinger Flow
IMTWF	Incrementally Minibatched Truncated Wirtinger Flow
RWF	Reshaped Wirtinger Flow
IRWF	Incrementally Reshaped Wirtinger Flow
IMRWF	Incrementally Minibatched Reshaped Wirtinger Flow
$\langle \cdot, \cdot \rangle$	scalar product
$\lvert z \rvert$	absolute value

1 Introduction

1.1 Related work

- Citations can be done with `biblatex`. Here is how to cite a book [1], an article [2], a proceedings paper [3] and a technical report [4]. In this example, the actual references are defined in the database file `thesis.bib`, which is included in the header of this document.
- References should be done using `\cref`. This way, the type of the object we are referencing is automatically added. For instance, "`\cref{chap:Introduction}`" leads to "Chapter 1".
- Plots, even from data files, can be done with TikZ, cf. Figure 1.1
- Values with units should preferably printed with the `siunitx` package, e.g. 1 m is the result of "`\SI{1}{\metre}`". This works both in text and in math mode.

$|z|$

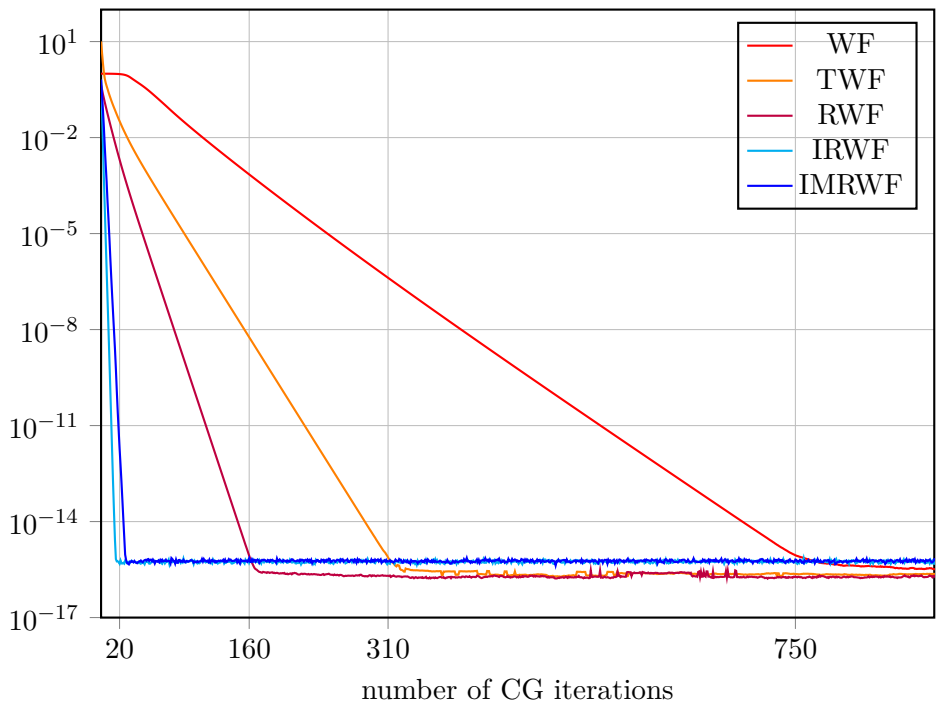

Figure 1.1: TikZ can create beautiful plots directly from data files. These plots use vector graphics and their fonts are fully consistent with the fonts of the document.

sdf fsarg The style defines multiple mathematical environments. All the environments allow to specify a name as optional parameter, as exemplified in Theorem 1.1.1.

1.1.1 Theorem (Theorem Name, optional)**.** *Here goes the actual theorem description.*

Proof. Here goes the proof of the theorem. This environment automatically puts a QED square at its end. Sometimes, the automatic placement is not optimal. In this case, `\qedhere` allows to place the symbol at a specific position, for instance in an equation:

$$a^2 + b^2 = c^2 \qquad \square$$

1.1.2 Example. Example of an example.

1.1.3 Definition. This is a definition.

1.1.4 Proposition. *This is a proposition.*

Equivalence proofs where each direction is shown separately can be formatted using the `\itemize` environment with custom labels. If the proof starts with this environment, put a `\leavevmode` before the environment to ensure that the first direction starts on a new line.

Proof.

"$\Rightarrow$": First direction.

"$\Leftarrow$": Second direction. $\square$

1.1.5 Lemma. *This is a lemma.*

1.1.6 Corollary. *This is a corollary.*

sdf

1 Problem.

2 Wirtinger Flow

The whole thing about *Wirtinger Flow* variants started with the seminal work of Candes and Soltanolkotabi[5]. The most important improvements chronologically were done by Candes and Chen[6], Kolte and Özgür[7], and Zhang et al.[8]. For a quite extensive survey on *Wirtinger Flow* variants please refer to Liu et al.[9]. Chandra et al.[10] gathered quite number of *Phase Retrieval* methods including a couple of *Wirtinger Flow* variants in the MATLAB® problem solving environment in a uniform manner.
We quickly go over the problem formulation, difficulties, algorithms, and at the of the chapter we give some numerical experiments we are going to refer to in the subsequent chapters.

2.1 Problem Formulation

Consider the ray $\boldsymbol{x} \in \mathbb{C}^{n\times 1}$ is emitted onto the object of interest and the diffracted rays are measured as $\boldsymbol{y} \in \mathbb{R}^{m\times 1}$ and is connected to the original ray by $\boldsymbol{y} = \varphi(\boldsymbol{A}\boldsymbol{x})$, where $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ and φ the usual element-wise absolute value(or the squared absolute value) from $\mathbb{C}^{m\times 1}$ to $\mathbb{R}^{m\times 1}$.
Candes and Soltanolkotabi[5] considered φ to be squared element-wise absolute value and the loss function to be quadratic. The summary for all the variants in terms of formulation is in table2.1

2.2 Difficulties

The loss function is non-convex. Set $n = 1$, $m = 2$, $\boldsymbol{x}_1 = \big(1+i\big)^{1\times 1}$, $\boldsymbol{x}_2 = \big(-1-i\big)^{1\times 1}$, $\boldsymbol{A} = \begin{pmatrix} 1 \\ i \end{pmatrix}^{2\times 1}$, $\boldsymbol{y} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}^{2\times 1}$, and $\lambda = 1/2$ to build a counterexample. Non-convexity is bad news for optimization as it can be seen vividly in [11] and [12]. To make the matter worse the loss function is not holomorphic(it can be easily seen that Cauchy-Riemann equations[13] do not hold) and therefore complex differentiability is out of the question[13].

Wirtinger Flow Variant	φ	loss functions
Wirtinger Flow	$\lvert\boldsymbol{z}\rvert^2$	quadratic
Truncated Wirtinger Flow	$\lvert\boldsymbol{z}\rvert^2$	quadratic
Incrementally Truncated Wirtinger Flow	$\lvert\boldsymbol{z}\rvert^2$	quadratic
Reshaped Wirtinger Flow	$\lvert\boldsymbol{z}\rvert$	quadratic
Incrementally Reshaped Flow	$\lvert\boldsymbol{z}\rvert$	quadratic

Table 2.1: φ and the loss function used in [5], [6], [7], [8]

Input: $\boldsymbol{y} = \{y_i\}_{i=1}^m$, $\{\boldsymbol{a}_i\}_{i=1}^m$;
Parameters: Lower and upper thresholds α_l, α_u for truncation in initialization, step size μ;
Initialization: Let $\boldsymbol{z}^{(0)} = \lambda_0 \tilde{\boldsymbol{z}}$, where $\lambda_0 = \frac{mn}{\sum_{i=1}^m \|\boldsymbol{a}_i\|_1} \cdot \left(\frac{1}{m}\sum_{i=1}^m y_i\right)$ and $\tilde{\boldsymbol{z}}$ is the leading eigenvector of

$$\boldsymbol{Y} := \frac{1}{m}\sum_{i=1}^m y_i \boldsymbol{a}_i \boldsymbol{a}_i^* \mathbf{1}_{\{\alpha_l \lambda_0 < y_i < \alpha_u \lambda_0\}}. \tag{2.1}$$

Update loop: for $t = 0 : T-1$ do

$$\boldsymbol{z}^{(t+1)} = \boldsymbol{z}^{(t)} - \frac{\mu}{m}\sum_{i=1}^m \left(\boldsymbol{a}_i^* \boldsymbol{z}^{(t)} - y_i \cdot \frac{\boldsymbol{a}_i^* \boldsymbol{z}^{(t)}}{|\boldsymbol{a}_i^* \boldsymbol{z}^{(t)}|}\right) \boldsymbol{a}_i. \tag{2.2}$$

Output $\boldsymbol{z}^{(T)}$.

Algorithm 1: Reshaped *Wirtinger Flow* suggested by [8]

Input: $\boldsymbol{y} = \{y_i\}_{i=1}^m$, $\{\boldsymbol{a}_i\}_{i=1}^m$;
Initialization: Same as in RWF (Algorithm 1);
Parameters: Lower and upper thresholds α_l, α_u for truncation in initialization, step size μ;
Update loop: for $t = 0 : T-1$ do
Choose i_t uniformly at random from $\{1, 2, \ldots, m\}$, and let

$$\boldsymbol{z}^{(t+1)} = \boldsymbol{z}^{(t)} - \mu \left(\boldsymbol{a}_{i_t}^* \boldsymbol{z}^{(t)} - y_{i_t} \cdot \frac{\boldsymbol{a}_{i_t}^* \boldsymbol{z}^{(t)}}{|\boldsymbol{a}_{i_t}^* \boldsymbol{z}^{(t)}|}\right) \boldsymbol{a}_{i_t}, \tag{2.3}$$

Output $\boldsymbol{z}^{(T)}$.

Algorithm 2: Incremental Reshaped *Wirtinger Flow* (IRWF) suggested by [8]

Input: $\boldsymbol{y} = \{y_i\}_{i=1}^m$, $\{\boldsymbol{a}_i\}_{i=1}^m$;
Initialization: Same as in RWF (Algorithm 1);
Update loop: for $t = 0 : T-1$ do
Choose Γ_t uniformly at random from the subsets of $\{1, 2, \ldots, m\}$ with cardinality k, and let

$$\boldsymbol{z}^{(t+1)} = \boldsymbol{z}^{(t)} - \mu \cdot \boldsymbol{A}_{\Gamma_t}^* \left(\boldsymbol{A}_{\Gamma_t} \boldsymbol{z}^{(t)} - \boldsymbol{y}_{\Gamma_t} \odot \mathrm{Ph}(\boldsymbol{A}_{\Gamma_t} \boldsymbol{z}^{(t)})\right), \tag{2.4}$$

where $\boldsymbol{A}_{\Gamma_t}$ is a matrix stacking $\boldsymbol{a}_i^*$ for $i \in \Gamma_t$ as its rows, $\boldsymbol{y}_{\Gamma_t}$ is a vector stacking y_i for $i \in \Gamma_t$ as its elements, $\odot$ denotes element-wise product, and $\mathrm{Ph}(\boldsymbol{z})$ denotes a phase vector of $\boldsymbol{z}$.
Output $\boldsymbol{z}^{(T)}$.
Algorithm 3: Minibatch Incremetnal Reshaped *Wirtinger Flow* (minibatch IRWF) suggested by [8]

3 Deep Unfolding

talam

4 UNROLLED WIRTINGER FLOWS

5 Results

Bibliography

[1] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. 3rd ed. Prentice Hall, 2007.

[2] L. Armijo. "Minimization of functions having Lipschitz continuous first partial derivatives." In: *Pacific Journal of Mathematics* 16.1 (1966), pp. 1–3.

[3] A. Chambolle. "Total Variation Minimization and a Class of Binary MRF Models." In: *Energy Minimization Methods in Computer Vision and Pattern Recognition*. Ed. by A. Rangarajan, B. Vemuri, and A. Yuille. Vol. 3757. LNCS. Springer, 2005, pp. 136–152.

[4] T. F. Chan, S. Esedoḡlu, and M. Nikolova. *Algorithms for Finding Global Minimizers of Image Segmentation and Denoising Models*. Tech. rep. 04-54. University of California, Los Angeles, Sept. 2004.

[5] E. J. Candes, X. Li, and M. Soltanolkotabi. *Phase Retrieval via Wirtinger Flow: Theory and Algorithms*. Apr. 2015.

[6] Y. Chen and E. J. Candes. *Solving Random Quadratic Systems of Equations Is Nearly as Easy as Solving Linear Systems*. 2016. arXiv: `1505.05114 [cs.IT]`.

[7] R. Kolte and A. Özgür. *Phase Retrieval via Incremental Truncated Wirtinger Flow*. 2016. arXiv: `1606.03196 [cs.IT]`.

[8] H. Zhang, Y. Zhou, Y. Liang, and Y. Chi. *Reshaped Wirtinger Flow and Incremental Algorithm for Solving Quadratic System of Equations*. 2016. arXiv: `1605.07719 [stat.ML]`.

[9] J.-W. Liu, Z.-J. Cao, J. Liu, X.-L. Luo, W.-M. Li, N. Ito, and L.-C. Guo. "Phase Retrieval via Wirtinger Flow Algorithm and Its Variants." In: *2019 International Conference on Machine Learning and Cybernetics (ICMLC)*. 2019, pp. 1–9.

[10] R. Chandra, Z. Zhong, J. Hontz, V. McCulloch, C. Studer, and T. Goldstein. *PhasePack: A Phase Retrieval Library*. 2017. arXiv: `1711.10175 [math.OC]`.

[11] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[12] J. Nocedal and S. J. Wright. *Numerical Optimization*. 2nd ed. Springer New York, 2006.

[13] L. V. Ahlfors. *Complex Analysis. An introduction to the theory of analytic functions of one complex variable*. 2nd ed. McGraw-Hill Book Company, 1966.