Signal Reconstruction From The Modulus of its Fourier Transform

Eliyahu Osherovich, Michael Zibulevsky, and Irad Yavneh

24/12/2008

Technion - Computer Science Department - Technical Report CS-2009-09 - 2009

Abstract

The problem of signal reconstruction from the magnitude of its Fourier transform arises in many applications where the wave phase is apparently lost or impractical to measure. One example of such an application that has attracted a substantial number of researchers in recent years is Coherent Diffraction Imaging (CDI). CDI is a "lens-less" technique for 2D and 3D nano-objects reconstruction with unprecedented resolution making possible visualization at the atomic level. Currently, the prevailing method of phase recovery is the Hybrid Input-Output (HIO) method, that was developed by Fienup in nineteen-seventies as a modification of the even older Gerchberg-Saxton method. In this work we analyze the problem of the phase retrieval from a non-convex optimization's standpoint. To our knowledge, we are the first to provide analysis of the second order derivatives (the Hessian) of the commonly used error measure. This analysis reveals some important details of already existing algorithms, and leads us to the development of a significantly faster reconstruction method.

Chapter 1

Introduction

1.1 Motivation

Rapid development of nano-technology has resulted in great interest in imaging techniques capable of providing satisfactory images of nano-structures. One of the most promising techniques for such a high resolution imaging is Coherent Diffraction Imaging (CDI). CDI has been successfully applied to visualizing a variety of nano-structures, such as nano-tubes [24], nano-crystals [22], defects inside nano-crystals [17], proteins and more [**?**]. In CDI, a highly coherent beam of X-Rays or electrons is incident on a specimen, generating a diffraction pattern. Under certain conditions the diffracted wavefront is approximately equal (within a scale factor) to the Fourier transform of the specimen. After being recorded by a CCD sensor, the diffraction pattern is used to reconstruct the specimen. However, due to the physical nature of the sensor, the phase of the diffracted wave is completely lost. Hence, we are limited to recording only the intensity (squared amplitude) of the diffracted wave. Effectively, this is equivalent to recording the magnitude of the specimen's Fourier transform. Due to the lack of phase information we face the problem of a signal reconstruction from partial information about its Fourier transform. More specifically, we are to reconstruct a signal from the magnitude of its Fourier transform. Reconstruction based on this incomplete data in the Fourier space, is performed by a software algorithm, that, in effect, replaces the objective lens of a typical microscope. The advantage in using no lenses is that the final image is aberration-free and so resolution is only diffraction and dose limited, i.e., dependent on the wavelength, aperture size and exposure time. However, the reconstruction algorithm is not trivial, and particularly computationally expensive, since the number of unknowns can easily surpass 10^9 for a three-dimensional signal.

1.2 Data Acquisition model

Of course, in the real world, the object $x(t)$ as well as its Fourier transform (denoted by $\hat{x}(\omega)$) are both continuous functions of t and ω respectively, where t and ω are multidimensional coordinate vectors. However, data acquisition and further processing are done on digital computers, hence, a complete analysis of the problem has to address issues associated with the data sampling and quantization. Note that we record the diffraction pattern intensity, not the object itself. Hence, our sampling happens in the Fourier domain and not in the object domain. Given that the object $x(t)$ has a limited spatial extent, say $x(t) = 0$ for $t \notin [0, 2T_c]$, the sampling in the Fourier domain must be done at the rate of $1/2T_c$ to prevent aliasing in the object domain. However, a more thorough examination of the problem yields an even higher sampling rate requirements. Recall that we record only the intensity of the diffraction pattern. The intensity can be represented as follows

$$I(\omega) = |\hat{x}(\omega)|^2 = \bar{\hat{x}}(\omega)\hat{x}(\omega), \tag{1.1}$$

where the overbar ($\bar{\cdot}$) operator denotes complex conjugate. Hence, the inverse Fourier transform of the measured intensity $I(\omega)$ provides the auto-correlation function of the object

$$\mathcal{F}^{-1}\left[I(\omega)\right] = x(-t) * x(t). \tag{1.2}$$

Since the intensity represents the Fourier transform of the auto-correlation, and the auto-correlation is twice as large as the object, the diffraction pattern intensity should be sampled at least twice as finely as the object to capture all possible information about the object. Therefore, as long as we deal with a discrete representation of both the object and the Fourier transform magnitude, related to each other through the Discrete Fourier Transform (DFT), we must double the size of the signal by padding it with zeros in order to satisfy the above sampling rate requirement. In what follows we will refer to a zero-padded signal as described above as a *sufficiently* padded signal.

Hereinafter in this paper we consider the phase retrieval problem for discrete signals only and the DFT transform is assumed to be unitary. Problems of the quantization process are beyond the scope of this work.

1.3 Reconstruction from incomplete Fourier data

Before we approach our main problem, let us consider various scenarios of incomplete Fourier data. Recall that the Fourier transform of a real signal x,

is, in general, complex. There are two common representations of a complex number z: one as a sum of its real and imaginary parts

$$z = \Re_z + i\Im_z,$$

and another one as a product of z's magnitude with the complex exponent of its argument

$$z = r_z e^{i\phi_z},$$

where $r_z = |z|$.

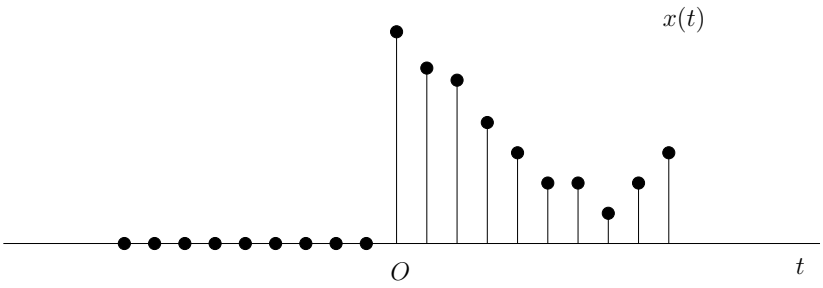
Let us consider first a situation when only the real part of a Fourier transform is known. As we show below, a perfect reconstruction of x is possible provided that the signal x was sufficiently padded with zeros, i.e., the padding size is equal to that of the original signal in every dimension. Recall that for any real signal x there exists a unique representation as a sum of two signals

$$x = x_e + x_o,$$

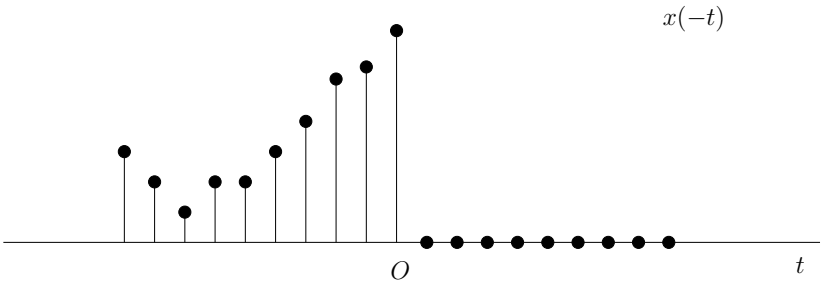
such that x_e is even and x_o is odd. It can be shown that $x_e(t) = \frac{1}{2}(x(t)+x(-t))$ and $x_o = \frac{1}{2}(x(t) - x(-t))$. Recall also that the Fourier transform of an even signal is real and that of an odd function is purely imaginary. Hence, we conclude that the real part of $\hat{x}$ is nothing but the Fourier transform of x_e, and the imaginary part is that of x_o. Hence, we are able to get the even part x_e by the inverse Fourier transform of the real part of $\hat{x}$. Reconstructing x from x_e is trivial, we should take the right part of x_e multiplied by 2 everywhere except at origin. Similarly, almost perfect reconstruction is possible from the imaginary part of the Fourier transform. In the latter case x can be reconstructed everywhere besides the origin. The idea is illustrated in Figure 1.1 on page 5.

Let us now consider the second representation, where a complex number is represented by its modulus and phase. We start with signal reconstruction from its Fourier representation phase. Several authors (see e.g. [6, 5, 13, 14, 16]) have shown that under certain conditions reconstruction is possible within a scale factor. Following is a theorem providing sufficient conditions for reconstruction (from [6])

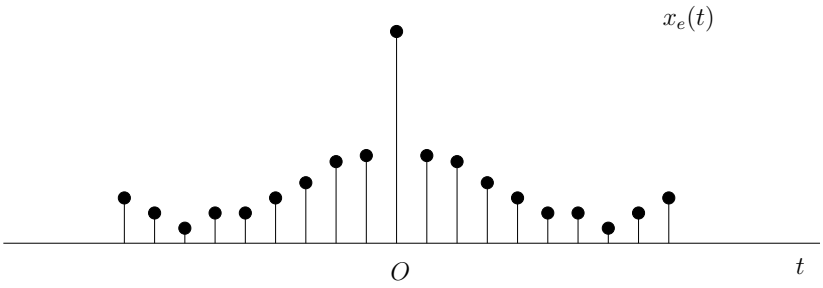
Theorem 1. *Let $x[n]$ be a sequence which is zero outside the interval $0 \leq n \leq N-1$ with $x[0] \neq 0$ and which has a z-transform with no zeros in reciprocal pairs or on the unit circle. Let $y[n]$ be any sequence which is zero outside the interval $0 \leq n \leq N-1$. If $\phi_y(\omega) = \phi_x(w)$ at $(N-1)$ distinct frequencies within the interval $0 < \omega < \pi$, then $y[n] = \beta x[n]$ for some positive constant β. If $\tan(\phi_y(\omega)) = \tan(\phi_x(\omega))$ at $(N-1)$ distinct frequencies in the interval $0 < \omega < \pi$, then $y[n] = \beta x[n]$ for some real constant β.*



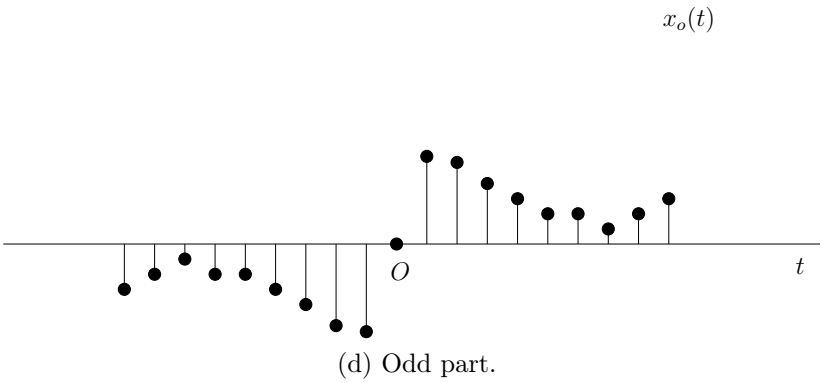
(a) Original signal.



(b) Reversed signal.

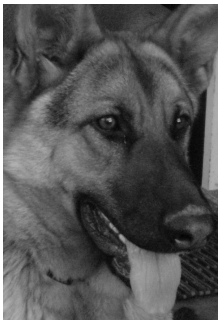


(c) Even part.



(d) Odd part.

Figure 1.1: Even and Odd parts of a signal.



(a) a dog image

(b) a cat image

(c) dog's magnitude with cat's phase

(d) cat's magnitude with dog's phase

Figure 1.2: Importance of phase

Whence, it follows that a reconstruction from the phase alone is possible (up to a scale factor) provided that the signal was padded with a sufficient number of zeros, similarly to reconstruction from the real or imaginary part of the Fourier transform.

In general, the phase of a signal's Fourier representation tends to preserve more features of the original signal. As we will see later reconstruction from the Fourier magnitude is not always possible. In addition, as an informal confirmation of the phase importance see Figure 1.2.

Finally we consider the case of known magnitude of the Fourier transform, which is the main subject of this paper. It turns out that reconstruction from the Fourier transform magnitude is significantly more difficult than reconstruction from other cases of incomplete Fourier data. Uniqueness of the reconstruction depends on reducibility of the z-transform of the sought signal. It has been shown (see e.g. [5]) that, if the z-transform of a signal $x[n]$ has at most one irreducible non-symmetric factor, then, the Fourier transform magnitude of the sufficiently zero-padded signal defines $x[n]$ uniquely up to trivial transformation. Trivial transformations, here, include a linear shift,

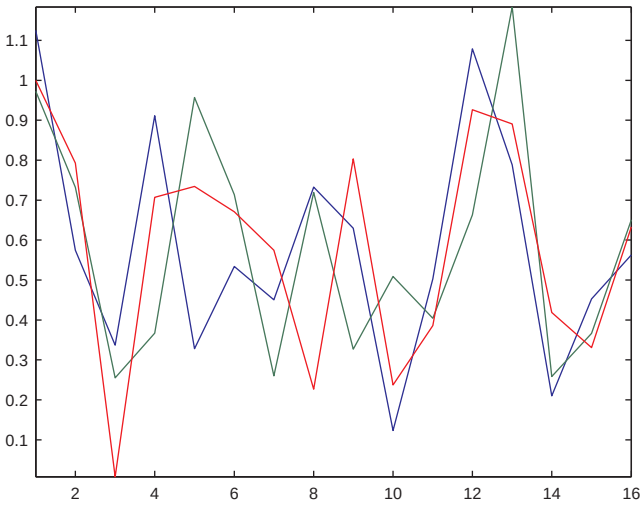


Figure 1.3: Several solutions in one dimension

coordinate-reversal, or a change in the sign of the signal, i.e.,

$$y[n] \sim x[n] \qquad \text{if} \quad y[n] = \pm x[k \pm n] \tag{1.3}$$

for some integer k. And the z-transform $X(z)$ of a signal $x[n]$ is said to be symmetric if, for some vector d of positive integers

$$X(z) = \pm z^{-d} X\left(z^{-1}\right),$$

where z^d is defined for two vectors $z = [z_1, z_2, \ldots, z_m]^T$ and $d = [d_1, d_2, \ldots d_m]$ as follows

$$z^d = z_1^{d_1} z_2^{d_2} \cdots z_m^{d_m}.$$

It should be pointed out that verification of reducibility of a z-transform is not practical. However, for a multidimensional signal it has been shown [7] that within the set of all polynomials in $m > 1$ variables, the subset of reducible polynomials is a set of measure zero as is the set of symmetric polynomials. Hence, it is very unlikely to have a non-unique (in the sense of the trivial transformations similarity as defined by Equation (1.3)) solution to the reconstruction problem in case of a multidimensional signal. Moreover irreducibility can be enforced by adding a certain reference signal [2]. In one-dimensional problems, the uniqueness is, in contrast, uncommon. Figure 1.3 illustrates three signals having the same Fourier magnitude, that are not related to each other via a trivial transformation.

The rest of the paper is organized as follows. In Chapter 2 we describe formally the problem in terms of constraints imposed in the Fourier and object

domains. Chapter 3 provides a review of modern reconstruction methods. Mathematical tools and subsequent analysis of the problem are done in Chapter 4. In Chapter 5 we present our results along with comparison with existing methods. Then we discuss directions of future research in Chapter 6. A short summary concluding the work appears in Chapter 7.

Please note that in order to simplify the typography and to improve readability we often depart from a strict mathematical notation, however, all cases of such an informal notation are explained and clarified when necessary.

Chapter 2

General Problem Description

2.1 Problem Definition

First, we start with a formal description of the reconstruction problem. Let us denote by x an unknown m-dimensional signal that we seek to reconstruct from the magnitude of its Fourier transform. For obvious reasons, knowing the magnitude of the Fourier transform is not sufficient for unique reconstruction of the signal.[1]Hence, one has to provide additional information to either guarantee unique reconstruction or to speed up the process. This additional information is usually given in the object domain. Thus, in a typical reconstruction problem one has two types of constraints: one in the Fourier space and another one in the object space. Among the Fourier space constraints may be the magnitude of the Fourier transform, phases known at some frequencies, and probably others. In the object space the constraints may include support information, bounds on x values, e.g., non-negativity, etc.

2.1.1 Fourier space constraints

In this paper we adopt the following notation for the Fourier transform. Two signals x and $\hat{x}$ will denote a Fourier transform pair, namely, their relationship with each other is given by

$$\hat{x} \triangleq \mathcal{F}[x],$$
$$x \triangleq \mathcal{F}^{-1}[\hat{x}],$$

[1]Note that even sufficiently oversampled Fourier transform, as required by Theorem 1, does not guarantee uniqueness.

where $\mathcal{F}[\cdot]$ represents the Unitary (Discrete) m-dimensional Fourier transform. In this notation, our Fourier domain constraint is given by

$$|\hat{x}| = r,$$

where r is the known (measured) magnitude of the Fourier transform of the sought signal x. Virtually all authors have adopted the following functional as a data discrepancy measure in the Fourier space

$$E = \| |\hat{x}| - r \|^2 , \tag{2.1}$$

where $\| \cdot \|$ denotes the standard L_2 vector norm. This approach is not optimal for several reasons. First, the data measured by the sensor is proportional to the scattered wave intensity which is proportional to r^2 and not to r. Consequently, a better choice would be to force $|\hat{x}|^2$ to be equal to r^2 by minimizing an appropriate norm. Second, by using a least squares formulation one implicitly assumes a normal noise distribution in the measurements. However, in our case, the physical process of data acquisition corresponds to counting the number of photons or electrons hitting the sensor. Measurements in such a process are known to have Poisson distribution of noise. Moreover, using $|x|^2$ instead of $|x|$ is more computationally efficient as we will show later. Impact of different choices of the error functional E is demonstrated in Section 5.2.2.

2.1.2 Object space constraints

The very basic information about the object is probably information about its space extent, which is called support. Additional constraints may impose bounds on values that x may receive. E.g. in the CDI x must be real and non-negative. In electron microscopy, on the other hand, x is a complex signal whose magnitude is known. The latter case is known as signal reconstruction from two intensity measurements. Historically, this was the problem that led to now classical Gerchberg-Saxton algorithm [4] which is, in fact, the progenitor of the currently most popular algorithms.

Chapter 3

Previous Work

3.1 Gerchberg-Saxton (Error Reduction) Algorithm

We start with the Gerchberg-Saxton algorithm that was the first particularly successful algorithm for the phase retrieval problem from two intensity measurements. Even more important, this algorithm serves as a basis for the later algorithms. The algorithm consists of the following four simple steps.

1. Fourier transform the current estimate of the signal.
2. Replace the magnitude of the resulting computed Fourier transform with the measured Fourier magnitude to form an estimate of the Fourier transform.
3. Inverse Fourier transform the estimate of the Fourier transform.
4. Replace the magnitude of the resulting computed signal with the measured signal modulus to form a new estimate of the signal.

As depicted in Figure 3.1. The Gerchberg-Saxton algorithm is easily generalized to a large class of problems. The generalized Gerchberg-Saxton algorithm can be used for any problem in which partial constraints (in the form of measured data or information known *apriori*) are known in each of the two domains, the object (or signal) domain and the Fourier domain. One simply transforms back and forth between the two domains, satisfying the constraints in one before returning to the other. This generalization of the Gerchberg-Saxton algorithm is known as the Error Reduction algorithm. It can be shown that the error decreases at each iteration of the Error Reduction algorithm, however, its convergence rate may be painfully slow. Moreover,

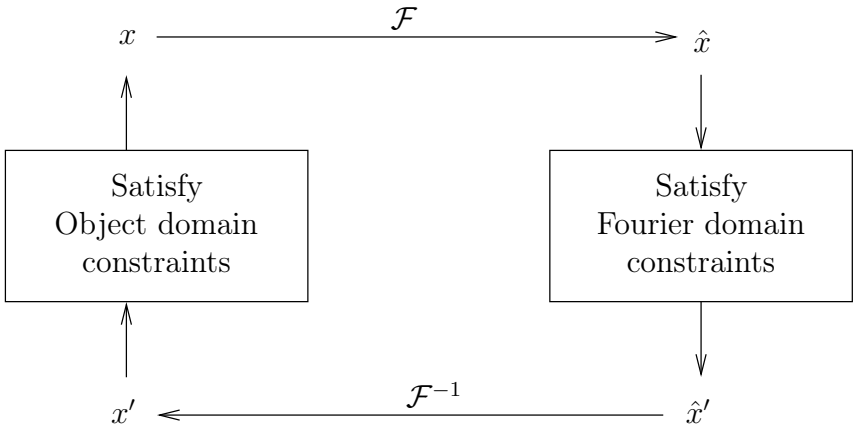


Figure 3.1: Block diagram of the generalized Gerchberg-Saxton algorithm.

convergence to a true solution is guaranteed only if the constraints are of a convex type [23].

3.2 Fienup Algorithms

Fienup in [3] suggested a family of iterative algorithms that are based on a different interpretation of the Error Reduction algorithm. These algorithms differ from the Error Reduction algorithm only in the object domain operations. The first three operations – Fourier transforming $x_k \xrightarrow{\mathcal{F}} \hat{x}_k$, satisfying the Fourier domain constraints $\hat{x}_k \rightarrow \hat{x}'_k$, and inverse Fourier transforming the result $\hat{x}'_k \rightarrow x'_k$ are the same for both algorithms. However, the further treatment is different. Fienup's insight was to group together these three steps into a non-linear system having an input x and an output x' as depicted in Figure 3.2 on page 13. The useful property of this system is that the output x' is always a signal having a Fourier transform that satisfies the Fourier domain constraints. Therefore, if the output also satisfies the object domain constraints, it is a solution of the problem.

Unlike the Error Reduction algorithm the input x must no longer be thought of as the current best estimate of the signal; instead, it can be thought of as a driving function for the next output, x'. The input x does not necessarily satisfy the object domain constraints.

Based on this novel interpretation Fienup suggested three algorithms for the phase retrieval problem from a single intensity and *apriori* knowledge of the signal x being non-negative everywhere.

input-output This algorithm is based on a claim that a small change of the

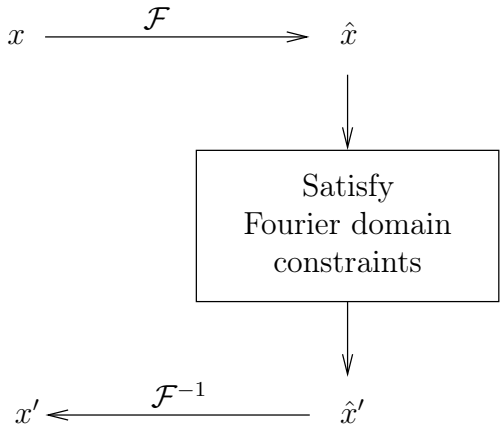


Figure 3.2: Block diagram for system for the input-output concept.

input results in a change of the output that is a constant α times the change in the input. Hence, if a change Δx is desired in the output, a logical choice of the change of the input to achieve that change in the output would be $\beta\Delta x$, where β is a constant, ideally equal to α^{-1}. For the problem of phase retrieval from a single intensity measurement the desired change of the output is

$$\Delta x_k(t) = \begin{cases} 0, & t \notin \nu, \\ -x'_k(t), & t \in \nu, \end{cases}$$

where ν is the set of points at which $x'(t)$ violates the object domain constraints. That is, where the constraints are satisfied, one does not require a change of the output. On the other hand; where the constraints are violated, the desired change of the output, in order to satisfy the non-negativity constraint is one that drives it towards a value of zero, and therefore, the desired change is the negated output at those points. Hence, the logical choice for the next input is

$$\begin{aligned} x_{k+1}(t) &= x_k(t) + \beta\Delta x_k(t) \\ &= \begin{cases} x_k(t), & t \notin \nu, \\ x_k(t) - \beta x'_k(t), & t \in \nu. \end{cases} \end{aligned} \tag{3.1}$$

output-output This algorithm is based on the following observation of the non-linear system depicted in Figure 3.2. If the output x' is used as an input, the resulting output will be x' itself, since it already satisfies the Fourier domain constraints. Therefore, irrespective of what input

actually resulted in the output x', the output x' can be considered to have resulted from itself as an input. From this point of view another logical choice for the next input is

$$\begin{aligned} x_{k+1}(t) &= x'_k(t) + \beta \Delta x_k(t) \\ &= \begin{cases} x'_k(t), & t \notin \nu, \\ x'_k(t) - \beta x'_k(t), & t \in \nu. \end{cases} \end{aligned} \tag{3.2}$$

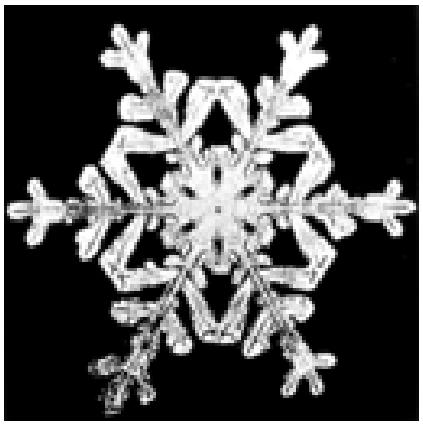
Note that if $\beta = 1$ the output-output algorithm becomes the Error Reduction algorithm. And since best results are usually obtained with $\beta \neq 1$ the Error Reduction algorithm can be viewed as a sub-optimal version of the input-output algorithm.

hybrid-input-output Finally we consider the third algorithm suggested by Fienup. This time the next input is formed by a combination of the upper line of Equation (3.2) with the lower line of Equation (3.1):

$$\begin{aligned} x_{k+1}(t) &= x'_k(t) + \beta \Delta x_k(t) \\ &= \begin{cases} x'_k(t), & t \notin \nu, \\ x_k(t) - \beta x'_k(t), & t \in \nu. \end{cases} \end{aligned} \tag{3.3}$$

The last algorithm known as the Hybrid Input Output (HIO) algorithm is currently the most widely used algorithm in the industry due to its simplicity and usually best convergence rate amongst the above three algorithms. However, mixing algorithms often yields results better than any standalone algorithm. Below we demonstrate the convergence of these algorithms on three test signals shown in Figure 3.3 on page 15 These three signals, one being a typical image in the X-Ray crystallography (Flake), another, being a typical "natural" image (Lena), and another one representing an image typical to computerized tomography (Phantom), will be used throughout this paper for algorithm performance evaluation and comparison. Figure 3.4 on page 16 demonstrates convergence rates of Fienup's Output-Output (OO) and Hybrid Input-Output (HIO) algorithms. The Input-Output algorithms failed to converge.

Since the HIO algorithm is consistently better than the others we shall compare our results with it.

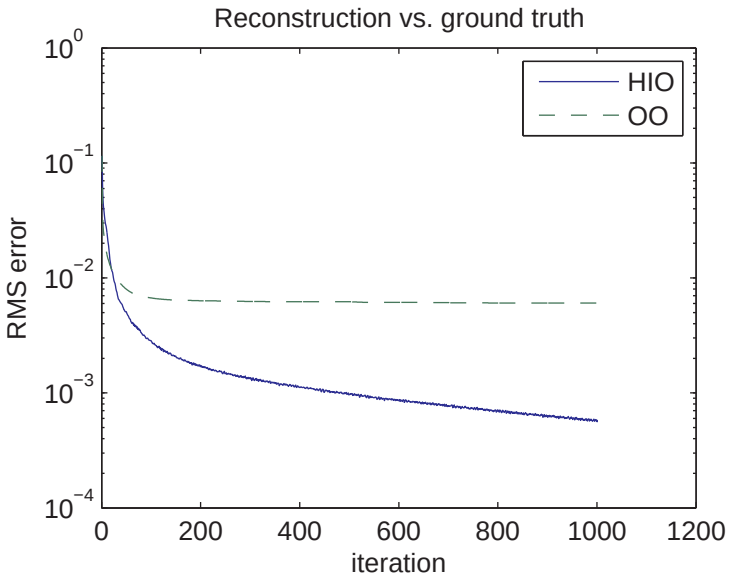


(a) Typical X-Ray Crystallography image.

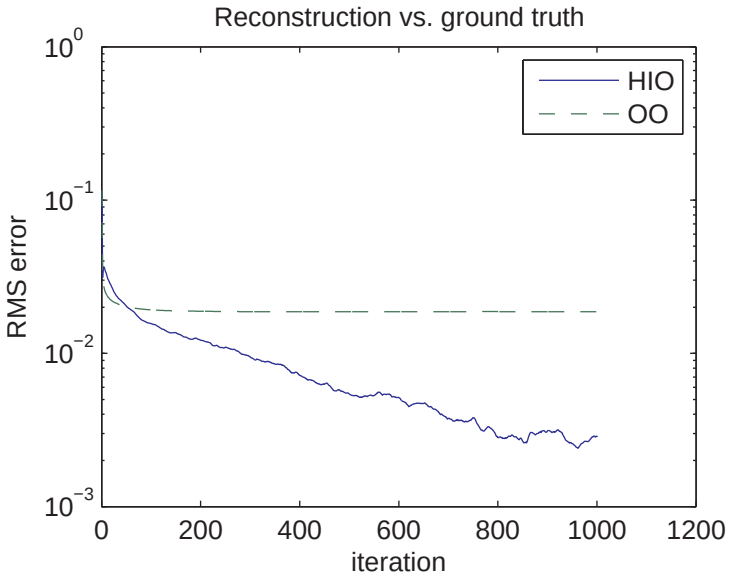
(b) Typical natural image.

(c) Typical tomography image.

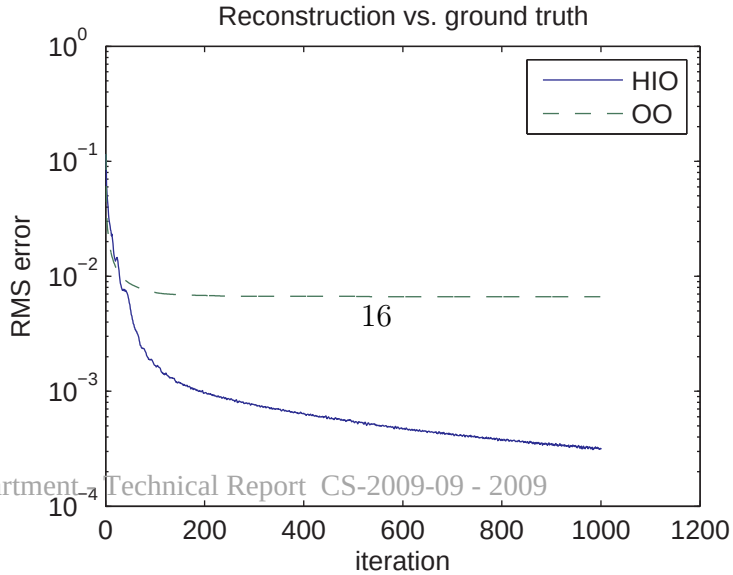
Figure 3.3: Signal examples.



(a) Snowflake reconstruction



(b) Lena reconstruction



Chapter 4

Optimization Framework

4.1 Mathematical Formulation

In this work our main approach is based on unconstrained optimization techniques. Hence, we shall start with a proper objective function definition. Let us consider first the most frequently used objective function in the Fourier domain,

$$E = \| |\hat{x}| - r \|^2, \tag{4.1}$$

where $\hat{x}$ denotes the Fourier transform of a signal x, r denotes the measured magnitude of the Fourier transform, and $\| \cdot \|$ denotes the standard L_2 vector norm. Note, that $\hat{x}$ and r are not necessarily one-dimensional vectors, hence, strictly speaking, the L_2 is not properly defined in all cases. A proper notation would be

$$E = \| \operatorname{vec} (|\hat{x}| - r) \|^2,$$

where the operator $\operatorname{vec}(\cdot)$ is a simple re-arrangement of a multidimensional signal x into a column vector in some predefined order. For example, let x be a two-dimensional $m \times n$ signal (matrix) and x_i its i-th column; then $\operatorname{vec}(x)$ is an $mn \times 1$ vector

$$\operatorname{vec}(x) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}.$$

Thus, in our convention the vec operator transforms a matrix into a column vector by stacking the matrix columns one beneath the other. Of course, the vec operator is defined for signals of arbitrary (finite) dimensionality. For the sake of brevity, hereinafter in this paper we shall use x and $\operatorname{vec}(x)$

interchangeably and an appropriate form should be clear from the context. Let us now review the objective function defined by Equation (4.1)

$$\begin{aligned} E &= \| |\hat{x}| - r \|^2 \\ &= \| |\mathcal{F}[x]| - r \|^2 \\ &= \| |Fx| - r \|^2 . \end{aligned} \tag{4.2}$$

Here, $\mathcal{F}[x]$ denotes the Discrete Fourier Transform (DFT) operator and F is the corresponding matrix in the sense that

$$\mathrm{vec}(\mathcal{F}[x]) = F\,\mathrm{vec}(x).$$

We introduce the DFT matrix F just for convenience, however, practical implementation shall not use F explicitly. Note also that Fx means, actually, $F\,\mathrm{vec}(x)$, however, the shorter form is used, as mentioned earlier. Consider now the final form of the objective function we obtained in Equation (4.2)

$$E = \| |Fx| - r \|^2 ,$$

which can be viewed as a non-linear real function of a complex argument

$$E = \varphi(Fx), \tag{4.3}$$

so that $\varphi : \mathbb{C}^N \mapsto \mathbb{R}$, where N is the number of elements in x. For large N, storing the Hessian matrix of φ (or an approximation of it) is not feasible due to its size of $N \times N$ elements. Therefore, Newton type optimization methods are not applicable in most realistic cases. In this work we use the SESOP method [15] that was specifically developed for cases where the linear operator F is computationally expensive. Moreover, it was reported to provide better results than Conjugate Gradients methods. Details of efficient implementation will be discussed in Section 4.3, meanwhile we shall develop the necessary mathematical basis. For an optimization method to be efficient one must provide information about the objective function gradient and the Hessian. One option is to consider E as a real function of a real argument x and to perform all required computations. However, this approach is less suitable for an efficient implementation using the SESOP algorithm and, more importantly, this approach is less extensible. That is changing, the functional will require the whole computation to be done again from scratch. Therefore, we use another approach which, is more elegant and easily extensible. The method is developed in the following subsections.

4.1.1 Gradient and Hessian Definition via Differentials

Following the approach of [11] let f be a differentiable scalar function of an $n \times 1$ vector of real variables x. Consider now the first differential of f.

$$\mathrm{df} = \nabla \mathrm{f}^{\mathrm{T}} \mathrm{d}x,$$

where ∇f denotes the gradient of f. Alternatively, the above formula can be re-written as follows

$$\mathrm{d}f = \langle \nabla f, \mathrm{d}x \rangle \,, \tag{4.4}$$

where $\langle \cdot, \cdot \rangle$ denotes the standard inner product over $\mathbb{R}^n$. Equation (4.4) leads to a new definition of the gradient. Namely, the gradient ∇f of function f is defined as to satisfy $\mathrm{d}f = \langle \nabla f, \mathrm{d}x \rangle$. Similarly, we define the Hessian $\nabla^2 f$

$$\mathrm{d}\nabla f = \left\langle \nabla^2 f, \mathrm{d}x \right\rangle . \tag{4.5}$$

Note that, unlike in Equation (4.4), this time the inner product is between a matrix and a vector, in this case $\langle H, g \rangle \equiv Hg$.

Now, let us denote $Fx = z$, hence Equation (4.3) becomes

$$\begin{aligned} E(x) &= \varphi(Fx) \\ &= \varphi(z). \end{aligned}$$

In order to find the gradient and the Hessian of E we use the method of differentials. Let us consider the first differential of E

$$\begin{aligned} \mathrm{dE(x)} &= \mathrm{d}\varphi(\mathrm{z}) \\ &= \langle \nabla \varphi, \mathrm{d}z \rangle \\ &= \langle \nabla \varphi, \mathrm{d}(Fx) \rangle \\ &= \langle \nabla \varphi, F \mathrm{d}x \rangle \\ &= \langle F^* \nabla \varphi, \mathrm{d}x \rangle \,. \end{aligned}$$

Hence, using our definition we obtain

$$\nabla E = F^* \nabla \varphi, \tag{4.6}$$

where F^* denotes the Hermitian (conjugate) transpose of F. In a similar

manner we can find the Hessian of E

$$\begin{aligned}
\mathrm{d}\nabla \mathrm{E} &= \mathrm{d}(F^* \nabla \varphi) \\
&= F^* \mathrm{d}\nabla \varphi \\
&= F^* \left\langle \nabla^2 \varphi, \mathrm{d}z \right\rangle \\
&= \left\langle F^* \nabla^2 \varphi, \mathrm{d}z \right\rangle \\
&= \left\langle F^* \nabla^2 \varphi, \mathrm{d}(Fx) \right\rangle \\
&= \left\langle F^* \nabla^2 \varphi, F\mathrm{d}x \right\rangle \\
&= \left\langle F^* \nabla^2 \varphi F, \mathrm{d}x \right\rangle .
\end{aligned}$$

Hence, according to the definition of the Hessian

$$\nabla^2 E = F^* \nabla^2 \varphi F. \tag{4.7}$$

The only problem now is that the gradient $\nabla\varphi$ and the Hessian $\nabla^2\varphi$ are not formally defined, since, in our case, φ is a real scalar function of a complex vector. Therefore we shall develop a suitable formulation for such a gradient and Hessian in the next sections.

4.1.2 Complex gradient

Let us consider now the case of a real scalar function $\varphi(z)$ of a complex vector argument $z = [z_1, z_2, \ldots, z_n]^T$. One would like to use the standard definition for the gradient of φ

$$\nabla\varphi = \begin{bmatrix} \frac{\partial\varphi}{\partial z_1} \\ \frac{\partial\varphi}{\partial z_2} \\ \vdots \\ \frac{\partial\varphi}{\partial z_n} \end{bmatrix}.$$

However, this approach is not feasible in general case, since the derivative $\frac{\partial\varphi}{\partial z_i}$ is not defined, because φ is not a holomorphic function as we show in the next Lemma.

Lemma 2. *Let f be a real function of a complex argument z, then f cannot be holomorphic unless it is constant.*

Proof. Let us denote the complex argument $z = x + iy$ and $f(z) = u(z) + iv(z) = u(x, y) + iv(x, y)$. Let us assume now that f is holomorphic. Therefore, it must satisfy the Cauchy-Riemann equations

$$\begin{cases} \frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \\ \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}. \end{cases}$$

However, f is real, hence $v(x,y)=0$, which, in turn means that $\frac{\partial u}{\partial x}=\frac{\partial u}{\partial y}=0$. That is, $u(x,y)$ must is a constant and so is f. $\square$

Since constant functions are not particularly interesting we must develop a new complex gradient operator for real functions of complex argument. Brandwood in [1] suggested to treat $\varphi(z)$ as a function of two independent variables z and $\bar{z}$, where $\bar{z}$ denotes the complex conjugate of z. Using our gradient definition via differentials we can write

$$\begin{aligned}
\mathrm{d}\varphi &= \frac{\partial\varphi}{\partial z}\mathrm{d}z + \frac{\partial\varphi}{\partial\bar{z}}\mathrm{d}\bar{z} \\
&= 2\Re\left(\frac{\partial\varphi}{\partial z}\mathrm{d}z\right) \\
&= 2\Re\left\langle\overline{\left(\frac{\partial\varphi}{\partial z}\right)},\mathrm{d}z\right\rangle \\
&= 2\Re\left\langle\frac{\partial\varphi}{\partial\bar{z}},\mathrm{d}z\right\rangle \\
&= \Re\left\langle 2\frac{\partial\varphi}{\partial\bar{z}},\mathrm{d}z\right\rangle,
\end{aligned}$$

where, $\Re(z)$ represents the real part of a complex number z. According to our previous definition of the gradient we establish that

$$\nabla\varphi(z) = 2\frac{\partial\varphi}{\partial\bar{z}}. \tag{4.8}$$

Before we apply this result to our objective function let us show that the above definition provides desired qualities of a gradient. It was shown in [1] that the above definition of the complex gradient is consistent with two major qualities of the usual gradient. These theorems are provided below.

Theorem 3. *Let $f:\mathbb{C}\mapsto\mathbb{R}$ be a real valued function of a complex variable z. Let $f(z)=g(z,\bar{z})$, where $g:\mathbb{C}\times\mathbb{C}\mapsto\mathbb{R}$ is a function of two complex variables, such that $g(z,a)$ and $g(b,z)$ are each analytic functions of z. Then a necessary and sufficient condition for f to have a stationary point is that $\partial f/\partial z=0$, where the partial derivative with respect to z treats $\bar{z}$ as a constant in g. Similarly, $\partial f/\partial\bar{z}=0$ is also a necessary and sufficient condition.*

Theorem 4. *Let f and g be two functions as defined in Theorem 3, the gradient $\nabla f\equiv 2\frac{\partial g}{\partial\bar{z}}$ defines the direction of the maximum rate of change of f with z.*

Proof. Consider Equation (4.8)

$$\mathrm{d}f = \Re\left\langle 2\frac{\partial f}{\partial \bar{z}}, \mathrm{d}z \right\rangle.$$

Obviously

$$|\mathrm{d}f| = \left|\Re\left\langle 2\frac{\partial f}{\partial \bar{z}}, \mathrm{d}z \right\rangle\right| \leq \left|\left\langle 2\frac{\partial f}{\partial \bar{z}}, \mathrm{d}z \right\rangle\right|. \tag{4.9}$$

However, according to the Cauchy-Schwarz inequality

$$\left|\left\langle 2\frac{\partial f}{\partial \bar{z}}, \mathrm{d}z \right\rangle\right| \leq \left\|2\frac{\partial f}{\partial \bar{z}}\right\| \cdot \|\mathrm{d}z\|. \tag{4.10}$$

It is easy to verify that the equality in Equations (4.9) and (4.10) only holds when $\partial f/\partial \bar{z} = \alpha \mathrm{d}z$ for some real positive scalar α. □

Note that Brandwood's definition of the complex gradient was $^{\partial f}/_{\partial \bar{z}}$. He argued that for $f(z) = \bar{z}$ the gradient should be 1 not 2. Although both choices provide the same direction, the rate of change is not correct for the Brandwood's definition.

Now, with a complete theory in hand, we can return to our problem and to derive the gradient of our objective function. Using Equation (4.6) we obtain

$$\begin{aligned}\nabla E &= F^* \nabla \varphi \\ &= F^* 2\frac{\partial \varphi}{\partial \bar{z}}.\end{aligned}$$

Now, recalling that $\varphi(z) = \||z| - r\|^2$ we obtain

$$\begin{aligned}\nabla \varphi &= 2\frac{\partial \varphi}{\partial \bar{z}} \\ &= 4\left(|z| - r\right)\frac{\partial}{\partial \bar{z}}\left(|z|\right) \\ &= 4\left(|z| - r\right)\frac{\partial}{\partial \bar{z}}\left(z\bar{z}\right)^{1/2} \\ &= 4\left(|z| - r\right)\frac{1}{2}z^{1/2}\bar{z}^{-1/2} \\ &= 2\left(z - r\frac{z^{1/2}}{\bar{z}^{-1/2}}\right) \\ &= 2\left(z - r\frac{z}{|z|}\right).\end{aligned} \tag{4.11}$$

To obtain the gradient ∇E we must, by Equation (4.6), (left) multiply the above result by F^*

$$\nabla E = 2F^* \left(z - r\frac{z}{|z|} \right). \tag{4.12}$$

Note that Equation (4.12) gives us the gradient ∇E as a function of z. To obtain an equivalent expression as a function of x we substitute $z = Fx$ to obtain

$$\begin{aligned} \nabla E &= 2F^* \left(z - r\frac{z}{|z|} \right) \\ &= 2F^* \left(Fx - r\frac{Fx}{|Fx|} \right) \\ &= 2 \left(x - F^* r\frac{Fx}{|Fx|} \right). \end{aligned} \tag{4.13}$$

A few words about notation are in order. Note that quantities like r, z, x, Fx all represent vectors, thus the product rz or division $z/|z|$ are not defined. What we mean is element-wise product or division of corresponding vectors, hence, the result is also a vector of the same size.

Note that the expression $F^* r\frac{Fx}{|Fx|}$ has a clear physical meaning: start with a signal x, Fourier transform it into $\hat{x} \equiv Fx$, then replace the modulus of $\hat{x}$ with r and inverse Fourier transform the result. This is exactly x' we have already seen in Section 3.2. Later, in 4.2we will see additional properties of x'.

4.1.3 Complex Hessian

Our interest in the Hessian is twofold. First, there is no algorithm at the moment that uses the information available from the second order derivatives. All current algorithms are gradient based. Second, an analysis of the Hessian matrix should be useful in understanding the problem behaviour. Therefore, we would like to have an elegant and easy to compute method for the Hessian matrix calculations. The method, of course, should be suitable for a real-valued function of a complex argument. There were several attempts to define the Hessian in an appropriate form, first by Van den Bos in [21], later by Hjorgunes and Gesbert [8, 9, 10], however, none of these attempts has succeeded in producing results similar to that of the complex gradient. Nevertheless, for our purposes we can expand the idea of treating z and $\bar{z}$ as independent variables to find an elegant formula for Hessian-vector multiplication, since this is what is actually required in our algorithm.

Let us consider the first differential of the gradient as we obtained in 4.11

$$\begin{aligned}\mathrm{d}\left(\nabla\varphi\right) &= \frac{\partial\nabla\varphi}{\partial z}\mathrm{d}z + \frac{\partial\nabla\varphi}{\partial\bar{z}}\mathrm{d}\bar{z} \\ &= \frac{\partial\left(2z - 2rz^{1/2}\bar{z}^{-1/2}\right)}{\partial z}\mathrm{d}z + \frac{\partial\left(2z - 2rz^{1/2}\bar{z}^{-1/2}\right)}{\partial\bar{z}}\mathrm{d}\bar{z} \\ &= \left(2 - \frac{r}{|z|}\right)\mathrm{d}z + \frac{rz^2}{|z|^3}\mathrm{d}\bar{z}.\end{aligned}$$

Hence, multiplication of the Hessian $\nabla^2\varphi$ by a vector v reads

$$\nabla^2\varphi v = \left(2 - \frac{r}{|z|}\right)v + \frac{rz^2}{|z|^3}\bar{v}. \tag{4.14}$$

Although we do not have the Hessian $\nabla^2\varphi$ in an explicit form we do have the Hessian of E which will be analyzed in Section 4.2.

4.2 Mathematical analysis

In order to analyze the spectral properties of the Hessian matrix $\nabla^2 E$ we shall start with the Fourier transform matrix F, which plays a vital role in the Hessian decomposition, as follows from Equation (4.15). Recall that we defined F as follows

$$F\,\mathrm{vec}(x) \equiv \mathrm{vec}(\mathcal{F}[x]).$$

Consider a one-dimensional signal $x = [x_1, x_2, \ldots, x_n]^T$. Obviously, x can be represented as a weighted sum of the standard basis vectors

$$x = \sum_{i=1}^{n} x_i e_i,$$

where e_i is the vector with a 1 in the i-th coordinate and zeros everywhere. Thus, we have

$$\begin{aligned}
\mathcal{F}[x] &= \mathcal{F}\left[\sum_{i=1}^{n} x_i e_i\right] \\
&= \sum_{i=1}^{n} \mathcal{F}\left[x_i e_i\right] \\
&= \sum_{i=1}^{n} x_i \mathcal{F}\left[e_i\right] \\
&= \left[\mathcal{F}\left[e_1\right], |\mathcal{F}\left[e_2\right], \ldots \mathcal{F}\left[e_n\right]\right] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \\
&\equiv F_n x.
\end{aligned}$$

Hence, for a one-dimensional signal of size n we define the matrix $F \equiv F_n$ such that its i-th column is given by $\mathcal{F}\left[e_i\right]$. Obviously, $Fx = \mathcal{F}[x]$. The matrix F_n exhibits several characteristics:

1. F_n is unitary due to our choice of unitary DFT.
2. F_n is symmetric, i.e., $F_n^T = F_n$.
3. $F_n^{-1} = F_n^*$.
4. F_n^2 is a permutation matrix that reflects a signal around the origin, i.e.,

$$F_n^2 \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 \\ x_n \\ x_{n-1} \\ \vdots \\ x_3 \\ x_2 \end{bmatrix}.$$

5. $F_n^4 = I_n$, where I_n is the identity matrix of size $n \times n$.
6. As follows from property 5, the eigenvalues λ of F_n satisfy a characteristic equation $\lambda^4 = 1$. Therefore, the eigenvalues of F_n are the fourth roots of unity: λ is $+1$, -1, $+i$, or $-i$. Since there are only four distinct eigenvalues of this $n \times n$ matrix, each must have some multiplicity. The

problem of multiplicity was solved by McClellan and Parks in [12]. The multiplicity depends on the value of n modulo 4 and is given by the following table.

size n	$\lambda = +1$	$\lambda = -1$	$\lambda = +i$	$\lambda = -i$
$4m$	$m+1$	m	m	$m-1$
$4m+1$	$m+1$	m	m	m
$4m+2$	$m+1$	$m+1$	m	m
$4m+3$	$m+1$	$m+1$	$m+1$	m

Table 4.1: Multiplicities of the eigenvalues λ of the unitary Fourier DFT matrix of size $n \times n$.

Let us consider now a two dimensional $m \times n$ signal x. Due to the separability of the Fourier transform it can be first carried out along the first dimension and then along the second dimension. Let us consider this procedure in term of F_n.

$$\begin{aligned}\operatorname{vec}\left(\mathcal{F}[x]\right) &= \operatorname{vec}\left(\left(F_n\left(F_m x\right)^T\right)^T\right) \\ &= \operatorname{vec}\left(F_m^T x F_n^T\right) \\ &= \operatorname{vec}\left(F_m x F_n\right).\end{aligned}$$

By using the following identity

$$\operatorname{vec}(AYB) = \left(B^T \oplus A\right)\operatorname{vec}(Y)$$

we obtain

$$\begin{aligned}\operatorname{vec}\left(\mathcal{F}[x]\right) &= \operatorname{vec}\left(F_m x F_n\right) \\ &= \left(F_n^T \oplus F_m\right)\operatorname{vec}(x) \\ &= \left(F_n \oplus F_m\right)\operatorname{vec}(x) \\ &= F\operatorname{vec}(x).\end{aligned}$$

Hence, by the definition of F, we obtain $F = F_n \oplus F_m$, where the operator $\oplus$ denotes the Kronecker product. This result can easily be extended to a multidimensional case. Let x be a k-dimensional signal of size $d_1 \times d_1 \times \cdots \times d_k$, then the Fourier transform matrix can be built as follows:

$$F = F_{d_k} \oplus F_{d_{k-1}} \oplus \cdots \oplus F_{d_2} \oplus F_{d_1}.$$

The eigenvalues of F can be computed using the following Lemma

Lemma 5. *Given A and B denoting square matrices of size $p \times p$ and $q \times q$ respectively. Let $\alpha_1, \alpha_2, \ldots, \alpha_p$ be the eigenvalues of A and $\beta_1, \beta_2, \ldots, \beta_q$ those of B. Then, the eigenvalues of $A \oplus B$ are given by*

$$\alpha_i \beta_j, \qquad i = 1, 2, \ldots, p, \quad j = 1, 2, \ldots q.$$

Let us now derive an expression for the Hessian $\nabla^2 E$. Recall that $\nabla^2 E = F^* \left(\nabla^2 \varphi\right) F$. We do not have $\nabla^2 \varphi$ in hand, however, we have an operator of $\left(\nabla^2 \varphi\right) v$ multiplication, given by Equation (4.14). Hence, we can construct an analogous operator for $\nabla^2 E$.

$$\begin{aligned}
\left(\nabla^2 E\right) t &= F^* \left(\nabla^2 \varphi\right) F t \\
&= F^* \left(\nabla^2 \varphi\right) (F t) \\
&= F^* \left(\left(2 - \frac{r}{|\hat{x}|}\right) (F t) + \frac{r \hat{x}^2}{|\hat{x}|} \left(\overline{F t}\right) \right) \\
&= F^* \left(\left(2 - \frac{r}{|\hat{x}|}\right) (F t) + \frac{r \hat{x}^2}{|\hat{x}|^3} \left(\bar{F} t\right) \right) \\
&= F^* \left(\operatorname{diag}\left(2 - \frac{r}{|\hat{x}|}\right) F + \operatorname{diag}\left(\frac{r \hat{x}^2}{|\hat{x}|^3}\right) \bar{F} \right) t \\
&= F^* \left(\operatorname{diag}\left(2 - \frac{r}{|\hat{x}|}\right) F + \operatorname{diag}\left(\frac{r \hat{x}^2}{|\hat{x}|^3}\right) F^* \right) t.
\end{aligned}$$

Hence, we conclude that

$$\nabla^2 E = F^* \left(\operatorname{diag}\left(2 - \frac{r}{|\hat{x}|}\right) F + \operatorname{diag}\left(\frac{r \hat{x}^2}{|\hat{x}|^3}\right) F^* \right). \tag{4.15}$$

It can be shown that the eigenvalues of the Hessian $\nabla^2 E$ are given by

$$\lambda = 2 - \frac{r}{|\hat{x}|} \pm \frac{r}{|\hat{x}|}.$$

At the moment we have no way to know when, in the above equation, the sign should be plus and when minus. However, we observe that the number of pluses and minuses is exactly the same as the multiplicity of +1 and -1, respectively, as the eigenvalues of F^2. Analysis of the Hessian $\nabla^2 E$ eigenvalues leads to two important observations

1. There exist a subspace where the problem becomes convex. For example if one can guarantee that $|\hat{x}_k|$ at every iteration is (element-wise) greater than r, then the Hessian $\nabla^2 E$ is guaranteed to have only positive eigenvalues.

2. Once a solution is found, i.e., $|\hat{x}| = r$, the eigenvalues of the Hessian $\nabla^2 E$ becomes equal to 1 ± 1, hence, there are two distinct eigenvalues: 2 and 0, whose multiplicity is defined by the multiplicity of the eigenvalues of F^2. Hence, the problem is severely ill-conditioned and requires some regularization.

Another important observation is related to the Newton direction used in optimization. We are looking for a solution of the following equation

$$\left(\nabla^2 E\right) d = -\nabla E.$$

Let us consider the product

$$\begin{aligned}
\left(\nabla^2 E\right) \nabla E &= F^* \left(\nabla^2 \varphi\right) F \nabla E \\
&= F^* \left(\nabla^2 \varphi\right) F F^* \nabla \varphi \\
&= F^* \left(\nabla^2 \varphi\right) \nabla \varphi \\
&= F^* \left(\nabla^2 \varphi\right) \nabla \varphi \\
&= F^* \left(\nabla^2 \varphi\right) \left(2 \left(z - r\frac{z}{|z|}\right)\right) \\
&= 2F^* \left(\left(2 - \frac{r}{|z|}\right)\left(z - r\frac{z}{|z|}\right) + \frac{rz^2}{|z|^3}\overline{\left(z - r\frac{z}{|z|}\right)}\right) \\
&= 2F^* \left(2z - 2\frac{rz}{|z|} - \frac{rz}{|z|} + \frac{r^2 z}{|z|^2} + \frac{rz^2}{|z|^3}\left(\bar{z} - \frac{r\bar{z}}{|z|}\right)\right) \\
&= 2F^* \left(2z - 3\frac{rz}{|z|} + \frac{r^2 z}{|z|^2} + \frac{rz}{|z|} - \frac{r^2 z}{|z|^2}\right) \\
&= 2F^* \left(2z - 2r\frac{z}{|z|}\right) \\
&= 4F^* \left(z - r\frac{z}{|z|}\right) \\
&= 2\nabla E. \qquad (4.16)
\end{aligned}$$

Hence, we get an interesting result: the gradient ∇E is an eigenvector of the Hessian $\nabla^2 E$ with a corresponding eigenvalue 2. That means, actually, that the gradient search is equivalent to the Newton method in this case.

Hence, the step forcing the Fourier constraints $x \to x'$ is equal to a step in the steepest descent direction, as we showed in Equation 4.13 and the same step is, in effect, a Newton step as follows from Equation 4.16.

4.3 Efficient Implementation

We would like to expound on the representation of the error functional in the preceding chapter. Note that we prefer to express the error functional in the Fourier domain in terms of Fx and not in terms of x. Such a representation is not accidental. This choice is particularly suitable for an efficient optimization method called Sequential Subspace Optimization (SESOP) [15]. Consider the objective function we seek to minimize.

$$E(x) = \varphi(Fx) + f(x),$$

where $\varphi(Fx)$ represents the functional corresponding to the Fourier domain constraints and $f(x)$ is the functional related to the object domain constraints. The SESOP algorithm sequentially minimizes the objective function over subspaces spanned by several directions that are chosen in a certain manner. Possible choices of the directions will be describe later. For the moment let D be a matrix comprising the M chosen directions as its columns. Thus, at every iteration we finds a minimizer of $E(x)$ over the subspace spanned by the columns of M. Hence, the algorithm, in general, can be summarised as follows

1. Update matrix D, i.e, choose M directions.
2. Find
$$y^* = \text{argmin}_y(E(x_k + Dy)).$$
3. Update current iterate
$$x_{k+1} = x_k + Dy^*.$$
4. Repeat until convergence.

Let us consider the problem of a subspace optimization

$$\begin{aligned}\psi(y) &= E(x + Dy) \\ &= \varphi(F(x + Dy)) + f(x + Dy) \\ &= \varphi(Fx + FDy) + f(x + Dy) \\ &= \varphi(Fx + Ry) + f(x + Dy) \\ &= \varphi(z + Ry) + f(x + Dy).\end{aligned}$$

Since the number of search directions M is small we can use a modified Newton method for finding the minimizer y^*. Recall that the Newton method's search direction is given by

$$d = -\left(\nabla^2\psi\right)^{-1}\nabla\psi.$$

Using our results from Section 4.1 we obtain

$$\begin{aligned}\nabla\psi &= \nabla_y\varphi + \nabla_y f \\ &= R^*\nabla\varphi + D^*\nabla f\end{aligned}$$

and

$$\begin{aligned}\nabla^2\psi &= \nabla_y^2\varphi + \nabla_y^2 f \\ &= R^*\nabla^2\varphi R + D^*\nabla^2 f D,\end{aligned}$$

where ∇_y and ∇_y^2 denote, accordingly, the gradient and Hessian with respect to y. Note that the Hessian matrices $\nabla^2\varphi$ and $\nabla^2 f$ are not computed explicitly. Instead an operator form is used to compute a matrix-vector product, i.e., $(\nabla^2\varphi R)$ and $(\nabla^2 f D)$. Note also that finding a search direction does not require an (expensive) application of the linear operator F, since the matrices R and D were pre-computed earlier. A subsequent line search does not require an application of F either. Consider a step of size α in direction d

$$\begin{aligned}\psi(y+\alpha d) &= E(x + D(y+\alpha d)) \\ &= E(x + Dy + \alpha Dd) \\ &= \varphi(F(x + Dy + \alpha Dd)) + f(x + Dy + \alpha Dd) \\ &= \varphi(z + Ry + \alpha Rd) + f(x + Dy + \alpha Dd).\end{aligned}$$

Hence, once again, the matrices R and D are known and constant during the subspace optimization. Therefore, we need only one application of the linear operator F per every new direction we add to the subspace and one application of the adjoint operator F^* needed to compute the gradient $\nabla_x\varphi$.

Let us consider now the choice of the search directions. The simplest choice is a single direction of the steepest descent. In this case the SESOP becomes, in effect, the gradient descent method. Next, in addition to the current gradient, we add the last step direction. This approach can be viewed as a generalization of the Conjugate Gradients (CG) method. The two methods coincide in case of a linear quadratic problem. However, they demonstrate different behaviour for non-linear problems. Narkiss and Zibulevsky [15] argue that SESOP demonstrates better convergence rates than CG. In practice, additional directions may be added into the subspace search. One can add, for example, more previous directions and gradients to speed up the convergence rate. Note that this scheme requires only two applications of the linear operator F and one of the adjoint operator F^* per outer iteration, regardless of the number of the search directions. Of course, adding more directions will increase the cost of the subspace minimization, however, one can re-calculate the Hessian $\nabla^2\psi$ sparingly, using a BFGS update for intermediate calculations. This modification is planned for future work.

Chapter 5

Results

5.1 Convergence rate

Let us demonstrate the convergence rate of our algorithm as compared to that of the HIO. The result are shown in Figure 5.1.

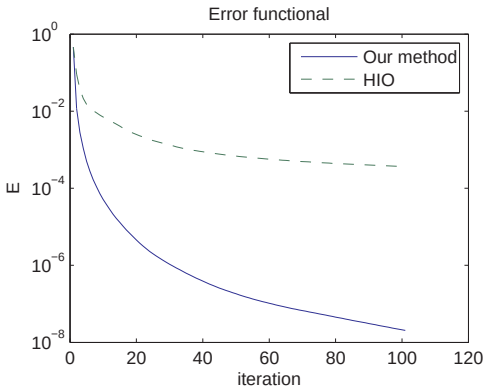
As can be seen in Figure 5.1, our algorithm demonstrates a significantly faster convergence rate. However, if we consider the reconstruction quality in the object domain the picture will be different. Let us look at Figure 5.2 that depicts the root means square error (RMS) in the object domain. .

As we can see in all cases except "Lena" the HIO algorithm provides better reconstruction in the object space even though it violates the Fourier domain constraints significantly. The reason for this phenomena is not yet understood. However, we can get some intuition if we look at the reconstruction results. Figure 5.3 demonstrates difference of the images obtained by our method with the ground truth image. As one can see the "snowflake" and "phantom" images have (multiple) shifted copies of themselves in the reconstructed images. This situation is possible as these two images do not have a tight support, especially the "phantom" image; and a shifted version of an image, has exactly the same Fourier magnitude. Finding a way to overcome this phenomena is one of our major goals of future research.

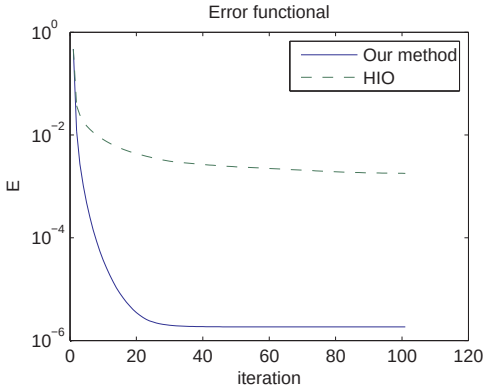
5.2 Robustness

5.2.1 Radius of convergence

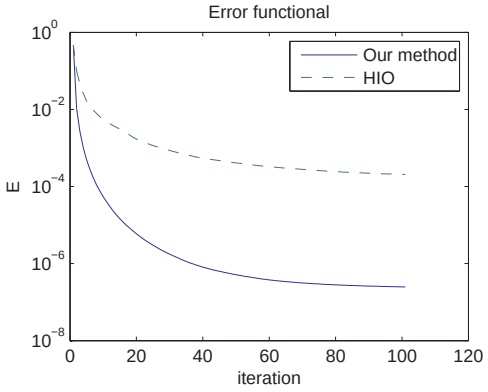
Non-convex optimization methods are usually not globally convergent. Hence, one must start sufficiently close to the solution in order to be able to converge to the global minimum. Our current experiments demonstrate that the HIO



(a) Snowflake reconstruction: Fourier domain

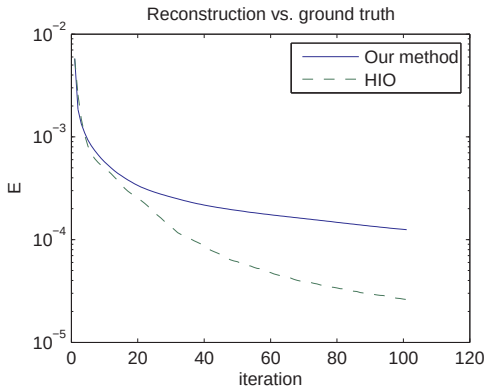


(b) Lena reconstruction: Fourier domain

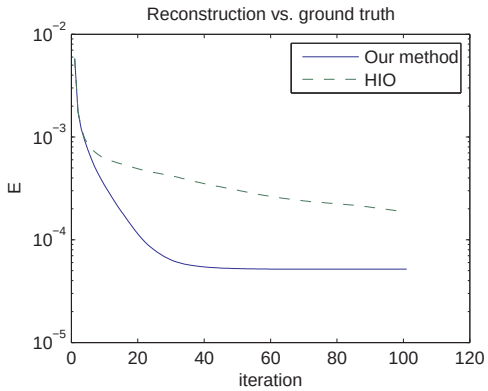


(c) Phantom reconstruction: Fourier domain

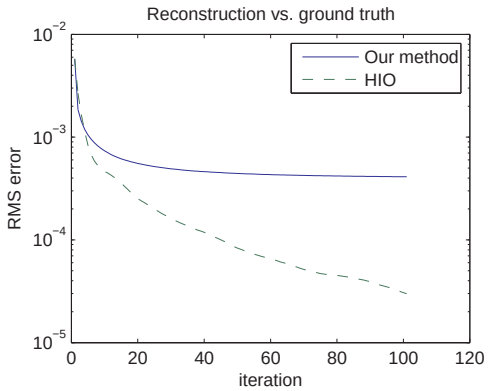
Figure 5.1: Convergence rate comparison: Fourier domain



(a) Snowflake reconstruction: object domain

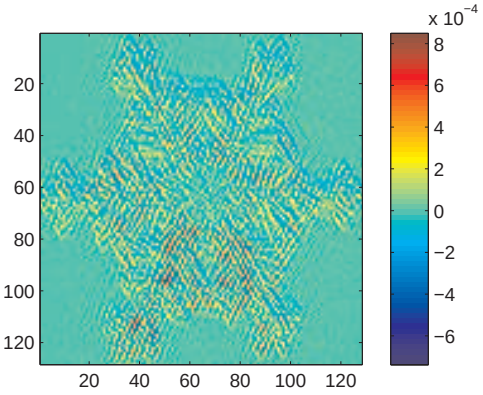


(b) Lena reconstruction: object domain

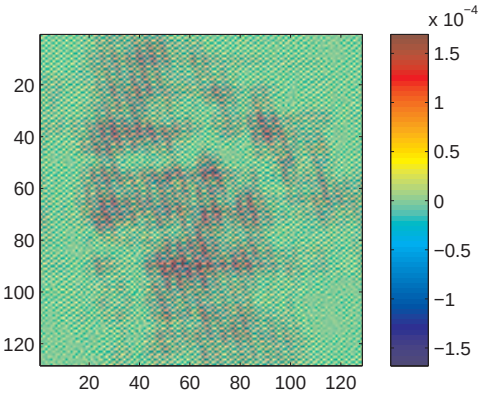


(c) Phantom reconstruction: object domain

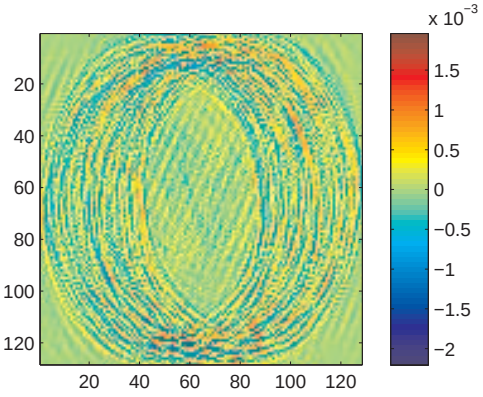
Figure 5.2: Convergence rate comparison: object domain



(a) Snowflake difference



(b) "Lena" difference



(c) Phantom difference

Figure 5.3: Reconstruction results: difference with the ground truth

algorithm is capable to reconstruct an acceptable image even when started at arbitrary point. Our algorithm, on the other hand, requires the starting point to be close to the solution. This probably happens because the HIO algorithm is not a usual optimization technique or, maybe, it implicitly minimizes some other functional, because the results of Section 5.1 demonstrate that a good reconstruction is achieved by the HIO algorithm despite a significant discrepancy in the Fourier domain. The reasons for such a behaviour and ways of improvement of our method should be addressed in future research.

5.2.2 Reconstruction from noisy data

Let us now analyze how an imperfect data affects reconstruction quality. It is important to recognize that in presence of noise the original signal is no longer a minimizer of the error functional. Hence, it is a common practice to introduce some *prior* on the reconstructed image in order to decrease the negative influence of data noise on the reconstruction quality. The choice of a good prior is of utmost importance and, ideally, should be chosen to fit the class of reconstructed images. The prior we used throughout this paper is the Total Variation (TV) [18],

$$TV(x) = \int |\nabla x|.$$

TV is generally accepted as a good prior for a broad class of images. Note that the HIO algorithm does not minimize any specific functional. Our method, in contrast, has a well defined error measure,

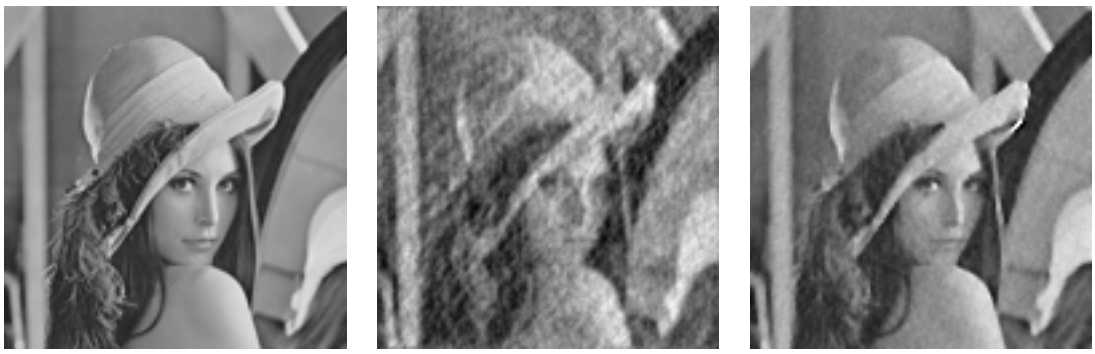
$$E_F = \| |\hat{x}| - r \|^2.$$

The above functional corresponds to the Maximum Likelihood estimate in case the data (r in this case) is perturbed with additive white Gaussian noise. Although this assumption is not correct in practice we present it as a pure mathematical model disconnected from the physical nature of the data. Figures 5.4, 5.2, and 5.3 demonstrate reconstruction results when the data r was perturbed with white Gaussian noise with signal to noise ratio (SNR) of -30dB. As is evident from the images the TV prior is very well suitable for piece-wise constant images like the phantom image.

Now we consider a model that should match the physical nature of the problem. Note that we record the scattered wave intensity thus, our data is, in effect, r^2 and not r. In addition, if we consider the particle nature of the waves, we should observer that the recorded image is based on the counting statistics of photons or electrons hitting the sensor. The noise of

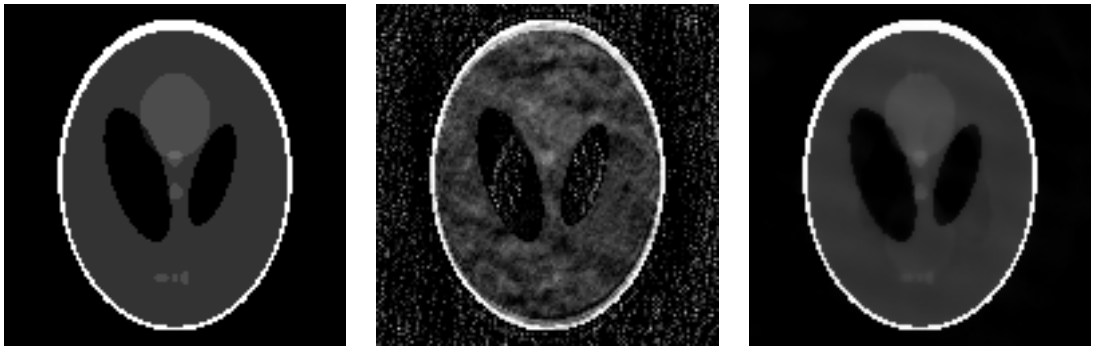
(a) original (b) HIO reconstruction: -11dB SNR (c) Our reconstruction: -17dB SNR

Figure 5.4: Snowflake reconstruction: noise in r with -30dB SNR



(a) original (b) HIO reconstruction: -17dB SNR (c) Our reconstruction: -24dB SNR

Figure 5.5: Lena reconstruction: noise in r with -30dB SNR



(a) original (b) HIO reconstruction: -6dB SNR (c) Our reconstruction: -24dB SNR

Figure 5.6: Phantom reconstruction: noise in r with -30dB SNR

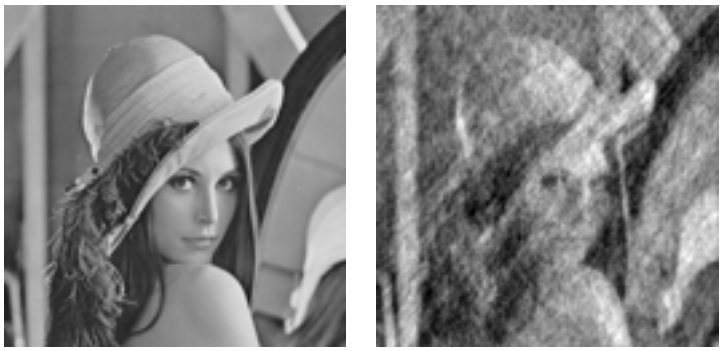
such a process is known to satisfy the Poisson distribution. Therefore a naive least squares functional, as shown below

$$E_{LS}(x) = \left\| \left|\hat{x}^2\right| - r^2 \right\|^2, \tag{5.1}$$

will be suboptimal. Instead, we have to use another functional that would satisfy the maximum likelihood condition. Such a functional, suitable for a Poisson noise in r^2 would be

$$E_{\mathcal{P}}(x) = |\hat{x}|^2 - r^2 \ln\left(|\hat{x}|^2\right). \tag{5.2}$$

It turns out that even a small amount of noise in r^2 degrade the reconstructed image severely. Figure 5.7 demonstrate the reconstruction we obtain wtih the HIO method when started with the true image as the initial guess Figure 5.8 depicts our reconstruction from the same starting point. First, using the least squares formulation as in Equation 5.1 (shown in Figure 5.8a). Then, using the functional that is designed for the Poisson noise as in Equation 5.2 (shown in Figure 5.8b). As is evident from the reconstruction results we obtain a better estimation using error functionals that were specifically developed for the noise model. The flexibility of choosing a functional suitable for the noise model is not possible within the HIO method.



(a) starting point (b) reconstruction (-16dB)

Figure 5.7: HIO reconstruction from a noisy data. Poisson noise in r^2 (-54dB SNR)

(a) Least Squares reconstruction: -21dB SNR (b) Possion noise fitted reconstruction: -25dB SNR

Figure 5.8: Our reconstruction from a noisy data. Poisson noise in r^2 (-54db SNR)

Chapter 6

Future Work

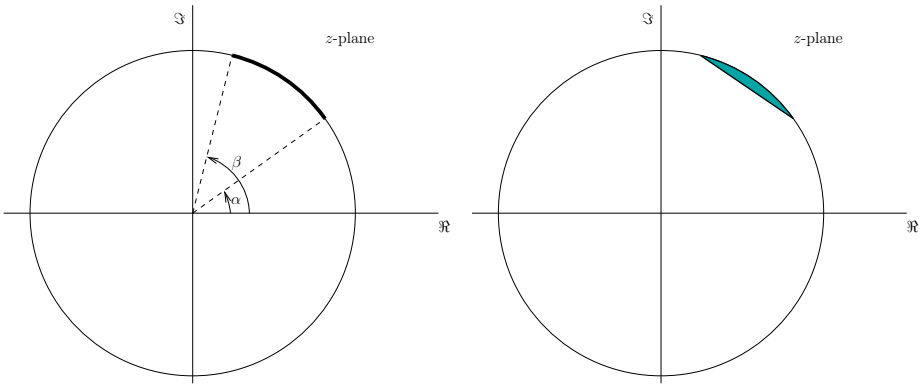
Our plans for future work are mainly concentrated on three major subjects: first, we have to improve the running time of the optimization algorithm; second, we shall find the best way to incorporate additional data into the existing flow; the last, but very important research area is how to make a multi-scale version of the algorithm. The purpose of the last improvements is twofold, first to further improve the running time of the algorithm; second, to improve its robustness allowing convergence for arbitrary starting point.

6.1 Improving optimization methods

Profiling of our algorithm running time data shows that the main time budget is spent in the inner loop of the Newton optimization. More specifically, generating the Hessian-vector multiplication operator and eventual multiplication itself are requiring a significant time. We shall investigate methods to evaluate the Hessian-vector multiplication only sparingly, intermediate values can be updated using a BFGS-like update strategy. Note that this improvement is very general and is not tailored to our problem. Success of this idea will result in a significantly faster optimization method.

Another possible improvement that has been partially implemented is to add a smart choice of search vectors. We tested an addition of an approximation to the Hessian direction as computed by the conjugate gradients and L-BFGS methods. Addition of these search directions did not prove to be worthwhile in the original problem, however, the same addition improved significantly the convexly relaxed version of the problem as described in Section 6.2.

Another idea that is applicable in general optimization is to test whether vector extrapolation methods [19, 20] can be applied to the Hessian update



(a) Non-convex constraint (b) Convex constraint

Figure 6.1: Convex relaxation

schemes like BFGS. In case of success, this may be of great interest to the optimization community.

6.2 Incorporating additional data (knowledge)

Another question we would like to address is how to incorporate additional knowledge (if available) into the problem. For example, provided that a low-resolution image of the sought signal is available from another source. What would be the best way to use it in our framework? Currently, such low-resolution images are mainly used for support information. However, it would be interesting to treat them as images blurred by a known convolution kernel and then to insert a deconvolution problem, which is known to be well posed, into our framework, to improve the condition number of the original problem.

In addition we consider a case where the phases are known approximately, i.e., there exists a bound on (a part of) the phase values. Say, the true phase belongs to some given interval $[\alpha, \beta]$ as shown in Figure 6.1a. In this case we can convert the problem into a convex one by the technique known as convex relaxation. Application of this technique to our problem is demonstrated in 6.1. Instead of forcing a complex number z to lie on the arc shown in Figure 6.1a we relax our requirement on the modulus letting z belong to the convex area shown in Figure 6.1b. This manipulation converts the problem to a convex one.

Unlike the original problem, the convex version is solved very efficiently, and more importantly, the convex version is globally convergent. Hence, a

(a) Snowflake - perfect reconstruction

(b) Lena - perfect reconstruction

(c) Phantom -48dB reconstruction SNR

Figure 6.2: Reconstruction with convex relaxation (phase uncertainty of 3 radians)

unique solution is obtained for *any* starting point. Of course, the obtained minimizer does not represent a correct solution any longer. Hence, we add the original (non-convex) constraint on the Fourier magnitude with a relatively small weight. Using this technique we were able to perfectly reconstruct the snowflake and "Lena" images withing a small number of iterations. Reconstruction of the phantom image stagnated for some reason, not yet determined. Hence, a perfect reconstruction was not achieved, however, the reconstruction quality was very good (-48dB SNR). Reconstructed images are shown in Figure 6.2. Note that the phase uncertainty was 3 radians. That makes us believe that after the stagnation problem observed in the phantom image, is resolved, a perfect reconstruction is possible with only one bit of information per phase value.

6.3 Multi-Scale approach

We are interested in a multi-scale approach for two reasons. First, this technique can be used for an accelleration of already existing algorithm, like HIO. Second, it can be used in our optimizatin problem as a method of finding a good approximation for the solution. This approximation may be used as a starting point for our method.

To deploy a multi-scale method one has to provide three basic components.

1. Data restriction method from the rough grid to the fine grid.

2. Data interpolation method from the fine grid to the rought grid.

3. A relaxation (solution) method that is suitable for the above restriction and interpolation operators.

At the moment it seems that a proper data restriction is the most challenging problem. Note that we cannot use only the data corresponding to low frequencies. This is not poissible since the high-frequency components of a signal do change the magnitude of the Fourier transform of the low-frequency components because of the aliasing. We investigated some schemes that keep the phase difference between two aliased frequencies constant. However, this work is only preliminary and requires further research.

Chapter 7

Summary

We presented an optimization method for the phase retrieval problem from one intensity measurement. Due to high non-linearity and non-convexity of the problem, the method convergence to the true solution only when started sufficiently close it. Compared to current methods, like the Hybrid Input-Outpum algorithm, our method has a smaller convergence radius. However, our approach allows a great deal of flexibility that is not poissible in the HIO method. For example, we can easily add priors on the image or, change the objective functional to fit the noise distribution in the data. These features allow better reconstruction in presence of a noise in the data. We are also able to incorporate an additional data into the problem, e.g., a partial information on the phase. Moreover, with this additional information our method demonstrates global convergence.

In the future work we shall concentrate on improving the convergence radius of our method. Multi-scale approach may help as a part of our method or as an accellerator of already existing methods. Further improvements of our optimization algorithm may result in a good general algorithm for smooth optimization.

Bibliography

[1] D. H. Brandwood. A complex gradient operator and its application in adaptive array theory. *IEE Proceedings F: Communications Radar and Signal Processing*, 130:11–16, February 1983.

[2] M. A. Fiddy, B. J. Brames, and J. C. Dainty. Enforcing irreducibility for phase retrieval in two dimensions. *Optics Letters*, 8(2):96–98, February 1983.

[3] J. R. Fienup. Phase retrieval algorithms: a comparison. *Applied Optics*, 21(15):2758–2769, 1982.

[4] R. W. Gerchberg and W. O. Saxton. A practical algorithm for the determination of phase from image and diffraction plane pictures. *Optik*, 35:237–246, 1972.

[5] M. Hayes. The reconstruction of a multidimensional sequence from the phase or magnitude of its fourier transform. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on*, 30(2):140–154, 1982.

[6] M. Hayes, Jae Lim, and A. Oppenheim. Signal reconstruction from phase or magnitude. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on*, 28(6):672–680, 1980.

[7] M.H. Hayes and J.H. McClellan. Reducible polynomials in more than one variable. *Proceedings of the IEEE*, 70(2):197–198, 1982.

[8] A. Hjorungnes and D. Gesbert. Complex-Valued matrix differentiation: Techniques and key results. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, 55(6):2740–2746, 2007.

[9] A. Hjorungnes and D. Gesbert. Hessians of scalar functions of complex-valued matrices: A systematic computational approach. In *Signal Processing and Its Applications, 2007. ISSPA 2007. 9th International Symposium on*, pages 1–4, 2007.

[10] A. Hjorungnes, D. Gesbert, and D.P. Palomar. Unified theory of Complex-Valued matrix differentiation. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 3, pages III–345–III–348, 2007.

[11] Jan R. Magnus and Heinz Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. John Wiley, 1999.

[12] J. McClellan and T. Parks. Eigenvalue and eigenvector decomposition of the discrete fourier transform. *Audio and Electroacoustics, IEEE Transactions on*, 20(1):66–74, 1972.

[13] R. P. Millane. Phase retrieval in crystallography and optics. *Journal of the Optical Society of America A*, 7(3):394–411, March 1990.

[14] R. P. Millane. Multidimensional phase problems. *Journal of the Optical Society of America A*, 13(4):725–734, April 1996.

[15] Guy Narkiss and Michael Zibulevsky. Sequential subspace optimization method for Large-Scale unconstrained problems. CCIT 559, Technion, EE Department, 2005.

[16] A.V. Oppenheim and J.S. Lim. The importance of phase in signals. *Proceedings of the IEEE*, 69(5):529–541, 1981.

[17] Mark A. Pfeifer, Garth J. Williams, Ivan A. Vartanyants, Ross Harder, and Ian K. Robinson. Three-dimensional mapping of a deformation field inside a nanocrystal. *Nature*, 442(7098):63–66, July 2006.

[18] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, November 1992.

[19] Avram Sidi. Efficient implementation of minimal polynomial and reduced rank extrapolation methods. *Journal of Computational and Applied Mathematics*, 36(3):305–337, September 1991.

[20] Avram Sidi, William F. Ford, and David A. Smith. Acceleration of convergence of vector sequences. *SIAM Journal on Numerical Analysis*, 23(1):178–196, February 1986.

[21] A. van den Bos. Complex gradient and hessian. *Vision, Image and Signal Processing, IEE Proceedings* -, 141(6):380–383, 1994.

[22] I. A. Vartanyants, I. K. Robinson, J. D. Onken, M. A. Pfeifer, G. J. Williams, F. Pfeiffer, H. Metzger, Z. Zhong, and G. Bauer. Coherent x-ray diffraction from quantum dots. *Physical Review B (Condensed Matter and Materials Physics)*, 71(24):245302–9, June 2005.

[23] D. C. Youla and H. Webb. Image restoration by the method of convex projections: Part 1 theory. *Medical Imaging, IEEE Transactions on*, 1(2):81–94, 1982.

[24] J. M. Zuo, I. Vartanyants, M. Gao, R. Zhang, and L. A. Nagahara. Atomic resolution imaging of a carbon nanotube from diffraction intensities. *Science*, 300(5624):1419–1421, May 2003.