Projected Wirtinger gradient descent for spectral compressed sensing

Suhui Liu

University of Iowa

Copyright © 2017 Suhui Liu

This dissertation is available at Iowa Research Online: https://ir.uiowa.edu/etd/5803

Recommended Citation


Follow this and additional works at: https://ir.uiowa.edu/etd

Part of the Mathematics Commons
PROJECTED WIRTINGER GRADIENT DESCENT FOR SPECTRAL COMPRESSED SENSING

by

Suhui Liu

A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Mathematics in the Graduate College of The University of Iowa

August 2017

Thesis Supervisors: Associate Professor Jianfeng Cai
Assistant Professor Weiyu Xu
This is to certify that the Ph.D. thesis of

Suhui Liu

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Mathematics at the August 2017 graduation.

Thesis Committee:

Jianfeng Cai, Thesis Supervisor

Weiyu Xu, Thesis Supervisor

Weimin Han

Xiaoyi Zhang

Qihang Lin
To my dearest parents.
ACKNOWLEDGEMENTS

There are so many people I would like to thank. First and foremost, I owe my deepest gratitude to my advisor, Professor Jianfeng Cai, who guided me into this wonderful world of compressed sensing. I am grateful for his patient guidance, continued support and steady encouragement to me during my Ph.D. in Iowa. His advice will benefit me the rest of my life.

I want to say special thanks to my co-advisor, Professor Weiyu Xu, besides his patient suggestions about my research, his kindness and encouragement also helped me to overcome difficulties when I doubted myself.

I would also like to thank all the other members of my defense committee: Professor Weimin Han, Professor Xiaoyi Zhang and Professor Qihang Lin, for taking the time to review my work. Their time and input have been greatly appreciated. I would also want to thank Professor Laurent Jay for giving my support for six years.

I would like to say special thanks to my friend, Minli Bao, thank you for bringing me joy in Iowa. Finally and most importantly, I am deeply indebted to my family, particularly my parents and my grandfather, for their unconditional love and support throughout all my endeavors, and for always being by my side throughout all the ups and downs of my life.
ABSTRACT

In modern data and signal acquisition, one main challenge arises from the growing scale of data. The data acquisition devices, however, are often limited by physical and hardware constraints, precluding sampling with the desired rate and precision. It is thus of great interest to reduce the sensing complexity while retaining recovery resolution. And that is why we are interested in reconstructing a signal from a small number of randomly observed time domain samples. The main contributions of this thesis are as follows.

First, we consider reconstructing a one-dimensional (1-D) spectrally sparse signal from a small number of randomly observed time-domain samples. The signal of interest is a linear combination of complex sinusoids at R distinct frequencies. The frequencies can assume any continuous values in the normalized frequency domain [0, 1). After converting the spectrally sparse signal into a low-rank Hankel structured matrix completion problem, we propose an efficient feasible point approach, named projected Wirtinger gradient descent (PWGD) algorithm, to efficiently solve this structured matrix completion problem. We give the convergence analysis of our proposed algorithms. We then apply this algorithm to a different formulation of structured matrix recovery: Hankel and Toeplitz mosaic structured matrix. The algorithms provide better recovery performance; and faster signal recovery than existing algorithms including atomic norm minimization (ANM) and Enhanced Matrix Completion (EMaC). We further accelerate our proposed algorithm by a scheme inspired...
by FISTA. Extensive numerical experiments are provided to illustrate the efficiency of our proposed algorithms. Different from earlier approaches, our algorithm can solve problems of very large dimensions very efficiently. Moreover, we extend our algorithms to signal recovery from noisy samples. Finally, we aim to reconstruct a two-dimension (2-D) spectrally sparse signal from a small size of randomly observed time-domain samples. We extend our algorithms to high-dimensional signal recovery from noisy samples and multivariate frequencies.
Compressed sensing (also known as compressive sensing, compressive sampling, or sparse sampling) is a signal processing technique for efficiently acquiring and reconstructing a signal, by finding solutions to underdetermined linear systems. Traditional signal processing theory demonstrates that the signals can be exactly recovered from a set of uniformly spaced samples taken at the Nyquist rate of twice the highest frequency of the signal. However, due to the limitation of physical and hardware constraints, it is usually hard to get all the samples of a signal, especially for signals with very high frequencies (before normalization). The theory of Compressed Sensing provides a golden opportunity to break the limitation of the sampling theorem in the traditional signal processing theory. Based on Compressed Sensing, compressible signals can be captured and represented at a sampling rate significantly below the Nyquist rate. Compressed Sensing can lead the traditional sampling theorem from Analog-to-Digital conversion to Analog-to-Information conversion.

In Compressed Sensing theory, there are two conditions that the signals can be reconstructed. The first one is sparsity, which requires the signal to be sparse in some domains. The second one is incoherence, which is applied by the isometric property. Compressed Sensing does not only have beautiful theories but also has a tremendous influence on many areas, including applied mathematics, computer science, astronomy, biology, communication, medicine, radar, seismology, to name a few.
# TABLE OF CONTENTS

**LIST OF TABLES** ................................................................. ix

**LIST OF FIGURES** ............................................................... x

**LIST OF ALGORITHMS** ......................................................... xii

**CHAPTER**

1 INTRODUCTION ................................................................. 1

1.1 Classical Parameter Extraction Methods .......................... 3

2 SPECTRAL COMPRESSED SENSING ........................................ 9

2.1 Conventional Compressed Sensing ................................. 9

2.2 Off-The-Grid Compressed Sensing ................................. 12

3 PWGD OF HANKEL MATRIX COMPLETION .............................. 16

3.1 Introduction ............................................................... 16

3.2 Problem Formulation .................................................... 18

3.2.1 Hankel Matrix Completion .................................... 18

3.3 Our algorithms ........................................................... 21

3.3.1 Motivation ............................................................ 21

3.3.2 The Algorithm ...................................................... 23

3.3.3 Acceleration by a FISTA-scheme ............................ 26

3.3.4 The Noisy-PWGD with Bounded Noise ....................... 27

3.4 Numerical Experiments ................................................ 29

3.4.1 Implementation of the Proposed Algorithm: Choosing Para-

rameters $\delta_1, \delta_2$ .................................................. 29

3.4.2 Acceleration of the Proposed Algorithm by a FISTA-like

Scheme ................................................................. 33

3.4.3 Phase Transition in the noiseless setting .................. 36

3.4.4 1D signal of large scale ....................................... 39

3.4.5 The Noisy-PWGD algorithm with Bounded Noise ........ 41

4 PWGD OF HANKEL TOEPLITZ MOSAIC MATRIX COMPLETION 43

4.1 Introduction ............................................................... 43
4.2 Problem Formulation: Hankel and Toeplitz Mosaic Matrix Completion .............................................. 44
4.3 Our Algorithm ................................................................. 45
4.4 Numerical Experiments ....................................................... 47

5 CONVERGENCE THEORY .................................................. 53

5.1 Introduction ................................................................. 53
5.2 Main results ................................................................. 55

6 EXTENSION TO N-D SIGNALS ........................................ 62

6.1 Introduction ................................................................. 62
6.2 2D Complex Signal and Problem Formulation ................. 64
  6.2.1 2D Signal Model ...................................................... 64
  6.2.2 Existing Algorithms in Absence of Noise .................. 66
  6.2.3 Problem Formulation ................................................ 70
6.3 Our Algorithms ............................................................. 73
  6.3.1 Our Algorithm ......................................................... 73
  6.3.2 Acceleration by a FISTA-scheme ............................... 76
  6.3.3 The Noisy-PWGD2D with Bounded Noise .................. 77
  6.3.4 Numerical Experiment: The Reconstruction of 2D Noisy Signal ...................................................... 79
6.4 Extension to Higher-Dimensional Models ....................... 80

7 CONCLUSION ............................................................... 81

REFERENCES ................................................................. 83
LIST OF TABLES

Table

3.1 Elpsed time(second) for the signals with different parameters. . . . . . . 30

3.2 The table clearly show that the speed of our projected Wirtinger methods outperform those of ANM and EMaC. . . . . . . . . . . . . . . . . . . 41

4.1 The table clearly show that the speed of our projected Wirtinger methods outperform those of ANM and EMaC. . . . . . . . . . . . . . . . . . . 52
LIST OF FIGURES

Figure

1.1 Identify Frequencies. ......................................................... 3

2.1 Basis Mismatch. ............................................................. 11

3.1 An example of 1-dimensional signal reconstruction. ................. 17

3.2 The effect of the parameter $\delta_1$ and $\delta_2$ on the convergence of the proposed algorithm PWGD for the signals. From top to bottom: (1) The signal with $N = 51, R = 1, M = 10$; (2) The signal with $N = 51, R = 3, M = 20$. 31

3.3 The effect of the parameter $\delta_1$ and $\delta_2$ on the convergence of the proposed algorithm PWGD for the signals. From top to bottom: (1) The signal with $N = 501, R = 5, M = 100$; (2) The signal with $N = 501, R = 8, M = 200$. 32

3.4 Convergence rate comparison between PWGD and PWGD-FISTA. From top to bottom: (1) The signal with $N = 51, R = 1, M = 10$; (2) The signal with $N = 51, R = 3, M = 20$. ......................................................... 34

3.5 Convergence rate comparison between PWGD and PWGD-FISTA. From top to bottom: (1) The signal with $N = 501, R = 8, M = 200$; (2) The signal with $N = 5001, R = 20, M = 1000$. ......................................................... 35

3.6 Phase transition of PWGD-FISTA. ........................................ 37

3.7 Phase transition of PWGD. ............................................... 37

3.8 Phase transition of ANM with separation condition $\Delta f > \frac{1.5}{\|N\|}$. ................. 38

3.9 Phase transition of ANM without separation condition. ............. 38

3.10 Phase transition of EMaC algorithm. ...................................... 39

3.11 Perfect recovery of a large scale signal with $N = 5001, R = 40, M = 2000$. 40

3.12 NMSE vs. SNR when the signals where frequencies are generated randomly for different algorithms. From top to bottom: (1) The signal with $N = 81, R = 2, M = 30$; (2) The signal with $N = 81, R = 2, M = 50$. .... 42
4.1 Phase transition of PWGD-HTM. .................................................. 48
4.2 Phase transition of PWGD. ......................................................... 49
4.3 Phase transition of ANM. ............................................................... 49
4.4 Phase transition of EMaC. ............................................................... 50
4.5 The probability of signal recovery when \((2N - 1, R) = (127, 40)\). ....... 51

6.1 The performance of noisy PWGD2D for a 101 \times 101 data matrix that contains 2 random frequencies. 0.8 of all entries are observed with signal-to-noise ratio is 10. Here, the reconstructed data against the true data for the first 500 time instances (after vectorized) are plotted. .......................... 79
# LIST OF ALGORITHMS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Alternating Projection Algorithm</td>
<td>22</td>
</tr>
<tr>
<td>3.2 Dykstra’s Projection Algorithm</td>
<td>23</td>
</tr>
<tr>
<td>3.3 Projected Gradient Descent Algorithm</td>
<td>24</td>
</tr>
<tr>
<td>3.4 Projected Wirtinger Gradient Descent Algorithm</td>
<td>25</td>
</tr>
<tr>
<td>3.5 FISTA-PWGD</td>
<td>27</td>
</tr>
<tr>
<td>4.1 Projected Wirtinger Gradient Descent Algorithm for HTM</td>
<td>46</td>
</tr>
<tr>
<td>6.1 2-D Projected Gradient Descent Algorithm</td>
<td>74</td>
</tr>
<tr>
<td>6.2 2-D Projected Wirtinger Gradient Descent Algorithm</td>
<td>75</td>
</tr>
<tr>
<td>6.3 FISTA-PWGD2D</td>
<td>77</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

In general, a signal describes how some physical quantity varies over time and/or space, and signal processing involves manipulating a signal to change its basic characteristics or extract information. In more specific terms, digital signal processing is the processing of digitized discrete-time sampled signals, for example, you can apply low pass filters to remove high frequencies from digital signals; you can also cancel unwanted noise. Moreover, you can compress the signal and then reconstruct.

Reconstructing a signal from a series of sampling measurements is a common topic in signal processing, which has numerous practical applications in radar imaging [30], wireless communication [16], seismology [5], fluorescence microscopy [35], nuclear magnetic resonance (NMR) spectroscopy [27] and so on. Because of the constraints imposed by sampling hardware and physical measurement conditions, sometimes we can only obtain partial information, instead of full information, of a signal. For example, when we try to infer the frequency components of a signal, we may only get a small number of discrete time domain samples of a signal.

In this thesis, we investigate the spectral compressed sensing problem, which aims to recover spectrally sparse signals from a small number of discrete time samples and identify frequencies. Here we say a signal is spectrally sparse if it is a weighted sum of 1-dimensional(1-D) complex sinusoids at, say, $R$ distinct continuous frequencies in the unit interval $[0, 1)$.

Specifically, let $x^{(\text{true})}(t)$ be a spectrally sparse signal expressed as a sum of $R$
complex exponentials as follows:

$$x^{(\text{true})}(t) = \sum_{k=1}^{R} d_k^{(\text{true})} e^{(2\pi i f_k^{(\text{true})} + \alpha_k^{(\text{true})})t}$$  \hspace{1cm} (1)

$$= \sum_{k=1}^{R} |d_k^{(\text{true})}| e^{i\phi_k^{(\text{true})}} e^{(2\pi i f_k^{(\text{true})} + \alpha_k^{(\text{true})})t}$$

$$= \sum_{k=1}^{R} |d_k^{(\text{true})}| a(f_k^{(\text{true})}, \phi_k^{(\text{true})})_t, t \in \mathcal{N}.$$  \hspace{1cm} \hspace{2cm}

Here $f_k^{(\text{true})}$ represents an unknown frequency which is normalized in $[0,1)$, the coefficient $d_k^{(\text{true})} = |d_k^{(\text{true})}| e^{i\phi_k^{(\text{true})}}$ is the complex amplitude, $\alpha_k^{(\text{true})} > 0$ is the damping factor, $\phi_k^{(\text{true})} \in [0,2\pi)$ is its phase. $\mathcal{N} = \{t : 0 \leq t \leq 2N - 2\}$ is the set of time indices. Here $a(f_k^{(\text{true})}, \phi_k^{(\text{true})}) \in \mathbb{C}^{|\mathcal{N}|}$ is a frequency-atom, with $t$-th element $a(f_k^{(\text{true})}, \phi_k^{(\text{true})})_t = e^{i((2\pi f_k^{(\text{true})} + \alpha_k^{(\text{true})})t + \phi_k^{(\text{true})})}$. When the phase is 0, we denote the frequency-atom simply by $a(f_k^{(\text{true})})$. This signal model covers signals in various applications, for example, in acceleration of medical imaging [29], analog-to-digital conversion [40], and inverse scattering in seismic imaging [5].

We denote the underlying uniformly sampled true signal as

$$x^{(\text{true})} = \sum_{k=1}^{R} d_k^{(\text{true})} a(f_k^{(\text{true})}, \phi_k^{(\text{true})}) \hspace{1cm} (2)$$

$$= [x^{(\text{true})}(0), x^{(\text{true})}(1), \ldots, x^{(\text{true})}(2N - 2)]^T \in \mathbb{C}^{2N-1}.$$  \hspace{2cm}

We assume that the signal is observed over the time index set $\mathcal{M} \subseteq \mathcal{N}$ of cardinality $|\mathcal{M}| = M \leq 2N - 1$, where $M$ observations are chosen randomly. Our goal is to reconstruct the true vector $x^{(\text{true})}$ from

$$y = x^{(\text{true})}_\mathcal{M} := \{x_m^{(\text{true})} \mid m \in \mathcal{M}\}. \hspace{1cm} (3)$$
1.1 Classical Parameter Extraction Methods

A long time ago, people began to develop parameter extraction methods. In 1795, Gaspard Riche de Prony developed a method, now well-known as Prony’s analysis (Prony’s method) [31]. Prony’s method extracts valuable information from a uniformly sampled signal and builds a series of damped complex exponentials or sinusoids. This allows for the estimation of frequency, amplitude, phase and damping components of a signal. Specifically, assume that there are $2N - 1$ complex data samples $x(0), x(1), ..., x(2N - 2)$, the signal model is given before:

$$x^{(true)}(t) = \sum_{k=1}^{R} |d_{k}^{(true)}| e^{i\phi_{k}^{(true)}} e^{(2\pi i f_{k}^{(true)} + \alpha_{k}^{(true)})t}$$

$$= \sum_{k=1}^{R} h_{k} z_{k} t.$$ 

where $t = 0, ..., 2N - 1$, $h_{k} = |d_{k}^{(true)}| e^{i\phi_{k}^{(true)}}$, $z_{k} = e^{(\alpha_{k}^{(true)} + 2\pi i f_{k}^{(true)})}$. The $R$ equations of the above signal model may be expressed in matrix form.
as:
\[
\begin{bmatrix}
  z_1^1 & z_2^1 & \cdots & z_R^1 \\
  z_1^2 & z_2^2 & \cdots & z_R^2 \\
  \vdots & \vdots & \ddots & \vdots \\
  z_1^R & z_2^R & \cdots & z_R^R
\end{bmatrix}
\begin{bmatrix}
  h_1 \\
  h_2 \\
  \vdots \\
  h_R
\end{bmatrix}
= \begin{bmatrix}
  x(1) \\
  x(2) \\
  \vdots \\
  x(R)
\end{bmatrix}.
\]

The matrix equation represents a set of linear equations that can be solved for the unknown vector of amplitudes.

Prony proposed to define the polynomial that has the $z_k$ exponents as its roots:

\[
F(z) = \prod_{k=1}^{R} (z - z_k) = (z - z_1)(z - z_2)\ldots(z - z_R).
\]

The polynomial may be represented as the sum:

\[
F(z) = \sum_{r=0}^{R} a_r z^{R-r} = a_0 z^R + a_1 z^{R-1} + \ldots + a_{R-1} z + a_R.
\]

Shifting the index on (4) from $t$ to $t - r$ and multiplying by the parameter $a_r$ yield:

\[
a_r x(t - r) = a_r \sum_{k=1}^{R} h_k z_k^{t-r},
\]

then we have

\[
\sum_{r=1}^{R} a_r x(t - r) = \sum_{k=1}^{R} h_k z_k^{t-R} \left( \sum_{r=1}^{R} a_r z_k^{R-r} \right).
\]

The right-hand summation in the above equation may be recognize as polynomial $F(z)$, evaluated at each of its roots $z_k$ yielding the zero result:

\[
\sum_{r=1}^{R} a_r x(t - r) = 0.
\]

The equation can be solved for the polynomial coefficients. In the second step the roots of the polynomial $F(z)$ can be calculated. The damping factors and frequencies may be determined from the roots $z_k$. We refer the readers to [28] for more details.
The matrix pencil method is another conventional signal reconstruction method for estimating parameters (frequencies, amplitudes and phase angles) of exponentially sinusoids \[34\]. The details are described as follows: First the signal model is the same as the signal model given in the beginning. \( x(t) \) is a sum of \( R \) damped complex exponentials.

\[
x^{(\text{true})}(t) = \sum_{k=1}^{R} |d_k^{(\text{true})}| e^{i\phi_k^{(\text{true})}} e^{(2\pi if_k^{(\text{true})}+\alpha_k^{(\text{true})})t} \tag{4}
\]

\[
= \sum_{k=1}^{R} |d_k^{(\text{true})}| e^{i\phi_k^{(\text{true})}} z_k^t, t = 0, 1, ..., 2N - 2. \tag{5}
\]

Here \( z_k = e^{2\pi if_k^{(\text{true})}+\alpha_k^{(\text{true})}} \).

Matrix pencil method finds the estimates for the values of \( d_k' \)'s, \( \phi_k' \)'s and \( z_k' \)'s from the measured data \( x(t) \). It does this by a two-step process. First, it finds the poles \( z_k' \)'s as the solution of a generalized eigenvalue problem by using a mathematical entity known as the matrix pencil. This matrix pencil is formed using the sampled values of \( x(t) \). In the second step, it then uses the new found poles to estimate the complex amplitudes, \( A_k' \)'s and phase angles, \( \phi_k' \)'s by solving a least squares problem.

For a discrete signal of length \( 2N-1 \), \( x(t) \), we can define two \((2N-1-L) \times L\) matrices \( Y_1 \) and \( Y_2 \), as the following:

\[
Y_1 = \begin{bmatrix}
    x(0) & x(1) & \ldots & x(L-1) \\
    x(1) & x(2) & \ldots & x(L) \\
    \vdots & \vdots & \ddots & \vdots \\
    x(2N-L-2) & x(2N-L-1) & \ldots & x(2N-3)
\end{bmatrix},
\]

\[
Y_2 = \begin{bmatrix}
    x(1) & x(2) & \ldots & x(L) \\
    x(2) & x(3) & \ldots & x(L+1) \\
    \vdots & \vdots & \ddots & \vdots \\
    x(2N-L-1) & x(2N-L) & \ldots & x(2N-2)
\end{bmatrix}
\]
where $L$ is referred to as the pencil parameter.

The parameters of the complex exponentials, $z_k'$s can be found as the generalized eigenvalues of the matrix pencil $Y_2 - \lambda Y_1$ [34].

Next the matrix $Y$ is constructed.

\[
Y = \begin{bmatrix}
  x(0) & x(1) & \ldots & x(L) \\
  x(1) & x(2) & \ldots & x(L+1) \\
  \vdots & \vdots & \ddots & \vdots \\
  x(2N - L - 2) & x(2N - L - 1) & \ldots & x(2N - 2)
\end{bmatrix}.
\]

Then we take the SVD of $Y$,

\[
Y = USV^*.
\]

This SVD operation estimates the order, $R$ of the signal $x(t)$. Also we have

\[
Y_1 = U_1 S_1 V_1^*,
\]

\[
Y_2 = U_2 S_2 V_2^*,
\]

and it can be shown that the eigenvalues, $z_k'$s of the matrix pencil $Y_2 - \lambda Y_1$ can be estimated by the those of $V_1^\dagger V_2$, where $V_1^\dagger$ is the Moore-Penrose pseudo-inverse of $V_1$. The damping factors, $\alpha_k'$s and frequencies, $f_k'$s are then determined from $z_k'$s.

With the eigenvalues found, the amplitudes $d_k'$s and phase angles $\phi_k'$s are then found by solving the least squares problem as shown in the following equation:

\[
\begin{bmatrix}
  x(0) \\
  x(1) \\
  \vdots \\
  x(2N - 2)
\end{bmatrix} = \begin{bmatrix}
  1 & 1 & \ldots & 1 \\
  z_1 & z_2 & \ldots & z_R \\
  \vdots & \vdots & \ddots & \vdots \\
  z_1^{(2N-2)} & z_2^{(2N-2)} & \ldots & z_R^{(2N-2)}
\end{bmatrix} \begin{bmatrix}
  h_0 \\
  h_1 \\
  \vdots \\
  h_R
\end{bmatrix},
\]

where $z_k = e^{2\pi if_k^{(true)} + \alpha_k^{(true)}}$ and $h_k = |d_k^{(true)}| e^{i\phi_k^{(true)}}$.

We refer the readers to [34] for more details.
Estimation of signal parameters via rotational invariant techniques (ESPRIT) [33], and the Tufts and Kumaresan approach [41] are another two well-known classical parameter extraction methods. We will not describe the details of the methods here. To summarize, early methods need time domain samples on uniformly sampled integer time points. More specifically, they use the following $2N - 1$ samples in the time domain $x(t), t = 0, 1, ..., 2N - 2$; and then, in order to get the frequencies of $x(t)$, these early methods used linear algebra techniques involving linear structured matrices such as Hankel and Toeplitz matrices. However, due to physical measurement limitations, it is usually hard to get all the $2N - 1$ samples of $x(t), t = 0, 1, ..., 2N - 2$, especially for signals with very high frequencies (before normalization) [40].

All these classical parameter extraction methods followed the famous Nyquist-Shannon sampling theorem, which demonstrates that the sampling rate must be twice the highest frequency. In the framework of the Nyquist-Shannon sampling theory, the process of acquiring information from the signal is inevitably to go through the following five steps: sample, compress, transmit, decompress and process, the sampling is an important bridge between the analog of real signals and digitalization of processed signals. On one hand, from the point of view of the analog signal, Nyquist-Shannon sampling theorem determines that the sampling frequency must be at least twice the highest frequency of the signal, so that the spectral aliasing does not occur and the signal is perfectly reconstructed. On the other hand, from the point of view of digital signals, the number of sample in the time domain must be the same as the numbers in the frequency domain, in order to reconstruct signals perfectly. Although
this theorem guarantees the reconstruction of the signal, the requirement of collection of signals is too high. For example, the sampling of high-frequency signals requires expensive high frequency acquisition devices in medical MRI, the required time for obtaining data is also high. To reduce the load of the channel and store, the collected signal must be compressed and decompressed before the signal is used. These two processes need very strong ability of computing. The process of compressing and decompressing shows that the data derived from the theorem is very redundant. Moreover, one more weakness of these techniques is that they require the number of frequencies of the signal or at least an estimate of it. Besides, these techniques are often sensitive against noise and outliers [20]. As a result, the researchers began to explore whether there is a way to get information from signals directly, in order to save a large number of sampling and computing resources.
2.1 Conventional Compressed Sensing

Compressive sensing (CS) – also known under the terminology of compressed sensing, compressive sampling or sparse recovery – is another line of work in signal reconstruction, it shows that if the signal is sparse over some transform domain, the signal may be reconstructed with far fewer samples than the Nyquist sampling theorem requires. This area was initiated in 2006 by two innovative papers, namely by Donoho [19] and by Candes, Romberg, and Tao [10]. It has a tremendous influence on many areas, including applied mathematics, computer science, astronomy, biology, medicine, radar, seismology, to name a few [21].

To apply the theory of CS, we first discretize the frequency domain $[0, 1)$ by uniform grid $\mathcal{G}$ with meshsize $1/(2N - 1)$. Assume all frequencies $f^{(true)}_k$, $k = 1, \ldots, R$, are on the grid $\mathcal{G}$. Then, the discrete signal $x^{(true)}$ can be written as $x^{(true)} = Fc$, where $F$ is the discrete Fourier transform (DFT) matrix of order $2N - 1$, and $c \in \mathbb{C}^{2N-1}$ is a sparse vector with non-zero entries at indices $(2N - 1)f_k$'s. Then, the samples (3) can be written as $y = F_Mc$, where $F_M$ are partial rows of $F$. Equivalently, our goal has turned into recovering the sparse vector $c$. According to the theory of compressed sensing [10], when $M$ is uniformly randomly drawn from all subsets of $\{0, 1, \ldots, 2N - 2\}$ with cardinality $M$, the sparse vector $c$ (hence $x^{(true)}$)
can be recovered exactly with high probability by solving

\[
\min_{\mathbf{c}} \|\mathbf{c}\|_1 \quad \text{s.t.} \quad \mathbf{F_Mc} = \mathbf{y},
\]

provided \( M \geq O(K \log N) \). Efficient algorithms for solving (1) include Bregman iterations \([6] [7] [42]\) and iterative soft-thresholding algorithms \([2] [18]\). When the frequencies \( f_{k}^{(\text{true})} \) are not on the grid \( \mathcal{G} \), we expect to have a good approximation of \( \mathbf{x}^{(\text{true})} \) by solving (1), as the differences between the true frequencies and the grid \( \mathcal{G} \) can be as small as \( O(1/N) \). Unfortunately, this discretization method can lead to large recovery errors \([15]\). This phenomena is known as basis mismatch of compressed sensing. In basis mismatch, we will have non-negligible signal recovery errors resulting from the impact of discretization errors on CS signal recovery procedures, unless we make very fine grid discretization to reduce signal recovery error, but this leads to an undesirably large dictionary for signal recovery \([36]\). To overcome this limitation, we will consider frequencies \( f_{k}^{(\text{true})} \)'s on the continuous domain \([0, 1)\) instead of discretizing it with a meshsize of \( \frac{1}{2N-1} \).
Matrix completion (MC) bears some similarity with CS. Here, the goal is to recover a low-rank matrix from a small fraction of linear measurements. More specifically, one would want to recover the data matrix by solving the optimization problem

$$\min \quad \text{rank}(X)$$

s.t. \quad X_{ij} = M_{ij}, \; (i, j) \in \Omega,$$

(2)

However, this is unfortunately of little practical use because this optimization problem is not only NP-hard, but all known algorithms which provide exact solutions require time doubly exponential in the dimension $N$ of the matrix in both theory and practice [17].

If a matrix has rank $r$, $r$ is very small, then it has exactly $r$ nonzero singular
values so that the rank function in 4.2 is simply the number of non-vanishing singular values. The authors of [9] proposed recovering the unknown matrix by solving the nuclear norm minimization problem

$$\min \|X\|_*$$

s.t. \(X_{ij} = M_{ij}, \ (i,j) \in \Omega,\) (3)

where the nuclear norm \(\|X\|_*\) of a matrix \(X\) is defined as the sum of its singular values \(\|X\|_* := \sum_{i=1}^{r} \sigma_i(X)\). It has been shown that when the unknown matrix is low rank and also satisfies some strong incoherence assumption, then the unknown matrix can be reconstructed exactly from only about \(O(rN\log N)\) sampled entries by nuclear norm minimization, where \(r\) and \(N\) are respectively the rank and dimension of the unknown matrix.

### 2.2 Off-The-Grid Compressed Sensing

Recently there have been growing interests in designing new algorithms which can recover the continuous valued parameters precisely even from a small number of discrete non-uniform time samples. In [8], the authors proposed to use total variation minimization to find the continuous-valued frequencies from equi-spaced samples.

In [37], motivated by atomic norm minimization [11], the authors used atomic norm minimization to recover signal frequencies from nonuniform samples. Specifically, the authors propose the following atomic norm minimization to recover a spec-
trally sparse signal $\mathbf{x}^{(\text{true})}$ using randomly observed time domain samples $\mathcal{M} \subseteq \mathcal{N}$:

$$ \min_{\mathbf{x}} \| \mathbf{x} \|_A $$

s.t. 

$$ \mathbf{x}_m = \mathbf{x}_m^{(\text{true})}, m \in \mathcal{M} $$

where $\| \mathbf{x} \|_A$ represents the atomic norm of a signal $\mathbf{x}$, defined as

$$ \| \mathbf{x} \|_A = \inf \{ \sum_k |d_k| : \mathbf{x} = \sum_k d_k a(f_k) \}. $$

The atomic norm minimization (4) can be cast as an SDP as follows:

$$ \min_{u, \mathbf{x}, t} \frac{1}{2|\mathcal{N}|} \text{Tr}(\text{Toep}(\mathbf{u})) + \frac{1}{2} t $$

s.t. 

$$ \begin{bmatrix} \text{Toep}(\mathbf{u}) & \mathbf{x} \mathbf{x}^H \end{bmatrix} \succeq 0, $$

$$ \mathbf{x}_m = \mathbf{x}_m^{(\text{true})}, m \in \mathcal{M} $$

where $\text{Tr}(\cdot)$ is the trace function, and $\text{Toep}(\mathbf{u})$ is a Hermitian Toeplitz matrix whose first column is equal to $\mathbf{u}$, here

$$ \mathbf{u} = \sum_{k=1}^R d_k^{(\text{true})} a(f_k^{(\text{true})}, 0), t = \sum_{k=1}^R d_k^{(\text{true})}. $$

We refer the reader to [37] for more details. The atomic norm minimization requires a certain minimum separation between frequencies $\Delta_f = \frac{1}{(|\mathcal{N}| - 1)/4}$ for successful recovery.

In [8] and [37], the authors convert the signal frequency recovery into a low-rank Toeplitz matrix completion problem. In [13], the problem of recovering signal frequencies from nonuniform samples is formulated as a low-rank Hankel matrix completion problem, inspired by Pronys method and the matrix pencil method, the authors of [13] called the algorithm Enhanced Matrix Completion (EMaC). The EMaC
formulation is stated as follows:

\[
\min_x \|Hank(x)\|_* \\
\text{s.t.} \quad x_m = x_m^{(true)}, m \in M,
\]

where \([Hank(x)]_{jk} := x_{j+k}\) and \(\| \cdot \|_*\) represents the nuclear norm of a matrix, which is the sum of the singular values of the matrix. This convex program can be further rewritten as a semidefinite program (SDP):

\[
\min_x \frac{1}{2} \text{Tr}(Q_1) + \frac{1}{2} \text{Tr}(Q_2) \\
\text{s.t.} \quad U_2 = \begin{bmatrix} Q_1 & Hank(x) \\ Hank(x)^H & Q_2 \end{bmatrix} \succeq 0, \\
x_m = x_m^{(true)}, m \in M.
\]

Let \(Hank(x) = U\Sigma V^H\) be the singular value decomposition of the matrix \(Hank(x)\), and the matrices \(Q_1 := U\Sigma U^H\) and \(Q_2 := V\Sigma V^H\). We refer the readers to [32] for more details.

Though robust signal recovery is guaranteed theoretically through these methods in [37], [8], [13], convex optimization based low-rank structured matrix completions are not computationally efficient- the resulting optimization problems contain \(O(N^2)\) unknowns explicitly, where \(N\) is the dimension of signal. To solve the resulting matrix completion problems, off-the-shelf algorithms such as SDPT3 [38] use interior point methods which requires computing a Hessian matrix of size \(O(N^4)\) in its Newton step. First-order methods, such as alternating direction method of multipliers (ADMM) and proximal point algorithm (PPA), need a dual matrix that is unstructured [22], and, consequently, these algorithms require memory of size \(O(N^2)\), which
implies that these convex optimization approaches are not suitable for recovering signals of large dimensions.

Therefore, our object is to seek an algorithm that works for large scale signals and the frequencies of the signals should be continuous in [0, 1).
CHAPTER 3
PWGD OF HANKEL MATRIX COMPLETION

3.1 Introduction

In this chapter, our aim is to reconstruct a one-dimensional (1-D) spectrally sparse signal from a small number of randomly observed time-domain samples. The signal of interest is a linear combination of complex sinusoids at R distinct frequencies. The frequencies can assume any continuous values in the normalized frequency domain [0, 1).

There are several existing algorithms that work for the signals. In [37], Tang et al. introduced off-the-grid compressed sensing that employs atomic norm minimization to recover sparse continuous-valued frequencies from few randomly chosen nonuniformly-spaced time samples. Atomic norm minimization employs semidefinite programs (SDP) for recovery. Enhanced matrix completion (EMaC) [13] is another method that is able to reconstruct signals with frequencies taking continuous values. Enhanced matrix completion method converts the signal recovery problem to a Hankel matrix completion problem. Then the authors employ nuclear norm minimization for the structured matrix completion problem to recover spectrally sparse signals. This convex program can also be rewritten into a semidefinite program (SDP). Solving these SDPs is of high complexity, making it difficult to efficiently recover spectrally sparse signals of large scale and of high dimensions.

To efficiently recover large-scale signals, we propose a projected Wirtinger
gradient descent (PWGD) method for low-rank Hankel matrix completion. Instead of solving a convex relaxation of the low rank Hankel matrix completion problem, we directly deal with the non-convex low rank structured matrix completion problem. Our proposed PWGD algorithm is a feasible point algorithm, and it uses $O(NR)$ memory. Since the number of sinusoids, $R$, is usually much smaller than $N$, the proposed algorithm provides efficient large scale signal recovery. To speed up our proposed algorithm, an acceleration technique scheme similar to FISTA [2] is given. The practical applicability of our algorithm is validated by numerical experiments, which show that our algorithms can recover high-dimensional signals as a superposition of multiple sinusoids.

![Figure 3.1. An example of 1-dimensional signal reconstruction.](image)

This chapter is organized as follows. In section 1, we formulate our non-convex optimization problem based upon Hankel matrix completion. In section 2, we first
developed several heuristic algorithms based on our optimization problem without theorem guarantees. Then, we propose our algorithm named projected Wirtinger gradient descent (PWGD) to recover the 1D spectrally sparse signals. In section 3, we further accelerate our proposed algorithm by a scheme inspired by FISTA [2].

**Notations:** We denote the set of complex numbers as \( \mathbb{C} \). We reserve calligraphic uppercase letters for index sets. When we use an index set \( K \) as the subscript of a vector \( \mathbf{x} \), \( \mathbf{x}_K \), it represents the part of the vector \( \mathbf{x} \) over index set \( K \). We use the superscripts \( H, T, \) and \( * \) for matrix conjugate transpose, transpose, and conjugate respectively. For \( \mathbf{x} = [x_1, x_2, ..., x_{2n_1}]^T \), the Hankel matrix over \( \mathbf{x} \), denoted \( \text{Hank}(\mathbf{x}) \), is the Hankel matrix with first column equal to \( [x_1, ..., x_n]^T \) and last row equal to \( [x_n, ..., x_{2n_1}] \). The Hermitian Toeplitz matrix over \( \mathbf{x} \), denoted \( \text{Toep}(\mathbf{x}) \), has its first column equal to \( \mathbf{x} \). We write the Frobenius norm of a matrix \( A \in \mathbb{C}^{m \times n} \) as \( \|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |A_{ij}|^2} \), where \( A_{ij} \) is the element of \( A \) in its \( i \)-th row and \( j \)-th column.

### 3.2 Problem Formulation

#### 3.2.1 Hankel Matrix Completion

Our proposed algorithm first converts the reconstruction of \( \mathbf{x}^{(\text{true})} \) from (2) in the introduction to a Hankel matrix completion problem. This strategy has also been used in [13].

Let \( \mathcal{H} \) be a linear operator that maps a vector in \( \mathbb{C}^{2N-1} \) to a \( N \times N \) Hankel
matrix as follows

\[ \mathcal{H} : x \in \mathbb{C}^{2N-1} \rightarrow \mathcal{H}x \in \mathbb{C}^{N \times N}, \]

\[ [x_0, x_1, \ldots, x_{2N-2}]^T \rightarrow \begin{bmatrix} x_0 & x_1 & \ldots & x_{N-1} \\ x_1 & x_2 & \ldots & x_N \\ \vdots & \vdots & \ddots & \vdots \\ x_{N-1} & x_N & \ldots & x_{2N-2} \end{bmatrix}, \]

that is,

\[ H_{jk} = [\mathcal{H}x]_{jk} = x_{j+k}, \quad 0 \leq j, k \leq N - 1. \]

Define \( \mathbf{H}^{(\text{true})} = \mathcal{H}x^{(\text{true})}. \) It can be checked that the rank of \( \mathbf{H}^{(\text{true})} \) is equal to the number of frequencies \( R \), due to the following factorization

\[ \mathcal{H}x^{(\text{true})} = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ y_1 & y_2 & \ldots & y_{1N-1} \\ \vdots & \vdots & \ddots & \vdots \\ y_1^{N-1} & y_2^{N-1} & \ldots & y_{RN-1} \end{bmatrix} \begin{bmatrix} d_1^{(\text{true})} \\ d_2^{(\text{true})} \\ \vdots \\ d_R^{(\text{true})} \end{bmatrix} = \begin{bmatrix} 1 & y_1 & \ldots & y_{1N-1} \\ 1 & y_2 & \ldots & y_{2N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & y_R & \ldots & y_{RN-1} \end{bmatrix}. \]

Here \( y_k := e^{2\pi i f_k^{(\text{true})}} \).

Then, instead of constructing the true signal \( x^{(\text{true})} \) directly, we reconstruct the rank-\( R \) Hankel matrix \( \mathcal{H}x^{(\text{true})} \). Since \( H \) is one-to-one from a vector in \( \mathbb{C}^{2N-1} \) to an \( N \times N \) Hankel matrix, one can easily convert the reconstructed Hankel matrix back to a signal.

Instead of solving the convex optimization problem like (6), the signal reconstruction problem is formulated as

\[ \text{Find} \quad \text{a matrix } \mathbf{X} \]

\[ \text{s.t} \quad \text{rank}(\mathbf{X}) \leq R, \]

\[ X_{jk} = H_{jk}^{(\text{true})}, \quad (j, k) \in \Omega, \]

\[ \mathbf{X} \text{ is Hankel}, \]
where

\[
\Omega = \{(j, k) : j + k \in \mathcal{M}, 0 \leq j, k \leq N - 1\}
\]

is the positions of known entries in \(H^{(\text{true})}\). Since \(H\) is one-to-one from \(\mathbb{C}^{2N-1}\) to the set of all \(N \times N\) Hankel matrix, reconstructing \(x^{(\text{true})}\) is equivalent to reconstructing \(H^{(\text{true})}\). The problem (1) is referred to a low-rank Hankel matrix completion.

Following generic low-rank matrix completion [9], in [13], (1) is converted to a rank minimization problem and further relaxed to (6). It was shown that, if \(\mathcal{M}\) is uniformly randomly drawn from all subsets of \(\mathcal{N} = \{0, 1, \ldots, 2N - 2\}\) with cardinality \(M \geq O(R \log^4 N)\), and certain separation conditions between frequencies are satisfied, then the solution of (6) recover \(H^{(\text{true})}\) perfectly with dominant probability.

Though (6) is a convex optimization problem, there were no efficient ways to compute it for large problem dimensions. The unknown matrix is \(N \times N\), compared with \(O(N)\) unknowns in \(x^{(\text{true})}\). One may convert (6) to the SDP (5) and then employ available packages such as SDPT3 [39]. However, these packages use second-order methods, which require solving a huge linear system of order \(O(N^2) \times O(N^2)\) at each step. Also, it is not straightforward to adapt nuclear norm minimization algorithms for generic low-rank matrix completion to solving (6), as the Hankel constraint invokes \(O(N^2)\) linear equality constraints.

In this thesis, instead of considering convex optimizations (6), we aim at attacking the original non-convex problem (1) directly. Non-convex algorithms has been proven to have the advantage of fast convergence in sparsity and low-rank reconstruction [3] [26]. We propose an efficient algorithm based on projected Wirtinger gradient
descent for this particular spectrally sparse signal recovery problem.

### 3.3 Our algorithms

#### 3.3.1 Motivation

Our signal reconstruction problem can be converted to the non-convex problem (1), that is to say, our aim is to find a matrix $X$ such that its rank is no greater than $R$ and it is a Hankel matrix with consistent observed data. In order to solve this problem, the simplest algorithm uppermost in the mind should be alternating projection method.

First, let us define the set of all complex-valued matrices with rank no greater than $R$ as

$$
R^R_C = \{ L \in \mathbb{C}^{N \times N} | \text{rank}(L) \leq R \}. 
$$

Similarly, define the set of all complex-valued Hankel matrices that are consistent with the observed data

$$
H = \{ Hx | x \in \mathbb{C}^{2N-1}, x_M = x^{(\text{true})}_M \}.
$$

The set $R^R_C$ is an algebraic variety [12] and $H$ is an affine space.

Then, the signal recovery problem and (1) can be formulated as following optimization problem

$$
\min_{L \in R^R_C, H \in H} F(L,H) := \frac{1}{2} \| L - H \|_F^2.
$$

Given initial estimate $H_0 = Hx^0$ and $L_0 = 0$, the iterative scheme of alter-
nating projection algorithm has the form

\[
H_{t+1} = \mathcal{P}_\mathcal{H}(\mathcal{P}_\mathcal{R}(H_t))
\]

\begin{algorithm}
\textbf{Algorithm 3.1 Alternating Projection Algorithm}
\begin{enumerate}
\item \textbf{Input:} The observed signal $x_0^{(true)}$ on the location set $\mathcal{M}$.
\item \textbf{Initialize:} Form a Hankel matrix $H_0 \in \mathcal{H}$.
\item \textbf{Repeat:}
\begin{enumerate}
\item $L_{t+1} \in \mathcal{P}_\mathcal{R}(H_t)$
\item $H_{t+1} \in \mathcal{P}_\mathcal{H}(L_{t+1})$
\end{enumerate}
\item \textbf{until:} the stopping criteria is met
\item \textbf{Output:} the vector $\tilde{x}$ of anti-diagonals of $H_{t+1}$.
\end{enumerate}
\end{algorithm}

Moreover, the alternating projection algorithm of my problem can be modified in a different way, called Dykstra’s projection algorithm. It is a variant of the alternating projection algorithm. In its simplest form, the method finds a point in the intersection of two nonempty closed sets by iteratively projecting onto each of the set; the difference between the alternating projection algorithm and Dykstra’s projection algorithm is that there are intermediate steps.
Algorithm 3.2 Dykstra’s Projection Algorithm

1. **Input:** The observed signal $x_0^{\text{(true)}}$ on the location set $\mathcal{M}$.
2. **Initialize:** Form a Hankel matrix $H_0 \in \mathcal{H}$, $L_0 = 0$, and the intermediate matrix $U_0 = V_0 = 0$.
3. **Repeat:**
   For each $t = 0, 1, \ldots$ generate a sequence $\{L_t, H_t\}_{t \in \mathbb{N}}$ as follows:
   - **Step 1:** $L_{t+1} \in \mathcal{P}_{\mathbb{R}^R} (H_t + U_t)$
   - **Step 2:** $U_{t+1} = H_t + U_t - L_t$
   - **Step 3:** $H_{t+1} \in \mathcal{P}_{\mathcal{H}} (L_t + V_t)$
   - **Step 4:** $V_{t+1} = L_t + V_t - H_t$
4. **until:** the stopping criteria is met
5. **Output:** the vector $\tilde{x}$ of anti-diagonals of $H_{t+1}$.

However, currently, both alternating projection method and Dykstra’s Projection Methods are heuristic and thus we need to modify the algorithms for theoretic guarantees.

### 3.3.2 The Algorithm

In this section, we present our projected Wirtinger gradient algorithm for solving (1).

Our algorithm can be seen as a variant of classical alternating projection methods. Given initial estimate $H_0 = \mathcal{H}x^0$ and $L_0 = 0$, the iterative scheme of projected gradient descent algorithm has the form,

\[
\begin{align*}
L_{t+1} &\in \mathcal{P}_{\mathbb{R}^R} (L_t - \delta_1 \nabla_L F(L_t - H_t)), \text{ for some } 0 < \delta_1 < 1, \\
H_{t+1} &\in \mathcal{P}_{\mathcal{H}} (H_t - \delta_2 \nabla_H F(H_t - L_{t+1})), \text{ for some } 0 < \delta_2 < 1,
\end{align*}
\]

The objective $F(L, H) := \frac{1}{2} \| L - H \|_F^2$ is a real-valued function with complex variables, which is not differentiable in the ordinary complex calculus sense. Nev-
Algorithm 3.3 Projected Gradient Descent Algorithm

1. **Input:** The observed signal $x_0^{(\text{true})}$ on the location set $\mathcal{M}$.
2. **Initialize:** Form a Hankel matrix $H_0 \in \mathcal{H}, L_0 = 0, 0 < \delta_1, \delta_2 < 1$
3. **Repeat:**
   For each $t = 0, 1, \cdots$ generate a sequence $\{L_t, H_t\}$ as follows:
   Step 1: $L_{t+1} \in P_R(R\mathcal{C}((L_t - \delta_1 \nabla_{L_t} F(L_t - H_t)))$
   Step 2: $H_{t+1} \in P_H(H_t - \delta_2 \nabla_{H_t} F(H_t - L_{t+1}))$
4. **until:** the stopping criteria is met
5. **Output:** the vector $\tilde{x}$ of anti-diagonals of $H_{t+1}$.

Nevertheless, $F(L, H)$ is differentiable with respect to the real and imaginary parts of its variables. Thus, our gradient descent algorithm is performed on the real and imaginary parts respectively. Denote

$$Z = \begin{bmatrix} L \\ H \end{bmatrix} = \Re + i\Im$$

where $\Re$ and $\Im$ are the real and imaginary parts of $Z$. Rewrite $F$ as $F(\Re, \Im)$. Then, in our gradient descent algorithm, $\Re$ is updated by $\frac{\partial F}{\partial \Re}$ and $\Im$ by $\frac{\partial F}{\partial \Im}$. In other words, $Z$ is updated by $\frac{\partial F}{\partial \Re} + i\frac{\partial F}{\partial \Im}$. By Wirtinger calculus, we have the relation

$$\nabla F = \frac{\partial F}{\partial \Re} + i\frac{\partial F}{\partial \Im} = 2(\frac{\partial F}{\partial Z})^*.$$  

Direct calculation gives

$$2(\frac{\partial F}{\partial Z})^* = \begin{bmatrix} 2(\frac{\partial F}{\partial L})^* \\ 2(\frac{\partial F}{\partial H})^* \end{bmatrix} = \begin{bmatrix} L - H \\ H - L \end{bmatrix}.$$  

With the Wirtinger gradient, our proposed algorithm (3.3) can become

$$\begin{cases} L_{t+1} \in P_R(R\mathcal{C}(L_t - \delta_1 (L_t - H_t))), \\ H_{t+1} \in P_H(H_t - \delta_2 (H_t - L_{t+1})), \end{cases} \quad (6)$$

where $\delta_1 > 0$ and $\delta_2 > 0$ are step sizes, and $P_R(R\mathcal{C})$ and $P_H$ are projections onto $R\mathcal{C}$ and $\mathcal{H}$ respectively. We call (6) projected Wirtinger gradient descent (PWGD).
Algorithm 3.4 Projected Wirtinger Gradient Descent Algorithm

1. **Input:** The observed signal $x_0^{(true)}$ on the location set $\mathcal{M}$.
2. **Initialize:** Form a Hankel matrix $H_0 \in \mathcal{H}$, $L_0 = 0$, $0 < \delta_1, \delta_2 < 1$
3. **Repeat:**
   - For each $t = 0, 1, \ldots$ generate a sequence $\{L_t, H_t\}_{t \in \mathbb{N}}$ as follows:
     - Step 1: $L_{t+1} \in P_{\mathbb{R}^R}(L_t - \delta_1(L_t - H_t))$
     - Step 2: $H_{t+1} \in P_{\mathbb{H}}(H_t - \delta_2(H_t - L_{t+1}))$
4. **until:** the stopping criteria is met
5. **Output:** the vector $\tilde{x}$ of anti-diagonals of $H_{t+1}$.

It remains to find out $P_{\mathbb{R}^R}$ and $P_{\mathbb{H}}$ respectively, here the projections are defined as follows: $P_{\mathbb{R}^R}(X)$ is the projection of $L_t$ onto the space of all complex-valued matrices with rank smaller than $R$, it is the best-rank-$R$-approximation to $X$ by Eckhart-Young Theorem [23],

$$P_{\mathbb{R}^R}(X) = U_R \Sigma_R V_R^H,$$

where $\Sigma_R$ is the diagonal matrix that only retains $R$ largest nonnegative singular values of $X$, and $U_R, V_R$ are the matrices whose columns are the corresponding left and right singular vectors respectively.

$P_{\mathbb{H}}(\cdot)$ denotes the projection of $H_t$ onto the subspace of Hankel matrix space that are consistent with the observed data. ... The closed form of $P_{\mathbb{H}}(\cdot)$ is given by the following lemma

**Lemma 3.1.** We have

$$P_{\mathbb{H}}(X) = H z, \quad \text{where} \quad z_j = \begin{cases} x_j^{(true)}, & \text{if } j \in \mathcal{M}, \\ \text{mean}\{X_{kl} \mid k + l = j\}, & \text{otherwise}. \end{cases}$$
Proof. $\mathcal{P}_H(X)$ is the solution of the following least square problem

$$
\mathcal{P}_H(X) = \arg\min_Z \{ \|Z - X\|_F^2 : X \in H \} = H \cdot \arg\min_z \{ \|Hz - X\|_F^2 : z_M = x^{(true)} \}
$$

$$
= H \cdot \arg\min_z \left\{ \sum_{j=0}^{2N-2} \sum_{k+l=j} (z_j - X_{kl})^2 : z_M = x^{(true)} \right\}.
$$

It is obvious that the solution of the optimization problem in the last line is given by $z_j$ in (7).

3.3.3 Acceration by a FISTA-scheme

In this subsection, we propose a scheme to accelerate the convergence of the PWGD algorithm (6). Our scheme borrows from the fast iterative shrinkage-thresholding algorithm (FISTA) [2], which has been proven to be efficient in minimizing the sum of two convex functions with one having a Lipchitz continuous gradient.

The basic idea is to use a specific linear combination of two successive iterates. Although our problem is non-convex, we still can employ the linear combination scheme in FISTA for our model.

Our PWGD with FISTA scheme, called PWGD-FISTA, is constructed as follows: Given $k_0 = 1$, we generate $\{L_t, H_t\}$ by

$$
\begin{align*}
L_{t+1} &\in \mathcal{P}_{\mathcal{F}}(L_t - \delta_1(L_t - \tilde{H}_t)), \\
H_{t+1} &\in \mathcal{P}_{\mathcal{F}}(H_t - \delta_2(\tilde{H}_t - L_{t+1})), \\
k_{t+1} &\leq \frac{\sqrt{1+4k_t^2}+1}{2}, \\
\tilde{H}_{t+1} &\leq H_{t+1} + \frac{k_t-1}{k_{t+1}}(H_{t+1} - H_t)
\end{align*}
$$

(8)
Algorithm 3.5 FISTA-PWGD

1. **Input:** The observed signal \( x_0^{(\text{true})} \) on the location set \( \mathcal{M} \).
2. **Initialize:** Form a Hankel matrix \( H_0 \in \mathcal{H}, L_0 = 0, k_0 = 1, 0 < \delta_1, \delta_2 < 1 \)
3. **Repeat:**
   For each \( t = 0, 1, \cdots \) generate a sequence \( \{L_t, H_t\}_{t \in \mathbb{N}} \) as follows:
   - **Step 1:** \( L_{t+1} \in P_{\mathbb{R}^{|\mathcal{L}|}}(L_t - \delta_1 (L_t - \hat{H}_t)) \)
   - **Step 2:** \( H_{t+1} \in P_{\mathcal{H}}(H_t - \delta_2 (H_t - L_{t+1})) \)
   - **Step 3:** \( k_{t+1} = \sqrt{1+4k_t^2+1} \)
   - **Step 4:** \( \hat{H}_{t+1} = H_{t+1} + \frac{k_t-1}{k_{t+1}} (H_{t+1} - H_t) \)
4. **until:** the stopping criteria is met
5. **Output:** the vector \( \tilde{x} \) of anti-diagonals of \( H_{t+1} \).

Since \( \mathcal{H} \) is an affine subspace, the linear combination in the last line of (12) does not change of the feasibility of \( \hat{H}_{t+1} \), i.e.,..., \( \hat{H}_{t+1} \in \mathcal{H} \). This makes the computational complexity and storage of Step 1 and Step 2 in the PWGD-FISTA algorithm is the same as that in the PWGD algorithm (6). Also, the computational effort in Step 3 and Step 4 of (12) is negligible compared with that in Step 1 and Step 2. Therefore, the PWGD-FISTA algorithm preserves the computational simplicity of the PWGD algorithm. As we will see in the numerical experiments section, the PWGD-FISTA Algorithm converges faster than the PWGD algorithm.

3.3.4 The Noisy-PWGD with Bounded Noise

In practice, measurements are often contaminated by a certain amount of noise. To make our model and algorithm more practically applicable, we replace our
measurements by $x^{(\text{noisy})}$ by the following noisy model

$$x^{(\text{noisy})} = x^{(\text{true})} + \sigma n,$$

where $x^{(\text{true})}$ is the observed clean signal, and $n$ is standard additive white Gaussian noise (AWGN) with each entry i.i.d from $\mathcal{N}(0,1)$. The signal-to-noise ratio is defined as $\text{SNR} = 10 \log_{10} \frac{M}{(2N-1)\sigma^2}$, which has been scaled with respect to the number of observations. We assume that the noise magnitude is bounded by a known amount, $\|x^{(\text{noisy})}_m - x^{(\text{true})}_m\|_F = \|\sigma n_m\|_F \leq \delta, m \in \mathcal{M}$. In order to adapt our algorithm to such noisy measurement, the variation in the estimate should be very small. Our algorithm is then modified as follows

Find a matrix $X$

s.t. $\text{rank}(X) \leq R,$

$$X_{jk} = H_{jk}^{(\text{noisy})}, \ (j,k) \in \Omega,$$

$X$ is Hankel,

While the authors in compressed sensing off-the-grid [37], proposed the noisy Atomic norm minimization algorithm as follows

$$\min_x \|x\|_A$$

s.t. $\|x_m - x_m^{(\text{noisy})}\|_F \leq \delta, m \in \mathcal{M}.$

In [13], the authors also give the noisy case of Enhanced matrix completion algorithm,

$$\min_x \|\text{Hank}(x)\|_*$$

s.t. $\|x_m - x_m^{(\text{noisy})}\|_F \leq \delta, m \in \mathcal{M}.$
3.4 Numerical Experiments

3.4.1 Implementation of the Proposed Algorithm: Choosing Parameters $\delta_1, \delta_2$

The parameters $\delta_1$ and $\delta_2$ arise from PWGD for our optimization model (8). It has been shown that the PWGD converges for any $\delta_1, \delta_2 \in (0, 1)$. In this subsection, we will demonstrate the effect of the parameters on the convergence of the proposed algorithm through our numerical experiments and shall conclude that the parameters $\delta_1$ and $\delta_2$ should be chosen very close to 1.

In our experiments, for Figure 3.2(1), the number of time measurements $2N - 1$ of the original signal $x^{(true)}$ is set to be 101 i.e., the dimension $N$ of the corresponding Hankel matrix $H^{(true)}$ is 51, the sparsity level $R$ in our model is set to be 1, and the size $M$ of location set $\Omega$ is set to be 10, and for Figure 3.2(2), the number of time measurements $2N - 1$ of the original signal $x^{(true)}$ is still set to be 101 i.e., the dimension $N$ of the corresponding Hankel matrix $H^{(true)}$ is 51, the sparsity level $R$ in our model is set to be 3, and the size $M$ of location set $\Omega$ is set to be 20. For Figure 3.3(1), the number of time measurements $2N - 1$ of the original signal $x^{(true)}$ is set to be 1001, i.e., the dimension $N$ of the corresponding Hankel matrix $H^{(true)}$ is 501, the sparsity level $R$ in our model is set to be 5, and the size $M$ of location set $\Omega$ is set to be 100, for figure 3.3(2), the number of time measurements $2N - 1$ of the original signal $x^{(true)}$ is still set to be 1001, i.e., the dimension $N$ of the corresponding Hankel matrix $H^{(true)}$ is 501, the sparsity level $R$ in our model is set to be 8, and the size $M$ of location set $\Omega$ is set to be 200. PWGD is tested with five different values of $\delta_1 = \delta_2 \in \{0.3, 0.5, 0.7, 0.9, 0.9999\}$. The signal recovery efficiency, normal mean
square error NMSE between the initial matrix $H^{\text{true}}$ and the final matrix $H^{\text{final}}$, here we set $\frac{\|H^{\text{final}} - H^{\text{true}}\|_2}{\|H^{\text{true}}\|_2} \leq 0.0001$. The signal recovery efficiency against the number of iterations of the algorithm with different values of $\delta_1, \delta_2$ are then displayed in following table 3.1. We can see clearly that with the same number of iterations, the PWGD with larger values of parameters $\delta_1, \delta_2$ have better signal recovery efficiency (i.e., lower NMSE). Therefore, we shall always choose $\delta_1 = \delta_2 = 0.9999$ in the rest of my experiments.

<table>
<thead>
<tr>
<th>signal</th>
<th>$\delta_1, \delta_2 = 0.3$</th>
<th>$\delta_1, \delta_2 = 0.5$</th>
<th>$\delta_1, \delta_2 = 0.7$</th>
<th>$\delta_1, \delta_2 = 0.9$</th>
<th>$\delta_1, \delta_2 = 0.9999$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N = 51, R = 1, M = 10$</td>
<td>0.55s</td>
<td>0.49s</td>
<td>0.43s</td>
<td>0.36s</td>
<td>0.34s</td>
</tr>
<tr>
<td>$N = 51, R = 3, M = 20$</td>
<td>0.72s</td>
<td>0.64s</td>
<td>0.58s</td>
<td>0.50s</td>
<td>0.46s</td>
</tr>
<tr>
<td>$N = 501, R = 5, M = 100$</td>
<td>20.17s</td>
<td>17.85s</td>
<td>15.84s</td>
<td>13.89s</td>
<td>12.69s</td>
</tr>
<tr>
<td>$N = 501, R = 8, M = 200$</td>
<td>10.97s</td>
<td>9.52s</td>
<td>8.29s</td>
<td>7.06s</td>
<td>6.67s</td>
</tr>
</tbody>
</table>

Table 3.1. Elapsed time(second) for the signals with different parameters.
Figure 3.2. The effect of the parameter $\delta_1$ and $\delta_2$ on the convergence of the proposed algorithm PWGD for the signals. From top to bottom: (1) The signal with $N = 51, R = 1, M = 10$; (2) The signal with $N = 51, R = 3, M = 20$. 
Figure 3.3. The effect of the parameter $\delta_1$ and $\delta_2$ on the convergence of the proposed algorithm PWGD for the signals. From top to bottom: (1) The signal with $N = 501, R = 5, M = 100$; (2) The signal with $N = 501, R = 8, M = 200$. 
3.4.2 Acceleration of the Proposed Algorithm by a FISTA-like Scheme

All the figures 3.4(1), 3.4(2), 3.5(1) and 3.5(2) and the table ?? depicts the convergence curve of PWGD and PWGD-FISTA. We see clearly that the PWGD-FISTA algorithm converges faster than the PWGD algorithm. Roughly, the PWGD-FISTA needs only 2/3 number of iterations that PWGD requires to get solutions of the same accuracy.

Recall that PWGD with FISTA scheme, called PWGD-FISTA, is constructed as follows: Given $k_0 = 1$, we generate $\{L_t, H_t\}$ by

\[
\begin{cases}
    L_{t+1} \in \mathcal{P}_{\mathbb{R}^R}(L_t - \delta_1(L_t - \tilde{H}_t)), \\
    H_{t+1} \in \mathcal{P}_{\mathbb{R}^H}(H_t - \delta_2(\tilde{H}_t - L_{t+1})), \\
    k_{t+1} = \sqrt{1+4k^2_t+1}, \\
    \tilde{H}_{t+1} = H_{t+1} + \frac{k_t-1}{k_{t+1}}(H_{t+1} - H_t)
\end{cases}
\]

(12)
Figure 3.4. Convergence rate comparison between PWGD and PWGD-FISTA. From top to bottom: (1) The signal with $N = 51, R = 1, M = 10$; (2) The signal with $N = 51, R = 3, M = 20$. 
Figure 3.5. Convergence rate comparison between PWGD and PWGD-FISTA. From top to bottom: (1) The signal with $N = 501, R = 8, M = 200$; (2) The signal with $N = 5001, R = 20, M = 1000$. 
3.4.3 Phase Transition in the noiseless setting

To evaluate the practical ability of PWGD, we conducted a series of numerical experiments to examine the phase transition for exact recovery. We compare the performance and complexity of our algorithms, PWGD and PWGD-FISTA, with the Atomic Norm Minimization (ANM) [37] and the Enhanced Matrix Completion (EMaC) [13]. For each \((R, M)\) pair, 100 Monte Carlo trials were conducted. We generated a spectrally sparse complex signal \(x\) by randomly generating \(R\) frequencies in \([0, 1)\). PWGD and PWGD-FISTA were conducted by setting parameters \(\delta_1 = \delta_2 = 0.9999\). We use the convex programming modeling software CVX to solve the programs for EMaC and ANM. Each trail is considered successful if the normalized mean square error (NMSE) satisfies 

\[
\frac{\|x^{(\text{final})} - x^{(\text{true})}\|_2}{\|x^{(\text{true})}\|_2} \leq 0.005,
\]

where \(x^{(\text{final})}\) denotes the estimate return by my algorithm. The empirical success rate is calculated by averaging over 100 Monte Carlo trials.

The following figures illustrate the results of the Monte Carlo experiments when the dimension of the signal is from 20 to 127, the horizontal axis corresponds to the number \(M\) of the samples (i.e., the size of the location set \(\mathcal{M}\) ) revealed to the algorithm, while the vertical axis corresponds to sparsity level \(R\). The empirical success rate is reflected by the color of each cell. Compare to Figure 3.8, 3.9 and 3.10, Figure 3.6 and 3.7 demonstrate that the projected Wirtinger algorithms have better phase transition boundary than other algorithms, especially PWGD-FISTA. These phase transition diagrams tell the highly practical ability of projected Wirtinger algorithms, especially PWGD-FISTA, i.e., PWGD-FISTA and PWGD requires the
minimal number of samples to have perfect recovery of the signals.

Figure 3.6. Phase transition of PWGD-FISTA.

Figure 3.7. Phase transition of PWGD.
Figure 3.8. Phase transition of ANM with separation condition

\[ \Delta f > \frac{1.5}{\|\lambda\|} \]

Figure 3.9. Phase transition of ANM without separation condition.
3.4.4 1D signal of large scale

Next we demonstrate that our proposed algorithm is able to recover signals of large scale, and compare it with the Atomic Norm Minimization (ANM) [37] and the Enhanced Matrix Completion (EMaC) [13]. As we have argued, different from the existing convex optimization based methods such as ANM and EMaC, our proposed algorithm is able to work with large-scale spectrally sparse signals. In table 4.1, the elapsed time for signals of different scale are listed. For our algorithm, we use the same setting as in the previous section, For ANM and EMaC algorithms, we used the convex software package CVX to solve them. From the table, we can see that PWGD-FISTA and PWGD can greatly speed up the signal recovery for moderate scale and also work well for the signal of large scale.
Figure 3.11. Perfect recovery of a large scale signal with $N = 5001$, $R = 40$, $M = 2000$. 

\begin{center}
\includegraphics[width=\textwidth]{figure311.png}
\end{center}
Table 3.2. The table clearly show that the speed of our projected Wirtinger methods outperform those of ANM and EMaC.

3.4.5 The Noisy-PWGD algorithm with Bounded Noise

The following figures illustrate the NMSE against SNR under different algorithms. It can be clearly seen that our algorithms, including PWGD and PWGD-FISTA, have better performance than other existing algorithms.
Figure 3.12. NMSE vs. SNR when the signals where frequencies are generated randomly for different algorithms. From top to bottom: (1) The signal with $N = 81, R = 2, M = 30$; (2) The signal with $N = 81, R = 2, M = 50$. 
CHAPTER 4
PWGD OF HANKEL TOEPLITZ MOSAIC MATRIX COMPLETION

4.1 Introduction

In the early chapter, we are able to reconstruct a spectrally sparse signal through a Hankel matrix completion. In this chapter, Myung Cho, Weiyu Xu and we applied PWGD algorithm through a different formulation of structured matrix recovery. Considering the close relation of Hankel and Toeplitz matrices, we proposed to study a non-convex optimization approach to a different structured matrix completion. In particular, we consider the following nonconvex optimization formulations: low-rank Hankel and Toeplitz Mosaic (HTM) matrix completion. Then we still use general projected Wirtinger Gradient descent methods for directly solving this non-convex optimization formulation, without resorting to convex relaxations. Numerical results show that we can improve the performance or the speed of recovery using projected Wirtinger gradient descent, compared with atomic norm minimization [37] and nuclear norm minimization for Hankel matrix completion [13].

The remainder of this chapter is organized as follows. In Section 2, we introduce low-rank Hankel and Toeplitz Mosaic (HTM) matrix completion. Thereafter, in Section 3, we propose projected Wirtinger gradient descent to directly solve the HTM and Hankel completion problems. Finally, in Section 4, we demonstrate the efficiency of our algorithms in terms of signal recovery performance as well as the computational complexity.
4.2 Problem Formulation: Hankel and Toeplitz Mosaic Matrix Completion

The signal model is the same as discussed in the first chapters. In this section, we introduce a new optimization formulation, called Hankel and Toeplitz Mosaic (HTM) matrix completion, to recover spectrally sparse signals. Our HTM matrix completion is formulated as follows:

\[
\min_{\mathbf{x}, \mathbf{z}} \quad \text{Rank}(\mathbf{U}_3)
\]
\[
\text{s.t} \quad \mathbf{U}_3 = \begin{bmatrix} \text{Toep}(\mathbf{z}) & \text{Hank}(\mathbf{x}) \\ \text{Hank}(\mathbf{x})^H & \text{Toep}(\mathbf{z})^H \end{bmatrix} \succeq 0,
\]
\[
\mathbf{x}_m = \mathbf{x}_m^{(\text{true})}, m \in \mathcal{M}.
\]
4.3 Our Algorithm

To introduce the method for HTM matrix completion, we need the following definitions.

The set of all complex-valued matrices with rank no greater than $R$ is defined as the same as in the early chapter,

\[ \mathcal{R}_C^R = \{ L \in \mathbb{C}^{2N \times 2N} | \text{rank}(L) \leq R \}. \tag{1} \]

We further denote the set of all complex-valued Toeplitz Hankel mosaic matrices that are consistent with the observed data as

\[ \mathcal{H}_{HTM} = \left\{ \begin{bmatrix} \text{Toep}(z) & \text{Hank}(x) \\ \text{Hank}(x)^H & \text{Toep}(z)^H \end{bmatrix} : z \in \mathbb{C}^N, x \in \mathbb{C}^{2N-1}, x_M = x_M^{(true)} \right\}. \tag{2} \]

We then seek a matrix in $\mathcal{H}_{HTM}$ with rank no greater than $R$ by considering the following problem:

\[ \min_{L \in \mathcal{R}_C^R, H \in \mathcal{H}_{HTM}} F(L, H) := \frac{1}{2} \| L - H \|_F^2 \tag{3} \]

To solve (3), we still use the gradient descent algorithm. We start with the initial value $H_0$ and $L_0$. By considering Wirtinger calculus and applying gradient descent on the function with complex variables, we derive the update rule of our projected Wirtinger algorithm as follows:

\[
\begin{cases}
L_{t+1} \in \mathcal{P}_{\mathcal{R}_C^R}(L_t - \delta_1 (L_t - H_t)), \\
H_{t+1} \in \mathcal{P}_{\mathcal{H}_{HTM}}(H_t - \delta_2 (H_t - L_{t+1})),
\end{cases}
\tag{4}
\]

where $t$ is the iteration number, $\delta_1 > 0$ and $\delta_2 > 0$ are step size, $\mathcal{P}_{\mathcal{R}_C^R}$ and $\mathcal{P}_{\mathcal{H}_{HTM}}$ are the projections onto $\mathcal{R}_C^R$ and $\mathcal{H}_{HTM}$ respectively.
Algorithm 4.1 Projected Wirtinger Gradient Descent Algorithm for HTM

1. **Input:** The observed signal \( x_0^{(\text{true})} \) on the location set \( \mathcal{M} \).
2. **Initialize:** Form a Hankel matrix \( H_0 \in \mathcal{H}_{HTM} \), \( L_0 = 0 \), \( 0 < \delta_1, \delta_2 < 1 \)
3. **Repeat:**
   For each \( t = 0, 1, \cdots \) generate a sequence \( \{L_t, H_t\}_{t \in \mathbb{N}} \) as follows:
   Step 1: \( L_{t+1} \in \mathcal{P}_{\mathbb{C}^R}(L_t - \delta_1(L_t - H_t)) \)
   Step 2: \( H_{t+1} \in \mathcal{P}_{\mathcal{H}_{HTM}}(H_t - \delta_2(H_t - L_{t+1})) \)
4. **until:** the stopping criteria is met
5. **Output:** the vector \( \tilde{x} \) of anti-diagonals of \( H_{t+1} \).

\( \mathcal{P}_{\mathbb{C}^R}(X) \) is the projection of \( L_t \) onto the space of all complex-valued matrices with rank smaller than \( R \), it is the best-rank-R-approximation to \( X \) by Eckhart-Young Theorem [23],

\[
\mathcal{P}_{\mathbb{C}^R}(X) = U_R \Sigma_R V_R^H,
\]

where \( \Sigma_R \) is the diagonal matrix that only retains \( R \) largest nonnegative singular values of \( X \), and \( U_R, V_R \) are the matrices whose columns are the corresponding left and right singular vectors respectively.

The projection \( \mathcal{P}_{\mathcal{H}_{HTM}}(X) \) is carried out for Hankel and Toeplitz matrices separately due to its mosaic structure. More precisely, for any \( X = \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^H & X_{22} \end{bmatrix} \in \mathbb{C}^{2N \times 2N} \), we have

\[
\mathcal{P}_{\mathcal{H}_{HTM}}(X) = \begin{bmatrix}
\frac{1}{2} \mathcal{P}_T(X_{11} + X_{22}^T) & \mathcal{P}_HX_{12} \\
\mathcal{P}_HX_{12}^H & \frac{1}{2} \mathcal{P}_T(X_{11} + X_{22}^T)^T
\end{bmatrix}
\]

Here \( \mathcal{P}_H \) is the projection onto the set of Hankel matrices whose anti-diagonals con-
sistent with the observed data.

\[
P_H(X) = \text{Hank}(z), \quad \text{where} \quad z_j = \begin{cases} x_j^{(\text{true})}, & \text{if } j \in \mathcal{M}, \\ \text{mean}\{X_{kl} \mid k + l = j\}, & \text{otherwise.} \end{cases} \tag{5}
\]

The operator \(P_T\) is the projection onto the set of Toeplitz matrices, and is given by

\[
P_T(X) = \text{Toep}(z), \quad \text{where} \quad z_j = \text{mean}\{X_{kl} \mid k - l = j, k \geq l\}.
\]

### 4.4 Numerical Experiments

We compare the performance and complexity of our projected Wirtinger gradient descent methods for both HTM and Hankel matrix completion with the standard Atomic Norm Minimization (ANM) [37], and the Enhanced Matrix Completion (EMaC) [13]. We use CVX [24] to solve convex programs for ANM and EMaC. In all experiments, the phases and frequencies are sampled uniformly at random in \([0, 2\pi)\) and \([0, 1]\) respectively. The amplitudes \(d_k, k = 1, \ldots, R\), are chosen as \(d_k = 1 + 10^{0.5m_j}\), where \(m_j\) is randomly drawn in the uniform distribution on \([0, 1]\).

We carried out numerical experiments to compare the signal recovery performance of our projected Wirtinger method with ANM and EMaC. We abbreviate HTM and Hankel matrix completion using projected Wirtinger to PWGD-HTM and PWGD respectively. We varied the number of measurements \(M\) from 20 to 127, and the sparsity \(R\) from 1 to 40. We obtained the probability of successful signal recovery over 100 trials for each parameter setup. We consider a recovery successful if

\[
\|x^{(\text{true})} - \hat{x}\|/\|x^{(\text{true})}\| \leq \epsilon_{\text{succ}}, \quad \text{where} \quad \epsilon_{\text{succ}} = 10^{-2}, \quad \hat{x} \quad \text{is the estimated signal, and} \quad x^{(\text{true})} \quad \text{is the true signal.} \quad 4.1 \quad \text{and} \quad 4.2 \quad \text{demonstrate that the projected Wirtinger}
\]
algorithms improve the phase transition boundary over other algorithms (4.3 and 4.4). Even though PWGD-HTM has the best phase transition boundary curve, in certain regions (upper orange color region) of its phase transition, PWGD-HTM has smaller success rate than other algorithms. It would be interesting to understand this phenomenon more deeply. We leave this for future work.

Figure 4.1. Phase transition of PWGD-HTM.
Figure 4.2. Phase transition of PWGD.

Figure 4.3. Phase transition of ANM.
We provide 4.5 to more clearly show the advantage of our algorithms in signal recovery, where we use the success criterion $\epsilon_{\text{succ}} = 5 \times 10^{-3}$. In particular, when the sparsity $R$ is 40, PWGD-HTM has 50% success rate with around 87 measurements out of 127 respectively, while the success rate of other algorithms with the same number of measurements is 0. The 50% success rate of other algorithms is achieved around 110 for EMaC, and 120 for ANM.
Figure 4.5. The probability of signal recovery when \((2N - 1, R) = (127, 40)\).

We assess the computational complexity of our algorithms in terms of the average execution time by averaging over 10 trials. We provide results in table 4.1 when the signal dimension \((2N - 1)\) varies from 101 to 1401, \(M = \lfloor (2n - 1)/2 \rfloor\), and \(R = \lfloor 0.1(2N - 1) \rfloor\). We stopped our projected Wirtinger algorithms when \(\|H_{t+1} - H_t\|_F/\|H_t\|_F \leq 10^{-6}\) or the maximum number of iterations (MaxIt) is exhausted. We set MaxIt to 1000. The success rate was similar to that shown in previous experiments.
<table>
<thead>
<tr>
<th>signal</th>
<th>PWGD-HTM</th>
<th>PWGD</th>
<th>ANM</th>
<th>EMaC</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N = 101, M = 50, R = 10$</td>
<td>3.7s</td>
<td>0.1s</td>
<td>5.7s</td>
<td>27.1s</td>
</tr>
<tr>
<td>$N = 201, M = 100, R = 20$</td>
<td>7.1s</td>
<td>0.2s</td>
<td>51.6s</td>
<td>787.7s</td>
</tr>
<tr>
<td>$N = 401, M = 200, R = 40$</td>
<td>309.7s</td>
<td>0.9s</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>$N = 601, M = 300, R = 60$</td>
<td>733.4s</td>
<td>1.4s</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>$N = 1001, M = 500, R = 100$</td>
<td>3612.4s</td>
<td>6.6s</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>$N = 1401, M = 700, R = 140$</td>
<td>8999.2s</td>
<td>10.1s</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

*Not Available (Out of Memory)*

Table 4.1. The table clearly show that the speed of our projected Wirtinger methods outperform those of ANM and EMaC.
CHAPTER 5
CONVERGENCE THEORY

5.1 Introduction

In this chapter, we will prove the convergence of the proposed PWGD algorithm (6). Our proof is achieved by applying the convergence result in [1].

First let’s consider a general non-convex optimization problem

$$\min_{x,y} \psi(x, y) := \phi(x, y) + \theta(x) + \omega(y),$$

(1)

where the functions \(\theta : \mathbb{R}^n \mapsto \mathbb{R} \cup \{+\infty\}\) and \(\omega : \mathbb{R}^m \mapsto \mathbb{R} \cup \{+\infty\}\) are proper lower semicontinuous functions. \(\phi : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}\) is a \(C^1\) function. And \(\nabla \psi\) is Liphtshitz continuous on bounded subsets of \(\mathbb{R}^n \times \mathbb{R}^m\). In [1], a proximal alternating minimization algorithm is proposed for solving (1)

$$\begin{align*}
    x_{k+1} & \in \arg\min_{x \in \mathbb{R}^n} \psi(x, y_k) + \frac{1}{2\lambda_k} \|x - x_k\|_2^2, \\
y_{k+1} & \in \arg\min_{y \in \mathbb{R}^m} \psi(x_{k+1}, y) + \frac{1}{2\mu_k} \|y - y_k\|_2^2.
\end{align*}$$

(2)

On the premise that the function \(\psi\) satisfies the Kurdyka-Lojasiewicz (KL) property, [1] proved the convergence of (2).

The property is given as follows.

**Definition 5.1.** (*Kurdyka-Lojasiewicz property*) The function \(f\) is said to have the Kurdyka-Lojasiewicz property at \(\bar{x} \in \text{dom}\, f\) if there exist \(\eta \in (0, +\infty]\), a neighborhood \(U\) of \(\bar{x}\) and a continuous concave function \(\phi : [0, \eta) \to \mathbb{R}_+\) such that:

1. \(\phi(0) = 0\),
2. $\phi$ is $C^1$ on $(0,1)$,

3. for all $s \in (0,\eta)$, $\phi'(s) > 0$,

4. and for all $x$ in $U \cap [f(\bar{x}) < f < f(\bar{x}) + \eta]$, the Kurdyka-Lojasiewicz inequality holds

$$\phi'(f(x) - f(\bar{x})) \text{dist}(0,\partial f(x)) \geq 1.$$ 

Generally, the KL condition is not easy to check. A sufficient condition to guarantee the KL property is the semi-algebraic property.

**Definition 5.2.** (Semi-algebraic sets) A subset $S \subset \mathbb{R}^d$ is a real semi-algebraic set if there exists a finite number of real polynomial function $g_{ij}, h_{ij} : \mathbb{R}^d \to \mathbb{R}$ such that

$$S = \bigcup_{j=1}^{p} \bigcap_{i=1}^{q} \{u \in \mathbb{R}^d \mid g_{ij}(u) = 0, h_{ij}(u) < 0\}.$$ 

A proper and lower semi-continuous function is called semi-algebraic if its graph is a semi-algebraic set. Such a function satisfies the KL property with $\phi(s) = cs^{1-\alpha}$, for some $\alpha \in [0,1) \cap \mathbb{Q}$ and some $c > 0$. And there are some properties of semi-algebraic functions, the properties will be used later and they are listed below.

1. finite sums and products of semi-algebraic functions are semi-algebraic;

2. scalar products are semi-algebraic;

3. indicator functions of semi-algebraic sets are semi-algebraic;

4. composition of semi-algebraic functions or mappings are semi-algebraic.

And matrix theory provides a lot of semi-algebraic objects, such as positive semi-definite matrices, Stiefel manifolds (spheres, orthogonal group; constant rank matrices) and etc.
When specializing algorithm (2) to indicator functions, we obtain an alternating projection algorithm, which can be seen as a variant of the classical Von Neumann algorithm. Being given two closed sets \( C, D \) of \( \mathbb{R}^n \), the algorithm has the form

\[
\begin{align*}
  x_{k+1} &\in P_C \left( x_k - \frac{1}{1+\lambda_k} (x_k - y_k) \right), \\
y_{k+1} &\in P_D \left( y_k - \frac{1}{1+\mu_k} (y_k - x_{k+1}) \right),
\end{align*}
\]

(3)

here \( P_C, P_D \) are the projections onto \( C \) and \( D \).

The convergence of the sequences \((x_k, y_k)\) is a corollary of Corollary 12 of [1] and Theorem 3 and Example 2 of [4]. Moreover, our PWGD algorithm can be seen as a special case of (3), and the theorem and the proof will be given in the later section.

The remainder of this chapter is organized as follows. In section 2, first we will show that the alternating projection algorithm (3) is a special case of the proximal alternating minimization algorithm (2). Then we will apply Theorem 5.1 to the PWGD algorithm (6) to get its convergence.

### 5.2 Main results

From the general non-convex function (1), choose \( \theta = \delta_C \) and \( \omega = \delta_D \), which are indicator functions for the sets \( C \in \mathbb{R}^n \) and \( D \in \mathbb{R}^m \) respectively. Recall the indicator function \( \delta_C \) of a set \( C \) is defined as

\[
\delta_C(x) = \begin{cases} 
  0, & \text{if } x \in C, \\
  +\infty, & \text{if } x \notin C
\end{cases}
\]

Let

\[
\phi(x, y) = \frac{1}{2} \| x - y \|_2^2.
\]

Then we have the following lemma:

**Lemma 5.1.** If the objective function has the special form

\[
\min_{x, y} \psi(x, y) := \frac{1}{2} \| x - y \|_2^2 + \delta_C(x) + \delta_D(y),
\]

(4)
Then the proximal alternating minimization algorithm (2)

\[
\begin{align*}
    x_{k+1} &\in \arg\min_{x \in \mathbb{R}^n} \psi(x, y_k) + \frac{1}{2\lambda_k} \|x - x_k\|_2^2, \\
y_{k+1} &\in \arg\min_{y \in \mathbb{R}^m} \psi(x_{k+1}, y) + \frac{1}{2\mu_k} \|y - y_k\|_2^2,
\end{align*}
\]

becomes an alternating projection algorithm (3)

\[
\begin{align*}
x_{k+1} &\in \mathcal{P}_C \left( x_k - \frac{1}{1 + \lambda_k} (x_k - y_k) \right), \\
y_{k+1} &\in \mathcal{P}_D \left( y_k - \frac{1}{1 + \mu_k} (y_k - x_{k+1}) \right).
\end{align*}
\]

Proof.

\[
\begin{align*}
x_{k+1} &\in \arg\min_{x \in \mathbb{R}^n} \frac{1}{2} \|x - y_k\|_2^2 + \delta_C(x) + \frac{1}{2\lambda_k} \|x - x_k\|_2^2, \\
y_{k+1} &\in \arg\min_{y \in \mathbb{R}^m} \frac{1}{2} \|x_{k+1} - y\|_2^2 + \delta_D(y) + \frac{1}{2\mu_k} \|y - y_k\|_2^2.
\end{align*}
\]

Note that \(\delta_C\) and \(\delta_D\) are indicator functions, then we have

\[
\begin{align*}
x_{k+1} &\in \arg\min_{x \in \mathbb{R}^n} \frac{1}{2} \|x - y_k\|_2^2 + \frac{1}{\lambda_k} \|x - x_k\|_2^2, \\
y_{k+1} &\in \arg\min_{y \in \mathbb{R}^m} \frac{1}{2} \|x_{k+1} - y\|_2^2 + \delta_D(y) + \frac{1}{2\mu_k} \|y - y_k\|_2^2.
\end{align*}
\]

then combine some terms, we have

\[
\begin{align*}
x_{k+1} &\in \arg\min_{x \in \mathbb{R}^n} \frac{1}{2} (\|x - y_k\|_2^2 + \frac{1}{\lambda_k} \|x - x_k\|_2^2) + \delta_C(x), \\
y_{k+1} &\in \arg\min_{y \in \mathbb{R}^m} \frac{1}{2} (\|x_{k+1} - y\|_2^2 + \frac{1}{\delta_k} \|y - y_k\|_2^2) + \delta_D(y).
\end{align*}
\]
because $1 > \frac{\lambda_k}{1 + \lambda_k}$ and $\frac{1}{\lambda_k} > \frac{1}{1 + \lambda_k}$, and same relation for $\delta_k$, then we have

\[
\begin{align*}
\mathbf{x}_{k+1} &\in \arg\min_{\mathbf{x} \in \mathbb{R}_n} \frac{1}{2} \left( \frac{\lambda_k}{1 + \lambda_k} \| \mathbf{x} - \mathbf{y}_k \|_2^2 + \frac{1}{1 + \lambda_k} \| \mathbf{x} - \mathbf{x}_k \|_2^2 \right) + \delta_C(\mathbf{x}), \\
\mathbf{y}_{k+1} &\in \arg\min_{\mathbf{y} \in \mathbb{R}_m} \frac{1}{2} \left( \frac{\delta_k}{1 + \delta_k} \| \mathbf{x}_{k+1} - \mathbf{y} \|_2^2 + \frac{1}{1 + \delta_k} \| \mathbf{y} - \mathbf{y}_k \|_2^2 \right) + \delta_D(\mathbf{y}).
\end{align*}
\]

recall that the projection on the set $C$ is defined as $P_C(\mathbf{x}) = \arg\min_{\mathbf{z} \in C} \| \mathbf{x} - \mathbf{z} \|_2$, $\mathbf{z} \in C$ and also by the triangle inequality, we have

\[
\begin{align*}
\mathbf{x}_{k+1} &\in P_C \left( \frac{1}{1 + \lambda_k} \mathbf{x}_k + \frac{\lambda_k}{1 + \lambda_k} \mathbf{y}_k \right), \\
\mathbf{y}_{k+1} &\in P_D \left( \frac{1}{1 + \delta_k} \mathbf{x}_k + \frac{\delta_k}{1 + \delta_k} \mathbf{y}_k \right),
\end{align*}
\]

which is equivalent to (3).

The results in [1] imply the following convergence theorem of (3), which is a corollary of Corollary 12 of [1] and Theorem 3 and Example 2 of [4].

**Theorem 5.1.** Assume that the sets $C \subset \mathbb{R}^n$ and $D \subset \mathbb{R}^m$ are semi-algebraic. Let $(\mathbf{x}_k, \mathbf{y}_k)$ be generated by (3) with $0 < \delta_k, \lambda_k < 1$ for all $k$.

(a) Either $\| (\mathbf{x}_k, \mathbf{y}_k) \|_2 \to \infty$ as $k \to \infty$, or $(\mathbf{x}_k, \mathbf{y}_k)$ converges to a critical point of $\psi$.

(b) If we further assume $(\mathbf{x}_0, \mathbf{y}_0)$ is feasible and sufficiently close to a global minimizer of $\psi$, then $(\mathbf{x}_0, \mathbf{y}_0)$ converges to a global minimizer of $\psi$.

Next we apply Theorem 5.1 to the PWGD algorithm (6) to get its convergence. The PWGD algorithm (6) is in the same form as (3). However, our PWGD algorithm is performed in complex-valued matrix spaces, while the setting of Theorem 5.1 is in real. Nevertheless, as we discussed in last section, we can identify any complex-valued
matrix to a real one by concatenating its real and imaginary parts. Actually, as aforementioned, our Writinger gradient descent is exactly obtained in this way by considering the gradient with respect to the real and imaginary parts. Since the objective function $F(L, H)$ in (4) does not change after this identification, we only need to check the sets $\mathcal{R}_C^R$ in (1) and $\mathcal{H}$ in (3) are semi-algebraic when viewed as sets of real and imaginary parts. This is done by the following two lemmas.

**Lemma 5.2.** The set $\mathcal{I}_R$ defined as follows is a semi-algebraic set

$$\mathcal{I}_R = \left\{ [X, Y] \mid (X + iY) \in \mathcal{R}_C^R \right\}.$$  

*Proof.* Denote

$$\mathcal{P}_R = \left\{ [X, Y] \mid X, Y \in \mathbb{R}^{N \times N}, \text{rank}(X + iY) = R \right\}$$

and

$$\mathcal{Q}_R = \left\{ [X, Y] \mid X, Y \in \mathbb{R}^{N \times N}, \text{rank} \begin{pmatrix} X & -Y \\ Y & X \end{pmatrix} = 2R \right\}.$$  

We first prove $\mathcal{P}_R = \mathcal{Q}_R$ by showing $\mathcal{P}_R \subset \mathcal{Q}_R$ and $\mathcal{Q}_R \subset \mathcal{P}_R$ respectively. Let $[X, Y] \in \mathcal{P}_R$, and a singular value decomposition (SVD) of $X + iY$ is $(X + iY) = (U_{\text{Re}} + iU_{\text{Im}})\Sigma(V_{\text{Re}} + iV_{\text{Im}})^*$, where $U_{\text{Re}}, U_{\text{Im}}, V_{\text{Re}}, V_{\text{Im}} \in \mathbb{R}^{N \times R}$ and $\Sigma \in \mathbb{R}^{R \times R}$.

Then, by direct calculation, we see that an SVD of

$$\begin{pmatrix} X & -Y \\ Y & X \end{pmatrix}$$

is given by

$$\begin{pmatrix} X & -Y \\ Y & X \end{pmatrix} = \begin{pmatrix} U_{\text{Re}} & -U_{\text{Im}} \\ U_{\text{Im}} & U_{\text{Re}} \end{pmatrix} \begin{pmatrix} \Sigma & 0 \\ 0 & \Sigma \end{pmatrix} \begin{pmatrix} V_{\text{Re}} & -V_{\text{Im}} \\ V_{\text{Im}} & V_{\text{Re}} \end{pmatrix}^*.$$  

(5)
Therefore, \( \text{rank} \begin{bmatrix} X & -Y \\ Y & X \end{bmatrix} = 2R \), which implies \([X, Y] \in \mathcal{D}_R\) and further \(\mathcal{P}_R \subset \mathcal{D}_R\). Conversely, let \([X, Y] \in \mathcal{D}_R\). If \(\begin{pmatrix} \sigma, \\
 & \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \\ v & \begin{bmatrix} v_2 \\ v_1 \end{bmatrix} \end{pmatrix}\) is a singular triplet of \(\begin{bmatrix} X & -Y \\ Y & X \end{bmatrix}\), then \(\sigma, \begin{bmatrix} -u_2 \\ u_1 \end{bmatrix}, \begin{bmatrix} -v_2 \\ v_1 \end{bmatrix}\) is too by direct calculation. Therefore, the multiplicity of each singular value is even, and SVD’s of \(\begin{bmatrix} X & -Y \\ Y & X \end{bmatrix}\) must be in the form of (5). Consequently, \((X + iY) = (U_{\text{Re}} + iU_{\text{Im}})\Sigma(V_{\text{Re}} + iV_{\text{Im}})^*\) is an SVD of \((X + iY)\), which implies \(\text{rank}(X + iY) = R\). Therefore, \([X, Y] \in \mathcal{P}_R\). Thus, \(\mathcal{D}_R \subset \mathcal{P}_R\).

Since \(\mathcal{D}_R\) is the intersection of the set of all rank-2 real-valued matrices and the linear subspace of matrices in the form of \(\begin{bmatrix} X & -Y \\ Y & X \end{bmatrix}\), it is deducted from [1][Example 2] that \(\mathcal{D}_R\) is a semi-algebraic set. This together with \(\mathcal{P}_R = \mathcal{D}_R\) implies \(\mathcal{P}_R\) is a semi-algebraic set too.

Finally, it is obvious that \(\mathcal{I}_R = \bigcup_{r=0}^{R} \mathcal{P}_R\). As a matter of fact, \(\mathcal{I}_R\) is a semi-algebraic set.

**Lemma 5.3.** The set \(\mathcal{K}\) defined as follows is a semi-algebraic set

\[
\mathcal{K} = \{ [X, Y] \mid (X + iY) \in \mathcal{K} \}.
\]

**Proof.** Since \(\mathcal{H}\) is a linear operator,

\[
\mathcal{H}x = \mathcal{H}\Re(x) + i\mathcal{H}\Im(x).
\]
Further, for any \( x \) satisfying \( x_M = x_M^{\text{true}} \), we have \( \Re(x_M) = \Re(x_M^{\text{true}}) \) and \( \Im(x_M) = \Im(x_M^{\text{true}}) \). Therefore,

\[
\mathcal{K} = \Re(\mathcal{H}) + i\Im(\mathcal{H}) = \mathcal{K}_1 + i\mathcal{K}_2,
\]

where

\[
\mathcal{K}_1 = \{ \mathcal{H}r \mid r \in \mathbb{R}^{2N-2}, r_M = \Re(x_M^{\text{true}}) \}, \quad \mathcal{K}_2 = \{ \mathcal{H}i \mid i \in \mathbb{R}^{2N-2}, i_M = \Im(x_M^{\text{true}}) \}.
\]

This shows \( \mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2 \). Since both \( \mathcal{K}_1 \) and \( \mathcal{K}_2 \) are affine spaces, their product \( \mathcal{K} \) is also, which implies \( \mathcal{K} \) is semi-algebraic.

Combining Theorem 5.1 and Lemmas 5.2 and 5.3 leads to the following convergence results of the proposed algorithm (6).

**Theorem 5.2.** Let \( (L_t, H_t) \) be generated by (6) with \( 0 < \delta_1, \delta_2 < 1 \).

(a) Either \( \| (L_t, H_t) \|_F \to \infty \) as \( t \to \infty \), or \( (L_t, H_t) \) converges.

(b) If we further assume \( (L_0, H_0) \) is feasible and sufficiently close to a global minimizer of \( \left\{ \frac{1}{2} \| L - H \|_F^2 \mid L \in \mathbb{R}^R, H \in \mathcal{H} \right\} \), then \( (L_0, H_0) \) converges to a global minimizer of \( \min_{L \in \mathbb{R}^R, H \in \mathcal{H}} \frac{1}{2} \| L - H \|_F^2 \).

We would like to remark that the unboundedness in (5.2)(a) is not a problem and can be overcome by introduce a bound constraint in the set \( \mathcal{H} \). For example, we can define \( \tilde{\mathcal{H}} = \mathcal{H} \cap \{ X \mid \| X \|_\infty \leq B \} \) with \( B \) a very large number, and (6) is slightly modified by replacing \( \mathcal{H} \) by \( \tilde{\mathcal{H}} \). Then all the conditions in Theorem 5.1 can still be verified. Since \( \| (L_k, H_k) \|_F \not\to \infty \), we must have \( (L_k, H_k) \) converges.

The following theorem is about the rate of convergence of our PWGD algorithm, which applies Theorem 11 in [1].
Theorem 5.3. (rate of convergence) Let \((L_t, H_t)\) be generated by (6) with \(0 < \delta_1, \delta_2 < 1\). Assume further that \((L_k, H_k)\) converges to \((L_\infty, H_\infty)\) and objective function \(\psi\) has the Kurdyka-Lojasiewicz property at \((L_\infty, H_\infty)\) with \(\phi(s) = cs^{1-\alpha}, \alpha \in (0, 1), c > 0\). Then we have

- If \(\alpha = 0\), then the sequence \((L_k, H_k)\) converges in a finite number of steps.

- If \(\alpha \in (0, \frac{1}{2}]\), then there is \(c > 0\) and \(\tau \in [0, 1)\) such that

\[
\|(L_k, H_k) - (L_\infty, H_\infty)\| \leq c\tau^k.
\]

- If \(\alpha \in (\frac{1}{2}, 1)\), then there is \(c > 0\) such that

\[
\|(L_k, H_k) - (L_\infty, H_\infty)\| \leq ck^{1-\alpha}.
\]
CHAPTER 6
EXTENSION TO N-D SIGNALS

6.1 Introduction

This chapter is concerned with reconstruction of two-dimensional (2-D) signals from partial observed samples, which arises in many applications such as radar, inverse scattering, and super-resolution imaging. Suppose that the signal under study is a linear combination of \( R \) continuous-valued 2D complex sinusoids. The goal is to reconstruct a 2D spectrally sparse signal from a small size of randomly observed samples.

Conventional approaches include 2-D unitary ESPRIT, 2-D MUSIC, the Matrix Enhancement Matrix Pencil (MEMP) method, etc, which are based on parametric representation. However, many of these approaches require equi-spaced time-domain samples. They also rely on prior knowledge on the number of sinusoids. Moreover, these methods are often sensitive the number of sinusoids and noise.

Another line of work is concerned with Compressed sensing (CS). CS suggests that it is possible to recover a spectrally sparse signal from a very small size of time-domain samples. However, CS assumes the continuous-valued frequencies lie on the DFT grid, which causes basis mismatch.

Several recent works deal directly with continuous-valued frequencies, including 2D Atomic Norm Minimization (ANM) and Enhanced Matrix Completion (EMaC) and so on. However, since all these methods use convex optimization and are finally
rewritten as semi definite programs (SDPs), the numerical experiments were conducted using the advanced SDP solver SDPT3, this solver is based on interior point methods, which are typically inapplicable to large-scale data. In fact, SDP3 fails to handle \( n \times n \) data matrix signal when \( n \) exceeds 19. And thus these methods do not work for large-scale high-dimensional signals.

In this chapter, in order to efficiently reconstruct large-scale spectrally sparse signals, we extend our PWGD algorithm to 2D frequency models, which was called PWGD2D algorithm. Inspired by Enhanced Matrix Completion (EMaC) and traditional matrix pencil method, PWGD2D algorithm is based upon 2-fold Hankel matrix, whose rank is the same as the spectral sparsity \( R \). To speed up our proposed algorithm, an acceleration technique scheme similar to FISTA [2] is given. The practical applicability of our algorithm is validated by numerical experiments, which show that our algorithms can recover high-dimensional signals as a superposition of multiple sinusoids.

This chapter is organized as follows. In section 2, we first introduce the 2D signal model, then we discuss several existing algorithms such as conventional compressed sensing, ANM and EMaC algorithm, in section 3, we formulate the problem to reconstruct the 2D signal and propose our algorithm PWGD2D, then we discuss the convergence of the algorithm and give the acceleration of our algorithm by FISTA. Finally, in the last section, we discuss the noisy case of the algorithms.

**Notations:** We denote the set of complex numbers as \( \mathbb{C} \). We reserve calligraphic uppercase letters for index sets. When we use an index set \( K \) as the subscript
of a vector \( \mathbf{x} \), i.e., \( \mathbf{x}_K \), it represents the part of the vector \( \mathbf{x} \) over index set \( K \). We use the superscripts \( H \), \( T \), and \( \ast \) for matrix conjugate transpose, transpose, and conjugate respectively. \( \text{Tr}(\cdot) \) represents the trace function. We write the Frobenius norm of a matrix \( A \in \mathbb{C}^{m \times n} \) as \( \|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |A_{ij}|^2} \), where \( A_{ij} \) is the element of \( A \) in its \( i \)-th row and \( j \)-th column.

### 6.2 2D Complex Signal and Problem Formulation

#### 6.2.1 2D Signal Model

Without loss of generality, consider a two-dimensional (2D) square data matrix \( \mathbf{X}^{(\text{true})} \) of size \( n = (4N+1) \times (4N+1) \), where \( \mathcal{J} = \{0, 1, \ldots, 4N\} \times \{0, 1, \ldots, 4N\} \) denotes the union of the indices of \( \mathbf{X}^{(\text{true})} \). The assumption is imposed to simplify the development of the theoretical guarantees, and can be removed with little modifications, see [37] for a similar treatment. Each entry of \( \mathbf{X}^{(\text{true})} \) can be expressed as a weighted sum of \( R \) two-dimensional complex sinusoids at the time index \( t = [t_1, t_2] \in \mathcal{J} \), i.e.,

\[
\mathbf{X}^{(\text{true})}_{t_1, t_2} = \sum_{k=1}^{R} d_k^{(\text{true})} e^{2\pi i (f_k^{(\text{true})})^T \cdot t} = \sum_{k=1}^{R} d_k^{(\text{true})} e^{2\pi i f_{1k}^{(\text{true})} t_1} e^{2\pi i f_{2k}^{(\text{true})} t_2},
\]

where \( d_k^{(\text{true})} \) represents the complex amplitude associated with each \( 1 \leq k \leq R \). Let \( \Omega = \{ f_k^{(\text{true})} = [f_{1k}^{(\text{true})}, f_{2k}^{(\text{true})}] \in [0, 1) \times [0, 1), 1 \leq k \leq R \} \) be the set of distinct frequencies. For notational simplicity, we introduce the following unit-norm atoms:

\[
\begin{align*}
\mathbf{a}(f_{1k}^{(\text{true})}) &= [1, y_k, \ldots, y_k^{4N}]^T, \\
\mathbf{a}(f_{2k}^{(\text{true})}) &= [1, z_k, \ldots, z_k^{4N}]^T,
\end{align*}
\]
where $y_k = e^{2\pi i f_{1k}^{(true)}}$ and $z_k = e^{2\pi i f_{2k}^{(true)}}$. This allows us to write $X^{(true)}$ in a matrix form as follows

$$X^{(true)} = YDZ^T,$$

where $Y, Z$ are given by

$$Y = \begin{bmatrix} a(f_{11}^{(true)}), & \ldots, & a(f_{1R}^{(true)}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ y_1 & y_2 & \ldots & y_R \\ \vdots & \vdots & \ddots & \vdots \\ y_1^{4N} & y_2^{4N} & \ldots & y_R^{4N} \end{bmatrix} \in \mathbb{C}^{(4N+1)\times R},$$

and

$$Z = \begin{bmatrix} a(f_{21}^{(true)}), & \ldots, & a(f_{2R}^{(true)}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ z_1 & z_2 & \ldots & z_R \\ \vdots & \vdots & \ddots & \vdots \\ z_1^{4N} & z_2^{4N} & \ldots & z_R^{4N} \end{bmatrix} \in \mathbb{C}^{(4N+1)\times R},$$

and

$$D = \text{diag}([d_1, d_2, \ldots, d_R]) = \text{diag}(d)$$

$$= \begin{bmatrix} d_1^{(true)} \\ d_2^{(true)} \\ \vdots \\ d_R^{(true)} \end{bmatrix} \in \mathbb{C}^{R\times R}.$$
Denote by $\mathbf{x}^{(\text{true})} = \text{vec}((\mathbf{X}^{(\text{true})})^T) \in \mathbb{C}^{(4N+1)^2}$ the vectorized data matrix, then one has

$$
\mathbf{x}^{(\text{true})} = \mathbf{Z} \mathbf{d} = \sum_{k=1}^{d} d^{(\text{true})}_k \mathbf{a}(f^{(\text{true})}_1) \otimes \mathbf{a}(f^{(\text{true})}_2) = \sum_{k=1}^{d} d^{(\text{true})}_k \mathbf{c}(f^{(\text{true})}_k),
$$

where $\otimes$ represents Kronecker product, and

$$
\mathbf{c}(f^{(\text{true})}_k) = \mathbf{c}(f^{(\text{true})}_1, f^{(\text{true})}_2) := \mathbf{a}(f^{(\text{true})}_1) \otimes \mathbf{a}(f^{(\text{true})}_2) \in \mathbb{C}^{(4N+1)^2}.
$$

We assume that $M$ entries of $\mathbf{X}^{(\text{true})}$ are observed uniformly at random. Specifically, denote by $\mathcal{T} \subset \mathcal{J}$ as the index set such that $X^{(\text{true})}_{t_1, t_2}$ are observed if and only if $(t_1, t_2) \in \mathcal{T}$. Also denote $\Omega$ as the corresponding index set of the vectorized signal $\mathbf{x}$.

The focus of this chapter is to recover the unobserved entries of the original data matrix $\mathbf{X}^{(\text{true})}$. We note that the frequencies $\Omega$ can also be recovered using conventional approaches such as MEMP method [25] once the data matrix is recovered.

### 6.2.2 Existing Algorithms in Absence of Noise

There are several existing algorithms for recovering the signal from the incomplete observations $\mathbf{y} = \mathbf{x}_\mathcal{T}^{(\text{true})}$. The first one we want to refer to is conventional compressed sensing approach.

In order to apply the theory of compressed sensing, we represent $\mathbf{x}^{(\text{true})}$ as a sparse signal in a pre-determined basis by discretizing the 2-D plane $[0, 1) \times [0, 1)$ with grid points $t = [t_1, t_2] \in \Theta$, where $t_1, t_2 \in \{0, ..., \frac{4N}{4N+1}\}$. Write the result DFT basis as

$$
\mathbf{F} = [\mathbf{c}(t)_{t \in \Theta}] = \mathbf{F}_1 \otimes \mathbf{F}_1 \in \mathbb{C}^{(4N+1)^2 \times (4N+1)^2},
$$

where $\mathbf{F}_1$ is a DFT matrix with dimension $(4N+1) \times (4N+1)$. The vectorized signal
\( \mathbf{x}^{(\text{true})} \) can then be represented as

\[
\mathbf{x}^{(\text{true})} = \mathbf{F} \mathbf{c},
\]

and the observed samples can be written as

\[
\mathbf{y} = \mathbf{F}_T \mathbf{c},
\]

where \( \mathbf{F}_T \) are partial rows of \( \mathbf{F} \), and \( \mathbf{c} \) is approximately sparse. Compressed sensing suggests that we could recover \( \mathbf{x}^{(\text{true})} \) using the \( l_1 \)-minimization as

\[
\min_{\mathbf{c}} \| \mathbf{c} \|_1 \quad \text{s.t.} \quad \mathbf{F}_T \mathbf{c} = \mathbf{y}. \tag{1}
\]

The major issue with the above approach is that the frequencies \( f_k \) never lie perfectly on the grid \( \Theta \), resulting in basis mismatch between the true frequencies and the discrete grid.

Off-the-grid compressed sensing is another line of signal recovery. In [14], the authors propose the following atomic norm minimization to recover a two-dimensional (2D) spectrally sparse signal \( \mathbf{x}^{(\text{true})} \) using randomly observed time domain samples \( \Omega \in \mathcal{J} \):

\[
\min_{\mathbf{x}} \| \mathbf{x} \|_A \quad \text{s.t.} \quad x_i = x_i^{(\text{true})}, i \in \Omega, \tag{2}
\]

here the atomic set is defined as

\[
\mathcal{A} := \{ \mathbf{c}(\mathbf{f}^{(\text{true})}) | \mathbf{f}^{(\text{true})} \in [0, 1) \times [0, 1) \},
\]

and the atomic norm for a vectorized 2D signal \( \mathbf{x} = \text{vec}(\mathbf{X}^T) \) is defined as

\[
\| \mathbf{x} \|_A := \inf_{\mathbf{f} \in [0,1) \times [0,1), d_k \in \mathbb{C}} \left\{ \sum_k |d_k| \| \mathbf{x} = \sum_k d_k \mathbf{c}_k(\mathbf{f}_k) \right\}.
\]
The authors propose to approximate the atomic norm minimization algorithm in (2) via the following semi-definite program

$$\min_{\mathbf{T}, \mathbf{x}, t} \frac{1}{2} \text{Tr}(S(\mathbf{T})) + \frac{1}{2} t$$

s.t. \hspace{1cm} \mathbf{U}_1 = \begin{bmatrix} S(\mathbf{T}) & \mathbf{x} \\ \mathbf{x}^H & t \end{bmatrix} \succeq 0,$$

$$x_t = x_t^{(\text{true})}, \ t \in \mathcal{T},$$

(3)

here the matrix $\mathbf{T} := [x_{l_1, l_2}]$ with $-n_1 < l_1 < n_1, -n_2 < l_2 < n_2$, and $S(\mathbf{T})$ is the corresponding two-fold block Toeplitz matrix constructed from $\mathbf{T}$,

$$S(\mathbf{T}) = \begin{bmatrix} \mathbf{T}_0 & \mathbf{T}_{-1} & \cdots & \mathbf{T}_{-(n_1-1)} \\ \mathbf{T}_1 & \mathbf{T}_0 & \cdots & \mathbf{T}_{-(n_1-2)} \\ \mathbf{T}_2 & \mathbf{T}_1 & \cdots & \mathbf{T}_{-(n_1-3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{T}_{(n_1-1)} & \mathbf{T}_{(n_1-2)} & \cdots & \mathbf{T}_0 \end{bmatrix},$$

where each block $\mathbf{T}_l(-n_1 < l < n_1)$ is an $n_2 \times n_2$ Toeplitz matrix defined from the $l$th row of $\mathbf{T}$:

$$\mathbf{T}_l = \begin{bmatrix} x_{l,0} & x_{l,-1} & \cdots & x_{l, -(n_2-1)} \\ x_{l,1} & x_{l,0} & \cdots & x_{l, -(n_2-2)} \\ x_{l,2} & x_{l,0} & \cdots & x_{l, -(n_2-3)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{l,(n_2-1)} & x_{l,(n_2-2)} & \cdots & x_{l,0} \end{bmatrix}.$$ 

We refer the reader to [14] for more details. The atomic norm minimization requires a certain minimum separation condition between frequencies for successful recovery.
Enhanced Matrix Completion algorithm (EMaC) is another off-grid signal reconstruction method developed by Chen and Chi [13]. Inspired by matrix pencil method, the authors exploit the 2-fold block Hankel structure to build a structured matrix completion algorithm, which aims to recover a spectrally sparse signal from randomly observed samples. The 2D EMaC algorithm is given as follows

$$\min_{X} \|X_e\|_s$$ \hspace{1cm} (4)

s.t. \quad X_{j,k} = X_{j,k}^{(\text{true})}, (j,k) \in T,$$

the enhanced matrix $X_e$ with respect to the 2D signal $X$ is defined as a $k_1 \times (|J| - k_1 + 1)$ block Hankel matrix

$$X_e := \begin{bmatrix}
X_0 & X_1 & \cdots & X_{4N-k_1} \\
X_1 & X_2 & \cdots & T_{4N-k_1+1} \\
X_2 & X_3 & \cdots & T_{4N-k_1+2} \\
\vdots & \vdots & \ddots & \vdots \\
X_{(k_1-1)} & X_{k_1} & \cdots & X_{4N-1}
\end{bmatrix},$$

where $1 \leq k_1 \leq 4N$. Each block is a $k_2 \times (4N - k_2 + 1)$ Hankel matrix defined such that for every $l(0 \leq l \leq 4N)$:

$$X_l := \begin{bmatrix}
X_{l,0} & X_{l,1} & \cdots & X_{l,4N-k_2} \\
X_{l,1} & X_{l,2} & \cdots & X_{l,4N-k_2+1} \\
X_{l,2} & X_{l,3} & \cdots & X_{l,4N-k_2+2} \\
\vdots & \vdots & \ddots & \vdots \\
X_{l,k_2-1} & X_{l,k_2} & \cdots & X_{l,4N-1}
\end{bmatrix},$$

where $1 \leq k_2 \leq 4N$. 


The convex program (4) minimizes the nuclear norm of the enhanced form of all matrices compatible with the samples. It can be rewritten into a semi-definite program (SDP)

$$\min_{X} \frac{1}{2} \text{Tr}(Q_1) + \frac{1}{2} \text{Tr}(Q_2)$$

s.t. $U_2 = \begin{bmatrix} Q_1 & H(X) \\ H(X)^H & Q_2 \end{bmatrix} \succeq 0,$

$$X_{j,k} = X_{j,k}^{(\text{true})}, (j, k) \in T.$$ (5)

Let $X_e = U \Sigma V^H$ be the singular value decomposition of the matrix $H(X)$, and the matrices $Q_1 := U \Sigma U^H$ and $Q_2 := V \Sigma V^H$. We refer the reader to [32] for more details.

6.2.3 Problem Formulation

Our proposed algorithm first converts the reconstruction of $X^{(\text{true})}$ from 3.2.1 to a Hankel matrix completion problem. This strategy has also been used in [13].

The two-fold Hankel structure of the signal $X^{(\text{true})}$ is defined as a $k_1 \times (4N - k_1 + 1)$ block Hankel matrix

$$X_e = \begin{bmatrix} X_0 & X_1 & \cdots & X_{4N-k_1} \\ X_1 & X_2 & \cdots & X_{4N-k_1+1} \\ X_2 & X_3 & \cdots & X_{4N-k_1+2} \\ \vdots & \vdots & \ddots & \vdots \\ X_{k_1-1} & X_{k_1} & \cdots & X_{4N-1} \end{bmatrix}.$$ (6)

Each block is a $k_2 \times (4N - k_2 + 1)$ Hankel matrix defined as follows: for every
\( l(0 \leq l \leq 4N) \):

\[
X_l := \begin{bmatrix}
X_{l,0} & X_{l,1} & \cdots & X_{l,4N-k_2} \\
X_{l,1} & X_{l,2} & \cdots & X_{l,4N-k_2+1} \\
\vdots & \vdots & \ddots & \vdots \\
X_{l,k_2-1} & X_{l,k_2} & \cdots & X_{l,4N-1}
\end{bmatrix}, \tag{7}
\]

where \( 1 \leq k_1, k_2 \leq 4N \) are called pencil parameters. Notice that each block can be expressed as

\[
X_l = Z_L Y_d^l D Z_R, \tag{8}
\]

where \( Z_L, Z_R \) and \( Y_d \) are defined as

\[
Z_L := \begin{bmatrix}
1 & 1 & \cdots & 1 \\
z_1 & z_2 & \cdots & z_R \\
\vdots & \vdots & \ddots & \vdots \\
z_1^{k_2-1} & z_2^{k_2-1} & \cdots & z_R^{k_2-1}
\end{bmatrix},
\]

\[
Z_R := \begin{bmatrix}
1 & z_1 & \cdots & z_1^{k_2-1} \\
1 & z_2 & \cdots & z_2^{k_2-1} \\
\vdots & \vdots & \ddots & \vdots \\
1 & z_R & \cdots & z_R^{k_2-1}
\end{bmatrix},
\]

and

\[
Y_d := \text{diag}([y_1, y_2, \ldots, y_R]) = \begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
y_R
\end{bmatrix}.
\]
Plug (8) into (6). We then have the following:

\[
X_e = \begin{bmatrix}
Z_L \\
Z_L Y_d \\
\vdots \\
Z_L Y_{d^{k_1-1}}
\end{bmatrix}
D \begin{bmatrix}
Z_R, Y_d Z_R, \cdots, Y_d^{4N-k_2} Z_R
\end{bmatrix},
\]

This immediately implies that \(X_e\) is low-rank, i.e.

\[
\text{rank}(X_e) \leq R.
\]

Then, instead of constructing the true signal \(X^{(\text{true})}\) directly, we reconstruct the rank-R two-fold block Hankel matrix \(X_e\). Since the transformation is one-to-one from a data matrix to a block Hankel matrix, one can easily convert the reconstructed Hankel matrix back to a signal.

Instead of solving the convex optimization problem like (4), the signal reconstruction problem is formulated as

\[
\text{Find a matrix } X \\
\text{s.t.} \quad \text{rank}(X_e) \leq R,
\]

\[
X_{jk} = X_{jk}^{(\text{true})}, \quad (j, k) \in T,
\]

\(X_e\) is two-fold Hankel.

Similar to 1-D case, (9) can also be converted to a rank minimization problem and further relaxed to (4). However, there were no efficient ways to compute it for large problem dimensions. In this thesis, instead of considering convex optimizations (4), we aim at attacking the original non-convex problem (9) directly.
6.3 Our Algorithms

6.3.1 Our Algorithm

Our signal reconstruction problem can be converted to the non-convex problem (9), that is to say, our aim is to find a matrix $X$ such that it is a two-fold block Hankel matrix as given in last section and consistent with observed data and the rank of the enhanced form is no greater than $R$.

First, let us define the set of all complex-valued two-fold block Hankel matrices as given above and that are consistent with the observed data

$$\mathcal{H} = \{H_e \mid H \in \mathbb{C}^{(4N+1) \times (4N+1)}, H^T = X^{(\text{true})} \}.$$  \hfill (10)

Next, define the set of all complex-valued matrices with rank no greater than $R$ as

$$\mathcal{R}_C^R = \{ L \in \mathbb{C}^{(k_1k_2) \times (4N-k_1+1)(4N-k_2+1)} \mid \text{rank}(L) \leq R \}.$$ \hfill (11)

The set $\mathcal{R}_C^R$ is an algebraic variety and $\mathcal{H}$ is an affine space. Then, the signal recovery problem and (9) can be formulated as following optimization problem

$$\min_{L \in \mathcal{R}_C^R, H_e \in \mathcal{H}} F(L, H_e) := \frac{1}{2}\|L - H_e\|_F^2.$$ \hfill (12)

Now we present our projected Wirtinger gradient algorithm for solving (9).

Our algorithm can be seen as a variant of classical alternating projection methods. Given initial estimate $H_0 = X^{(\text{true})}_e$ and $L_0 = 0$, the iterative scheme of projected gradient descent algorithm has the form,

$$\begin{cases} L_{t+1} \in \mathcal{P}_{\mathcal{R}_C^R}(L_t - \delta_1 \nabla_{L_t} F(L_t - H_t)), \text{ for some } 0 < \delta_1 < 1, \\ H_{t+1} \in \mathcal{P}_{\mathcal{H}}(H_t - \delta_2 \nabla_{H_t} F(H_t - L_{t+1})), \text{ for some } 0 < \delta_2 < 1, \end{cases}$$ \hfill (13)
Algorithm 6.1 2-D Projected Gradient Descent Algorithm

1. **Input:** The observed signal $X^{(true)}_0$ on the location set $\mathcal{M}$.
2. **Initialize:** Form the two-fold block Hankel matrix $H_0 \in \mathcal{H}$, $L_0 = 0$, $0 < \delta_1, \delta_2 < 1$
3. **Repeat:**
   - For each $t = 0, 1, \cdots$ generate a sequence $\{L_t, H_t\}_{t \in \mathbb{N}}$ as follows:
     - **Step 1:** $L_{t+1} \in P_{\mathbb{R}}(L_t - \delta_1 \nabla_{L_t} F(L_t - H_t))$
     - **Step 2:** $H_{t+1} \in P_{\mathbb{H}}(H_t - \delta_2 \nabla_{H_t} F(H_t - L_{t+1}))$
4. **until:** the stopping criteria is met
5. **Output:** the matrix $H_{t+1}$

The objective $F(L, H_e) := \frac{1}{2}\|L - H_e\|^2_F$ is a real-valued function with complex variables, which is not differentiable in the ordinary complex calculus sense. Nevertheless, $F(L, H_e)$ is differentiable with respect to the real and imaginary parts of its variables. Thus, our gradient descent algorithm is performed on the real and imaginary parts respectively. Denote

$$Z = \begin{bmatrix} L \\ H_e \end{bmatrix} = \Re + i\Im$$

where $\Re$ and $\Im$ are the real and imaginary parts of $Z$. Rewrite $F$ as $F(\Re, \Im)$. Then, in our gradient descent algorithm, $\Re$ is updated by $\frac{\partial F}{\partial \Re}$ and $\Im$ by $\frac{\partial F}{\partial \Im}$. In other words, $Z$ is updated by $\frac{\partial F}{\partial \Re} + i\frac{\partial F}{\partial \Im}$. By Wirtinger calculus, we have the relation

$$\nabla F = \frac{\partial F}{\partial \Re} + i\frac{\partial F}{\partial \Im} = 2\left(\frac{\partial F}{\partial Z}\right)^*.$$  

Direct calculation gives

$$2\left(\frac{\partial F}{\partial Z}\right)^* = \begin{bmatrix} 2\left(\frac{\partial F}{\partial L}\right)^* \\ 2\left(\frac{\partial F}{\partial H_e}\right)^* \end{bmatrix} = \begin{bmatrix} L - H_e \\ H_e - L \end{bmatrix}.$$
With the Wirtinger gradient, our proposed algorithm (6.1) can become

\[
\begin{aligned}
L_{t+1} &\in \mathcal{P}_{\mathcal{C}^R}(L_t - \delta_1(L_t - H_t)), \\
H_{t+1} &\in \mathcal{P}_{\mathcal{H}}(H_t - \delta_2(H_t - L_{t+1})),
\end{aligned}
\]  

(14)

where \(\delta_1 > 0\) and \(\delta_2 > 0\) are step sizes, and \(\mathcal{P}_{\mathcal{C}^R}\) and \(\mathcal{P}_{\mathcal{H}}\) are projections onto \(\mathbb{C}^R\) and \(\mathcal{H}\) respectively. We call (14) 2-D projected Wirtinger gradient descent (PWGD2D).

**Algorithm 6.2 2-D Projected Wirtinger Gradient Descent Algorithm**

1. **Input:** The observed signal \(X^{(\text{true})}_0\) on the location set \(\mathcal{M}\).
2. **Initialize:** Form a two-fold block Hankel matrix \(H_0 \in \mathcal{H}, L_0 = 0, 0 < \delta_1, \delta_2 < 1\)
3. **Repeat:**
   - For each \(t = 0, 1, \ldots\) generate a sequence \(\{L_t, H_t\}_{t \in \mathbb{N}}\) as follows:
     - Step 1: \(L_{t+1} \in \mathcal{P}_{\mathcal{C}^R}(L_t - \delta_1(L_t - H_t))\)
     - Step 2: \(H_{t+1} \in \mathcal{P}_{\mathcal{H}}(H_t - \delta_2(H_t - L_{t+1}))\)
4. **until:** the stopping criteria is met
5. **Output:** the matrix \(H_{t+1}\)

It remains to find out \(\mathcal{P}_{\mathcal{C}^R}\) and \(\mathcal{P}_{\mathcal{H}}\) respectively, here the projections are defined as follows: \(\mathcal{P}_{\mathcal{C}^R}(X)\) is the projection of \(L_t\) onto the space of all complex-valued matrices with rank smaller than \(R\), it is the best-rank-R-approximation to \(X\) by Eckhart-Young Theorem [23],

\[
\mathcal{P}_{\mathcal{C}^R}(X) = U_R \Sigma_R V_R^H,
\]

where \(\Sigma_R\) is the diagonal matrix that only retains \(R\) largest nonnegative singular values of \(X\), and \(U_R, V_R\) are the matrices whose columns are the corresponding left and right singular vectors respectively.
\( \mathcal{P}_{\mathcal{H}}(\cdot) \) denotes the projection of \( H_t \) onto the subspace of the two-fold block Hankel matrices that are consistent with the observed data.

Based on Attouch and Boltes theory [1] [4], we provide convergence analysis of the algorithm of 1D case in chapter 5. The convergence analysis for 2D case is very similar to 1D case. We omit the details of the proofs here.

6.3.2 Acceleration by a FISTA-scheme

In this subsection, we propose a scheme to accelerate the convergence of the PWGD2D algorithm (14). Our scheme borrows from the fast iterative shrinkage-thresholding algorithm (FISTA) [2], which has been proven to be efficient in minimizing the sum of two convex functions with one having a Lipschitz continuous gradient. The basic idea is to use a specific linear combination of two successive iterates. Although our problem is non-convex, we still can employ the linear combination scheme in FISTA for our model.

Our PWGD with FISTA scheme, called PWGD-FISTA, is constructed as follows: Given \( k_0 = 1 \), we generate \( \{L_t, H_t\} \) by

\[
\begin{align*}
L_{t+1} &\in \mathcal{P}_{\mathcal{H}}(L_t - \delta_1(L_t - \tilde{H}_t)), \\
H_{t+1} &\in \mathcal{P}_{\mathcal{H}}(H_t - \delta_2(\tilde{H}_t - L_{t+1})), \\
k_{t+1} &= \frac{\sqrt{1+4k_t^2}+1}{2}, \\
\tilde{H}_{t+1} &= H_{t+1} + \frac{k_t-1}{k_{t+1}}(H_{t+1} - H_t)
\end{align*}
\] (15)
Algorithm 6.3 FISTA-PWGD2D

1. **Input:** The observed signal $X_0^{(true)}$ on the location set $\mathcal{M}$.
2. **Initialize:** Form a two-fold Hankel matrix $H_0 \in \mathcal{H}, L_0 = 0, k_0 = 1, 0 < \delta_1, \delta_2 < 1$
3. **Repeat:**
   - For each $t = 0, 1, \cdots$ generate a sequence $\{L_t, H_t\}_{t \in \mathbb{N}}$ as follows:
     - Step 1: $L_{t+1} \in \mathcal{P}_{\mathcal{H}}(L_t - \delta_1 (L_t - \hat{H}_t))$
     - Step 2: $H_{t+1} \in \mathcal{P}_{\mathcal{H}}(H_t - \delta_2 (H_t - L_{t+1}))$
   - Step 3: $k_{t+1} = \frac{\sqrt{1+4k_t^2}+1}{2}$
   - Step 4: $\hat{H}_{t+1} = H_{t+1} + \frac{k_t-1}{k_{t+1}}(H_{t+1} - H_t)$
4. **until:** the stopping criteria is met
5. **Output:** the matrix $H_{t+1}$.

Since $\mathcal{H}$ is an affine subspace, the linear combination in the last line of (15) does not change of the feasibility of $\hat{H}_{t+1}$, i.e., $\hat{H}_{t+1} \in \mathcal{H}$. This makes the computational complexity and storage of Step 1 and Step 2 in the PWGD-FISTA algorithm is the same as that in the PWGD algorithm (14). Also, the computational effort in Step 3 and Step 4 of (15) is negligible compared with that in Step 1 and Step 2. Therefore, the PWGD-FISTA2D algorithm preserves the computational simplicity of the PWGD2D algorithm.

6.3.3 The Noisy-PWGD2D with Bounded Noise

In practice, measurements are often contaminated by a certain amount of noise. To make our model and algorithm more practically applicable, we replace our measurements by $X^{(noisy)}$ by the following noisy model

$$X^{(noisy)} = X^{(true)} + \sigma N,$$
where \( \mathbf{X}^{(\text{true})} \) is the observed clean signal, and \( \mathbf{N} \) is standard additive white Gaussian noise (AWGN) with each entry i.i.d from \( \mathcal{N}(0, 1) \). The signal-to-noise ratio is defined as \( \text{SNR} = 10 \log_{10} \frac{M}{(2N-1)\sigma^2} \), which has been scaled with respect to the number of observations. We assume that the noise magnitude is bounded by a known amount, \( \| \mathbf{X}_{j,k}^{(\text{noisy})} - \mathbf{X}_{j,k}^{(\text{true})} \|_F = \| \sigma \mathbf{N}_{j,k} \|_F \leq \delta, (j, k) \in \mathcal{T} \). In order to adapt our algorithm to such noisy measurement, the variation in the estimate should be very small. Our algorithm is then modified as follows

\[
\text{Find a matrix } \mathbf{X} \\
\text{s.t. } \text{rank}(\mathbf{X}_e) \leq R, \quad (16)
\]

\[
X_{jk} = X_{jk}^{(\text{noisy})}, (j, k) \in \mathcal{T},
\]

\( \mathbf{X}_e \) is two-fold Hankel,

While in the paper [14], the noisy 2-D Atomic norm minimization algorithm is proposed as follows

\[
\min_{\mathbf{x}} \| \mathbf{x} \|_A \\
\text{s.t. } \| x_i - x_i^{(\text{noisy})} \|_F \leq \delta, t \in \Omega. \quad (17)
\]

Here \( \mathbf{x} \) is the vectorized form of the original signal.

In [13], the authors also give the noisy case of Enhanced matrix completion algorithm,

\[
\min_{\mathbf{x}} \| \mathbf{X}_e \|_* \\
\text{s.t. } \| X_{j,k} - X_{j,k}^{(\text{noisy})} \|_F \leq \delta, (j, k) \in \mathcal{T}. \quad (18)
\]
6.3.4 Numerical Experiment: The Reconstruction of 2D Noisy Signal

6.1 illustrates the performance of PWGD2D. We generate a $101 \times 101$ data matrix $X$ that contains 2 frequencies, and revealed 0.8 of the total entries uniformly at random. The noise was i.i.d Gaussian giving a signal-to-noise ratio of 10. The normalized reconstruction error $\|\hat{X} - X\|_F / \|X\|_F = 0.1685$, validating the stability of our algorithm in the presence of noise.

![Figure 6.1](image.png)

Figure 6.1. The performance of noisy PWGD2D for a $101 \times 101$ data matrix that contains 2 random frequencies. 0.8 of all entries are observed with signal-to-noise ratio is 10. Here, the reconstructed data against the true data for the first 500 time instances (after vectorized) are plotted.
6.4 Extension to Higher-Dimensional Models

The PWGD algorithm and the convergence theory can immediately extend to higher dimensional frequency models without difficulty. In fact, for \( n \)-dimensional frequency models, one can arrange the original data into a \( n \)-fold Hankel matrix of rank at most \( R \). For instance, consider the following 3D model:

\[
X_{t_1,t_2,t_3} = \sum_{k=1}^{R} d_k^{(true)} e^{2\pi i f_{1k}^{(true)} t_1} e^{2\pi i f_{2k}^{(true)} t_2} e^{2\pi i f_{3k}^{(true)} t_3}, \forall (t_1,t_2,t_3) \in n_1 \times n_2 \times n_3.
\]

An enhanced form can be defined as a 3-fold Hankel matrix such that

\[
X_e = \begin{bmatrix}
X_{0,e} & X_{1,e} & \cdots & X_{n_3-k_3,e} \\
X_{1,e} & X_{2,e} & \cdots & X_{n_3-k_3+1,e} \\
\vdots & \vdots & \ddots & \vdots \\
X_{k_3-1,e} & X_{k_3,e} & \cdots & X_{n_3-1,e}
\end{bmatrix},
\]

where \( X_{i,e} \) denotes the 2D enhanced form of the matrix consisting of all entries \( X_{t_1,t_2,t_3} \) obeying \( t_3 = i \). One can verify that \( X_e \) is of rank at most \( R \) and thereby apply PWGD algorithm on the 3D enhanced form. To summarize, for \( n \)-dimensional frequency models, PWGD (resp. Noisy-PWGD) works for all \( n \)-fold Hankel matrices that are consistent with the observed data. And the convergence theory can be similarly extended based on Attouch and Bolte’s theory. We omit the details here.
CHAPTER 7
CONCLUSION

In this thesis, a fast iterative algorithm is proposed to reconstruct spectrally sparse signals whose frequencies can be any values in the continuous domain \([0, 1)\) from a small size of time domain samples. Different from other existing algorithms, our algorithm is based upon a non-convex optimization problem, which is able to deal with signal of large scale. The convergence analysis of the algorithm is provided and numerical analysis is also given. We also provided an acceleration of the proposed algorithm, which was inspired by the scheme of FISTA. Then we extend our algorithms to signals in the presence of noise, which shows that our algorithms work better for noisy signals than the existing algorithms, including atomic norm minimization (ANM) and enhanced matrix completion (EMaC) algorithms. Finally we extend our algorithms to high dimensional signals.

There are still some open questions. First of all, numerical experiments have been used to show that signals can be successfully recovered from a very small size of time domain samples, and from the experiments, we can see that our algorithm have better recovery performance than the algorithms like atomic norm minimization (ANM) and enhance matrix completion (EMaC), thus it would be of great interest to find theoretical bound of the sample size for successful recovery of our algorithm. We leave the theoretical analysis of the bound for future work. Secondly, we are interested in apply our algorithms to the damped signals, especially the signals in NMR spectroscopy. The last but not the least, one of the main step of PWGD is
to compute SVD of a Hankel matrix, however, when dealing with really large-scale signals, we quickly run into memory issues in forming the large Hankel matrix and CPU issues due to running SVD itself. In theory, it seems like we don’t have to form the Hankel matrix itself. As long as we have the elements of either the first row or column of the matrix, we have complete information of all the entries of the Hankel matrix, and we should be able to perform the SVD without even forming this large matrix in memory. In the fact, there are matrix free algorithms (algorithms that use only matrix-vector multiplications rather than working directly with the entries of the matrix) that can compute approximate values of a few singular values/vectors of a large matrix. Since you want a low rank approximation to this matrix, you could use such an algorithm to find the $R$ largest singular values and associated singular vectors. One such algorithm uses Lanczos with fast block matrix-vector multiplication via FFTs, the theory is not hard to understand, but the main difficulty is solving the programming problems, which will be left for future work.
REFERENCES


