Convex optimization tools in complex frequency estimation

Jamie Caprani

Abstract

In the field of spectral estimation an important class of problems arises from models of sums of complex exponentials (sinusoids with decay),

$$F(t) = \sum_{j=1}^{K} c_j e^{\zeta_j t}, t \in \mathbb{R}, c_j, \zeta_j \in \mathbb{C}$$

where $\zeta_j = 2\pi(\alpha_j + i\beta_j)$ is a complex frequency mode with frequency parameter $i\beta_j$ and damping parameter α_j, c_j is a complex amplitude. In real life applications a recovered signal will be affected by noise so naturally our model will include noise, which we denote as $e(t)$.

$$F(t) = \sum_{j=1}^{K} c_j e^{\zeta_j t} + e(t)$$

In general, this becomes an approximation problem to recover the original signal for which there are many methods, e.g. Min-Norm, Music, ESPRIT,Non Linear LS.

It is intention of this paper to apply some new methods being developed in the field of optimization at Lund University, to the spectral estimation problem. These results rely on convex analysis, and include a new transform which is called $\mathcal{S}^2$. We will present some examples through already established concepts, with particular focus on the ESPRIT method for frequency estimation, before showcasing the new results.

Contents

1. Frequency estimation

Any signal or function $F(t)$ can be expressed as a sum of infinitely many exponential functions $e^{i\eta t}$ via the Fourier transform;

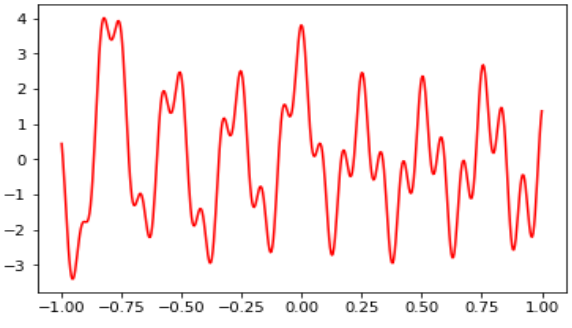
$$F(t) = \sum \hat{F}(\eta) e^{i\eta t} \tag{1}$$

where we sum over all possible frequencies η. The field of "Frequency estimation" can be viewed as the problem of retrieving the frequency content $\hat{F}$ from measurements of F. In many scenarios, this is solved by simply applying FFT (the Fast Fourier Transform) to samples of F. However, if F is poorly sampled or the signal contains high levels of noise, the problem is more complicated. In many applications, it is reasonable to assume that F has a very small frequency content, i.e. the sum (1) contains only a few non-negligible components, and this additional information can be incorporated in the reconstruction process to yield a better estimate of the frequency content of F.

In this thesis we will develop some optimization methods to solve the closely related problem of *sparse complex frequency estimation*. Here, we instead assume that the signal comprises a few exponential functions with both real and complex frequencies, i.e.

$$F(t) = \sum_{j=1}^{K} c_j e^{\zeta_j t}, t \in [-1, 1] \tag{2}$$

where $\zeta_j = \alpha_j + i\beta_j$ are complex numbers and K is a small number. Signals of this form appear for example in communications, economics, spectroscopy and medical imaging with MRI to name a few.



;

Figure 1: A signal with sparse complex frequency content.

1.1. Complex frequency estimation without noise; the ESPRIT-method

To begin, we first assume that the signal has no distortion from noise. Even in this case this problem is non-linear and there are various methods that work with this problem. The most common is the ESPRIT method which can be used to solve for the *complex frequencies* ζ_j of our input in (2). After retrieving the unique ζ_j values we can retrieve the c_j values with the least squares method. The ESPRIT algorithm relies on Hankel matrices, Kronecker's Theorem and Singular Value Decomposition (SVD). In this section we briefly outline the Esprit method, a more detailed explanation is given in the appendix.

Hankel matrices are matrices with a particular structure, namely that its values along its anti-diagonals are constant. i.e. a matrix H is called Hankel if $H(m,n) = H(\tilde{m},\tilde{n})$ whenever $m+n = \tilde{m}+\tilde{n}$. Thus a Hankel matrix can be formed elementwise by a vector. In the present situation, if we take discretized samples of the function $F(t)$ above; $f(j) = F\left(\frac{j-N}{N}\right)$ for $j \in 0,...,2N$, this yields a vector $f = (f_0,...,f_{2N})$ which in turn gives rise to a square Hankel matrix such that $H_f(m,n) = f_{m+n-2}, 1 \leq m,n \leq N$

$$H_f = \begin{pmatrix} f_0 & f_1 & f_2 & \cdots & f_N \\ f_1 & f_2 & & \iddots & \vdots \\ f_2 & & \iddots & & f_{2N-2} \\ \vdots & \iddots & & f_{2N-2} & f_{2N-1} \\ f_N & \cdots & f_{2N-2} & f_{2N-1} & f_{2N} \end{pmatrix}$$

Kronecker's Theorem for Hankel matrices essentially states that for a uniformly sampled function F, the above Hankel matrix has rank K if and only if f coincides at the sample points with a function that is a linear combination of K exponential functions, i.e. of the form (2). There are exceptions to this relation but they have 0 probability to arise in applications, more on this topic can be found in section 10 of [5].

A Singular Value Decomposition (SVD) is a factorization of a matrix, which can be seen as a generalization of the eigendecomposition of a positive semidefinite matrix. Given a complex MxM matrix its SVD is of the form $U\Sigma V^*$, where U, V are complex unitary matrices of dimensions MxM and V^* is the complex conjugate transpose of V. Σ is an MxM diagonal matrix with non-negative real numbers σ_i on the diagonal, known as the *singular values*.

We now have the necessary ingredients to present the basics of the ESPRIT-method. One first applies the SVD to the Hankel matrix H_f which gives the factorization

$$H_f = U\Sigma V^* \tag{3}$$

For this step it is important that the Hankel matrix has more than $K+1$ rows and columns. Applying Kronecker's Theorem we know that this Hankel Matrix will be of rank K. Thus our matrix Σ will have K non-zero singular values. For this reason the latter columns of U and V become unimportant, we will simply consider U and V to be MxK matrices. A basic observation for the ESPRIT method is that the columns of U are sums of the same exponential functions that give rise to f. A new matrix A is then formed as follows;

$$A = \Big(\big((U^-)^*\big)U^-\Big)^{-1}\big((U^-)^*\big)(U^+)$$

with U^+,U^- formed by removing the first and last row of U respectively. It turns out that the eigenvalues of A coincide with the numbers $(e^{\zeta_1/N},...,e^{\zeta_K/N})$, and this is the key observation underlying the ESPRIT method. Hence taking an eigenvalue decomposition $A = B^{-1}DB$, we can retrieve our initial values where $N\log(D_{j,j}) = \zeta_j$.

The ESPRIT method is highly accurate when working with noise free signals. That is for functions of the form (2) we can confidently retrieve $\zeta_1,...,\zeta_K$ to machine precision. If K is unknown the above tools can also be used to determine K, since it coincides with the rank of the matrix H_f.

1.2. Using ESPRIT on noisy signals

When noise is added to the input signal, the ESPRIT method can still be used but with decreasing accuracy as the noise increases. Even the issue of determining K becomes complicated, since for noisy signals H_f has full rank, but for small levels of noise the correct K can be estimated by looking for an index where the singular values drop drastically in magnitude. Assuming this has been done, we can still apply ESPRIT to

$$F(t) = \sum_{j=1}^{K} c_j e^{\zeta_j t} + e(t),\ t \in [-1, 1] \tag{4}$$

where $e(t)$ is the noise. This yields some complex frequencies $\tilde{\zeta}_1 ... \tilde{\zeta}_K$ but the relation between ζ and $\tilde{\zeta}$ is unclear. From previous experimentation it has been shown that the margin of error correlates to the scale of the noise. Some results on this can be found in [2].

The methods discussed in this paper aim to improve this method. This is mainly done through a preprocessing step, which aims to reduce noise distortion in the signal. This will give us more control of the input into the ESPRIT algorithm so that the output will be more predictable. Ideally, the preprocessing step should be designed to approximate (4) with a function of the form $\sum_{j=1}^{K} c_j e^{\hat{\zeta}_j}$, for example by minimizing the Frobenius norm (see Def 2.2)

$$\|f(t) - \sum_{j=1}^{K} c_j e^{\hat{\zeta}_j t}\|^2, \tag{5}$$

before this function is fed to the ESPRIT method, thereby retrieving the estimates $\hat{\zeta}_1, \ldots, \hat{\zeta}_K$. Unfortunately, the minimization of (5) is a highly non-linear problem. Moreover, before one could even attempt to minimize an expression of the form (5), there is the problem of estimating K. In this thesis we will propose one way of dealing with this non-linearity as well as simultaneously estimating K. More precisely, we will introduce a parameter μ which controls the trade-off between K and a weighted version of (5).

1.3. Optimization tools to reduce noise

To introduce these ideas, consider the problem

$$\min_a K\mu + \sum_{j=-N}^{N} |a(j) - f(j)|^2, \text{ subject to } a(j) = \sum_{p=1}^{K} c_p e^{\zeta_p j} \tag{6}$$

where μ is an additional parameter which penalizes the amount of complex exponentials used. Based on Kronecker's theorem, an almost equivalent problem is

$$\min_A \mu rank(A) + \|A - H_f\|^2, \text{ subject to } A \in Hankel \tag{7}$$

since $K = \mathsf{rank}A$ when A is generated by a function of the form $\sum_{p=1}^{K} c_p e^{\zeta_p j}$, as explained in Section 1.1. The key difference is that in the norm $\|A - H_f\|^2$, values of F near 0 appear more

frequently in the Hankel matrix H_f, and hence get a larger weight than values of F near the endpoints ± 1. This is an undesired side-effect of working with (7) as opposed to (6).

Both the above problems are still highly non-linear and worse yet (7) is also discontinuous. The benefit of (7) is that based on new techniques developed by M. Carlsson and C. Olsson [3, 4], the problem (7) can be "convexified" with the use of the transform $\mathcal{S}^2$, and hence approximate solutions to (7) can be found using standard convex optimization methods. In this thesis, we will in particular focus on the Alternating Directions Method of Multipliers, which is discussed in Section 4.

We now briefly discuss how to apply the $\mathcal{S}^2$ transform to this problem and leave most details to Section 2.4 and Section 3. The $\mathcal{S}^2$ transform acts on the penalty $\mu\mathsf{rank}$ to produce another penalty $\mathcal{S}^2(\mu\mathsf{rank})$ with the property that $\mathcal{S}^2(\mu\mathsf{rank})(A) + \|A - H_f\|^2$ is the convex envelope of $\mu\mathsf{rank}(A) + \|A - H_f\|^2$. Thus, considering the problem

$$\min_A \mathcal{S}^2(\mu\mathsf{rank})(A) + \|A - H_f\|^2, \text{ subject to } A \in Hankel \tag{8}$$

we obtain a convex problem closely related to (7) but which can be efficiently solved.

For readers unsure of what the convex envelope is we will give an intuitive way to visualise it. Imagine wrapping cling film around a non-convex object, say a stone. This will produce a convex object with a similar shape as the original one. The convex envelope is basically what you would get if you apply this process to the graph of a non convex function. More on this in Section 2.3.

This thesis is structured as follows; Section 2 will present some important definitions and necessary results, which includes the $\mathcal{S}^2$ transform. In Section 3 we compute $\mathcal{S}^2(\mu\mathsf{rank})(A)$. Section 4 gives an overview of ADMM, how to apply it and why it is an effective method to use with this particular problem. Section 5 proposes a method to improve upon the yield of the $\mathcal{S}^2$ transform for problems of this type. We will then give a brief review of the implementation of the algorithm and in the last section we will run tests on the ideas presented in this thesis.

2. Preliminary Results

2.1. General notation

$\mathbb{M}_n$ denotes the set of $n \times n$ complex matrices with the Frobenius norm. The entry on position (i, j) of a general matrix $X \in \mathbb{M}_n$ will be denoted $X_{(i,j)}$. $\mathcal{H}$ denotes the subset of Hankel Matrices. $\mathcal{V}$ is a finite dimensional linear vector space equipped with a scalar product. $\dot{\mathbb{R}}$ is the set of all real values excluding 0, i.e $\dot{\mathbb{R}} = \mathbb{R}\backslash\{0\}$. The singular value decomposition of a given matrix $A \in \mathbb{M}_n$ is denoted $A = U\Sigma V^*$, where $U \in \mathbb{M}_n$, $\Sigma \in \mathbb{M}_n$ and $V \in \mathbb{M}_n$. Σ is a diagonal matrix where the diagonal entries contain the singular values of A. The vector of singular values is thus denoted by σ. We denote the specific singular values as $\sigma_i(A) = \Sigma_{i,i}$ where the singular values are ordered non-increasingly. If $(u_j)_{j=1}^n$ and $(v_j)_{j=1}^n$ are the columns of U and V respectively, then we note that the SVD implies

$$A = \sum_{j=1}^{n} \sigma_j(A) u_j \langle x, v_j \rangle$$

We remind the reader that the rank of A equals the amount on non-zero singular values. In particular, if $\mathsf{rank}A = d < n$ it suffices to sum until d above.

2.2. Matrix theory

In this section we simply list a number of definitions and results from standard matrix theory which will be needed later.

Definition 2.1. *(Unitary matrix). A matrix is unitary if for a matrix* $U \in \mathbb{C}^{m\times n}$

$$UU^* = U^*U = I$$

Definition 2.2. *(Frobenius Norm). The Frobenius norm on* $\mathbb{M}_n$ *arises from the scalar product*

$$\langle X, Y\rangle = \sum_{i,j=1}^{n} X_{i,j}\overline{Y_{i,j}}$$

Proposition 2.3. *The Frobenius Norm is invariant under multiplication with unitary matrices. More precisely, for a given matrix* X *and an arbitrary unitary matrix* U

$$\|X\| = \|UX\|$$

Proof. With the following manipulations we can see that

$$\langle UX, Y\rangle = \sum_{i,j}\sum_{k} U_{i,k}X_{k,j}\overline{Y_{i,j}} = \sum_{k,j} X_{k,j}(\overline{\sum_{i}\overline{U_{i,k}}Y_{i,j}})$$

$$= \sum_{k,j} X_{k,j}(\sum_{i} U^*_{k,i}Y_{i,j}) = \langle X, U^*Y\rangle$$

where the second equality sign holds by simply changing the order of summation. Also we remember that U^* is also unitary. Thus we can see for a matrix Y that $\langle UX, Y\rangle = \langle X, U^*Y\rangle$. Applying this we see that we have equality of the norms by the following manipulations:

$$\|X\|^2 = \langle X, X\rangle = \langle U^*UX, X\rangle = \langle UX, UX\rangle = \|UX\|^2$$

■

The proof of the following result falls outside the scope of this thesis, and can be found be in [6]

Theorem 2.4. *(Von Neumann Trace Inequality for operators). Let* $X, Y \in \mathbb{M}_n$ *be arbitrary. Then*

$$\langle X, Y\rangle \leq \sum_{j} \sigma_j(X)\sigma_j(Y)$$

with equality if and only if the singular vectors can be chosen identically.

Definition 2.5. *Let* S *be a subset of* $\mathbb{R}^n$*. The characteristic function of* S *is the function*

$$\chi_S : \mathbb{R}^n \to \mathbb{R}$$

which is the function defined as

$$\chi_S(x) = \begin{cases} 1, & x \in S, \\ 0, & x \notin S \end{cases}$$

Definition 2.6. *For* $X \in \mathbb{M}_n$

$$\mathsf{rank}(X) = \sum_{j=1}^{d} \chi_{\dot{\mathbb{R}}}\big(\sigma_j(X)\big)$$

2.3. Convex analysis

Convex Analysis is a very useful field of mathematics particularly within optimization. The convexification of a problem ensures that all local minima are global minima, this becomes very advantageous with particular search-type algorithms where otherwise there is a high chance of convergence at a sub-optimal point.

In this section we discuss the *proximal operator*, *Fenchel-Legendre Transform*, *the convex envelope* and other important results which are closely related to these tools.

Definition 2.7. *A function f defined on $\mathbb{R}^n$ is convex if*

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y) \forall x, y, \alpha \in [0,1]$$

and strictly convex if

$$f(\alpha x + (1-\alpha)y) < \alpha f(x) + (1-\alpha)f(y) \forall x, y, \alpha \in [0,1]$$

Our first objective is to show that a finite convex function is continuous over it's domain. However, first we must present some technical results necessary to the proof.

Definition 2.8. *The* $\liminf$ *of a sequence a_k is defined as*

$$\liminf_{k\to\infty}\{a_k\} = \lim_{k\to\infty}\inf_{n\geq k}\{a_n\}$$

Given two sequences (a_k) and (b_k), it is easy to see that

$$\liminf_{n\geq k}\{a_n + b_n\} \geq \liminf_{n\geq k} a_k + \liminf_{n\geq k} b_k.$$

The next lemma shows that we sometimes get equality.

Lemma 2.9. *If* $\lim b_k$ *exists then*

$$\liminf_{k\to\infty}\{a_k + b_k\} = \liminf_{k\to\infty} a_k + \lim_{k\to\infty} b_k$$

and

$$\liminf_{k\to\infty}\{a_k \cdot b_k\} = \liminf_{k\to\infty} a_k \cdot \lim_{k\to\infty} b_k$$

Proof. Assume $\lim b_k = B$. Given $\epsilon > 0$ there exists k_0 such that $|b_j - B| < \epsilon$ for all $j > k_0$. In particular $B + \epsilon \geq b_j \geq B - \epsilon$. So for $k \geq k_0$ and $n \geq k$ we have

$$\inf_{n\geq k}\{a_k + b_k\} \geq \inf_{n\geq k} a_k + \inf_{n\geq k} b_k$$

From this we can clearly see that

$$\inf_{n\geq k}\{a_n + B - \epsilon\} \leq \inf_{n\geq k}\{a_n + b_n\} \leq \inf_{n\geq k}\{a_n + B + \epsilon\}$$

Since B and ϵ do not depend on n these terms can be moved outside the infimum operator.

$$\inf_{n\geq k}\{a_n\} + B - \epsilon \leq \inf_{n\geq k}\{a_n + b_n\} \leq \inf_{n\geq k}\{a_n\} + B + \epsilon$$

Then taking the limit of all terms we get

$$\liminf_{k\to\infty}\{a_n\} + B - \epsilon \leq \liminf_{k\to\infty}\{a_n + b_n\} \leq \liminf_{k\to\infty}\{a_n\} + B + \epsilon$$

Given that ϵ is arbitrary if we let $\epsilon \to 0$ then we are left with

$$\liminf_{k\to\infty}\{a_n\} + B \leq \liminf_{k\to\infty}\{a_n + b_n\} \leq \liminf_{k\to\infty}\{a_n\} + B,$$

as was to be shown. The proof for $\liminf_{n\geq k}\{a_k \cdot b_k\} = \liminf_{n\geq k} a_k \cdot \liminf_{n\geq k} b_k$ follows by a similar argument. ■

Theorem 2.10. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex and such that int(dom f) $\neq \emptyset$. Then, f is continuous over int(dom f).*

Proof. If necessary, we make a translation *wlog* so that the origin is in the interior of the domain of f. By rescaling if necessary, we may also assume that the unit box $\{x \in \mathbb{R}^n \mid \|x\|_\infty < 1\}$ is contained in dom$f$. Note also that$\|x\|_\infty = max\{|x_i|\}$. Let $v_i, i = 1, ..., 2^n$ be an enumeration of vertices of the unit box. Thus each entry v_i has entries 1 or -1. This also gives that every x with $\|x\|_\infty \leq 1$ is a convex combination of vertices v_i.
Equivalently, we can express this as

$$x = \sum_{i=1}^{2^n} \alpha_i v_i \quad \text{with } \alpha_i \geq 0 \text{ and } \sum_{i=1}^{2^n} \alpha_i = 1$$

By the convexity of f, we have

$$f(x) \leq \max f(v_i) = M$$

Similarly we want to show that f is bounded from below. *Wlog* we set $f(0) = 0$ and note that

$$0 = f(0) \leq \frac{1}{2}f(x_i) + \frac{1}{2}f(-x_i) \leq \frac{1}{2}f(x_i) + \frac{M}{2} \Rightarrow$$

$$\Rightarrow -\frac{M}{2} \leq \frac{1}{2}f(x_i) \Rightarrow -M \leq f(x_i)$$

Let x_k be a sequence whose limit is zero. Our aim is to prove that $\lim_{k\to\infty} f(x_k) = f(0)$. We introduce y_k and z_k as

$$y_k = \frac{x_k}{\|x_k\|_\infty} \quad \text{and} \quad z_k = \frac{-x_k}{\|x_k\|_\infty}$$

From this we can write 0 as a convex combination of y_k and z_k.

$$0 = \frac{1}{\|x_k\|_\infty + 1}x_k + \frac{\|x_k\|_\infty}{\|x_k\|_\infty + 1}z_k \quad \text{for all } k$$

By the convexity of f we have that

$$f(0) \leq \frac{1}{\|x_k\|_\infty + 1}f(x_k) + \frac{\|x_k\|_\infty}{\|x_k\|_\infty + 1}f(z_k) \quad \text{for all } k$$

Applying the lim inf operator to both sides of the inequality.

$$f(0) \leq \liminf_{k\to\infty} \Big\{ \frac{1}{\|x_k\|_\infty + 1} f(x_k) + \frac{\|x_k\|_\infty}{\|x_k\|_\infty + 1} f(z_k) \Big\} \tag{9}$$

Using Lemma 2.9 we set $a_k = \frac{1}{\|x_k\|_\infty + 1} f(x_k)$ and $b_k = \frac{\|x_k\|_\infty}{\|x_k\|_\infty + 1} f(z_k)$, since f is bounded above and below we know that $\lim_{k\to\infty} b_k$ exists.

$$\lim_{k\to\infty} |b_k| = \lim_{k\to\infty} \Big| \frac{\|x_k\|_\infty}{\|x_k\|_\infty + 1} f(z_k) \Big| \leq \lim_{k\to\infty} \frac{M}{\|x_k\|_\infty + 1}$$

Using the Squeeze theorem $\lim_{k\to\infty} \frac{\|x_k\|_\infty}{\|x_k\|_\infty + 1} f(z_k) = 0$. Since $\liminf_{k\to\infty} \{a_k + b_k\} = \liminf_{k\to\infty} \{a_k\} + \lim_{k\to\infty} \{b_k\}$ we can express (9) as follows

$$f(0) = \liminf_{k\to\infty} \frac{1}{\|x_k\|_\infty + 1} f(x_k)$$

From the second part of Lemma 2.9 we see that

$$\liminf_{k\to\infty} \frac{1}{\|x_k\|_\infty + 1} f(x_k) = 1 \cdot \liminf_{k\to\infty} f(x_k).$$

We now have the result

$$f(0) = \liminf_{k\to\infty} f(x_k).$$

Similarly we can write $x_k = (1 - \|x_k\|_\infty)0 + \|x_k\|_\infty y_k$ which using convexity gives

$$f(x_k) \leq (1 - \|x_k\|_\infty) f(0) + \|x_k\|_\infty f(y_k)$$

Applying lim sup we see that

$$\limsup_{k\to\infty} f(x_k) \leq \limsup_{k\to\infty} \Big((1 - \|x_k\|_\infty) f(0) + \|x_k\|_\infty f(y_k) \Big) \leq$$
$$\limsup_{k\to\infty} \Big((1 - \|x_k\|_\infty) f(0) \Big) + \limsup_{k\to\infty} \Big(\|x_k\|_\infty f(y_k) \Big) = \limsup_{k\to\infty} f(x_k)$$

where again we have used Lemma 2.9. Summing up we have

$$\lim_{k\to\infty} f(x_k) = f(0)$$

showing that f is continuous at 0. ■

We now give the definition of the Convex Envelope and the Fenchel-Legendre Transform. The Convex Envelope is a key ingredient to this paper, its main benefit is that its minima coincides with its original counterpart.

Definition 2.11. *The set of convex functions from $\mathbb{R}^n$ to $\mathbb{R}$ is denoted by $\Gamma(\mathbb{R}^n)$. The set $\Gamma(\mathbb{R}^n)$ is closed under several important operations such as addition and multiplication.*

We remark that we omit functions taking the value ∞ from $\Gamma(\mathbb{R}^n)$, which greatly simplifies many proofs.

Definition 2.12. *Let $f : \mathbb{R}^n \to \mathbb{R}$. Then*

$$\check{f} = \sup\{g \in \Gamma(\mathbb{R}^n) | g \leq f\}$$

is the convex envelope of f.

The $\mathcal{S}$ transform is essentially the sum of the Fenchel-Legendre transform of a function and a quadratic term, therefore the latter is a key result to the developments discussed in this thesis. The Fenchel-Legendre transform is a tool which is prominent in the field of Convex Analysis, which can be applied to a non-convex function and returns it's convex conjugate.

Definition 2.13. *The Fenchel conjugate, convex conjugate or Fenchel-Legendre transform of f is defined as: Given a functional $f : \mathbb{R}^n \mapsto \mathbb{R} \cup \{\infty\}$*

$$L(f)(y) = f^* := \sup_x \langle x, y\rangle - f(x)$$

Proposition 2.14. *(Fenchel Young inequality). Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be proper. Then for all $x \in \mathbb{R}^n$ and for all $u \in \mathbb{R}^n$*

$$f(x) + f^*(u) \geq \langle x, u\rangle$$

Proof. The definition of the convex conjugate yields $f^*(u) \geq \langle x, u\rangle - f(x)$ and the inequality follows. ■

The proof of the following Theorem is also beyond the scope of this thesis and is omitted from this paper. A proof of this result can be found in [11].

Theorem 2.15. *(Hahn Banach Separation Theorem). Let S_1 and S_2 be non-empty and convex subsets of $\mathcal{V}$. Furthermore, assume that S_1 and S_2 are disjoint and that S_1 has an interior point. Then there is a hyperplane that separates S_1 and S_2.*

Theorem 2.16. *(Fenchel-Moreau). If f is finite and continuous, then $f = f^{**}$ where*

$$f^{**}(x) := \sup_{x^*} \langle x^*, x\rangle - f^*(x^*)$$

Proof. By the definition of f^* we have that

$$f^*(y) = \sup_x \langle x, y\rangle - f(x) \tag{10}$$

which gives that

$$f(x) \geq \langle y, x\rangle - f^*(y), \text{ for every } y \in \mathbb{R}^n$$

Taking the supremum over y gives $f \geq f^{**}$. So we want to show that $f \leq f^{**}$. If we rearrange (10) we see that we have

$$-f^*(y) = \inf_x f(x) - \langle x, y\rangle$$

For notational purposes we set $-f^*(y) = a$. Further we get the inequality

$$-f^*(y) = a \leq f(x) - \langle x, y\rangle \text{ for all } x \Rightarrow \tag{11}$$

$$f(x) \geq a + \langle x, y\rangle \text{ for all } x$$

Thus we have

$$-f^*(y) = \sup\left\{a \,|\, f(x) \geq a + \langle x, y\rangle \,\forall x\right\}$$

Using the Hahn Banach Theorem as above, we know that for a point x_0, given that $\xi < f(x_0)$ There exists a hyper plane seperating the spaces such that

$$f(x) \geq \xi + \langle y, x - x_0\rangle = \xi - \langle y, x_0\rangle + \langle x, y\rangle \Rightarrow$$

$$f(x) - \langle x, y\rangle \geq \xi - \langle y, x_0\rangle$$

which combined with (11) gives

$$-f^*(y) \geq \xi - \langle y, x_0\rangle$$

or equivalently

$$-f^*(y) + \langle y, x_0\rangle \geq \xi.$$

We can then express this as follows

$$f^{**}(x_0) = \sup_y \langle x_0, y\rangle - f^*(y) \geq \xi.$$

Given that $\xi < f(x_0)$ we have that $\sup \xi \leq f(x_0)$. Also since $f^{**}(x_0) \geq \xi$ we get that $f^{**}(x_0) \geq \sup \xi$. The value of ξ is arbitrary so we can choose ξ such that $f^{**}(x_0) \geq f(x_0)$. This gives

$$f^{**}(x) \geq f(x) \text{ and } f^{**}(x) \leq f(x).$$

We have now proven what we set out to show, namely

$$f^{**}(x) = f(x)$$

■

Corollary 2.17. *If f is finite and continuous, then $f^{**} = \check{f}$. Where $\check{f}$ is the convex envelope of f.*

Proof. Assume some function $h = \check{f} = \sup\{g \in \Gamma(\mathbb{R}^n) | g \leq f\}$. Then $f^*(y) = \sup_x \langle x, y\rangle - f(x)$ and $h^*(y) = \sup_x \langle x, y\rangle - h(x)$ an

$$f^{**}(y) = \sup_x \langle x, y\rangle - f^*(x) \text{ and } h^{**}(y) = \sup_x \langle x, y\rangle - h^*(x)$$

Using that $\sup_x f(x) = \sup_x h(x)$ implies that $f^*(y) = h^*(y)$ and similarly $f^{**}(x) = h^{**}(x)$. Using the Fenchel Moureau Theorem $h^{**} = h$, thus $f^{**} = h = \check{f}$. ■

Definition 2.18 (The Proximal Operator)**.**

$$prox_f(x) = \arg\min_y \frac{1}{2}\|y - x\|_2^2 + f(y)$$

The proximal operator is used to make an approximation while having a trade-off parameter. For example,

$$\min_y \frac{1}{2}\|y - x\|_2^2$$

gives $y = x$ as the best approximation possible. With f present the proximal operator yields a point y near x for which $f(y)$ is lower.

Definition 2.19. *Let S be a subset of $\mathbb{R}^n$. The indicator function of S is the function*

$$\iota_S : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$$

which is the function defined as

$$\iota_S(x) = \begin{cases} 0, & x \in S, \\ +\infty, & x \notin S \end{cases}$$

Theorem 2.20. *If S is a subspace then the proximal operator of ι_S is equal to the orthogonal projection onto S.*

Proof. The proximal operator of ι_S can be expressed as

$$\begin{aligned} \mathsf{prox}_S(y) &= \arg\min_x \ \|x - y\|^2 + \iota_S(x) \\ &= \arg\min_{x \in S} \ \|x - y\|^2 \end{aligned}$$

which using basic linear algebra methods the solution is known to be the orthogonal projection onto the subspace S. ∎

2.4. $\mathcal{S}^2$ transform

The results in the previous section are easily extended to hold in any such space $\mathcal{V}$. We now introduce the $\mathcal{S}$ transform. The use of this transform is the backbone of the new results discussed in this paper.

Definition 2.21. *Given a functional f on $\mathcal{V}$ we define*

$$\mathcal{S}(f)(y) := \mathcal{L}\Big(f(\cdot) + \frac{1}{2}\|\cdot\|^2\Big)(y) - \frac{1}{2}||y||^2 = \sup_x -f(x) - \frac{1}{2}\|x - y\|^2$$

By repeating the process of the S transform we can calculate $\mathcal{S}^2 f(x)$.

$$\begin{aligned} \mathcal{S}^2(f)(x) &= -\left(\inf_y -\Big(\inf_w f(w) + \frac{1}{2}\|w - y\|^2 \Big) + \frac{1}{2}\|x - y\|^2 \right) \\ &= \sup_y \Big(\inf_w f(w) + \frac{1}{2}\|w - y\|^2 \Big) - \frac{1}{2}\|x - y\|^2 \end{aligned}$$

Proposition 2.22. *Let f be a $[0, \infty)$-valued functional on $\mathcal{V}$. Then $\mathcal{S}(f)$ takes values in $(-\infty, 0]$ and is continuous, whereas $\mathcal{S}^2(f)$ is finite valued, takes values in $[0, \infty)$ and is continuous in the interior of dom$(\mathcal{S}^2(f))$*

Proof. The changing of signs in the domain is clear to see from the definition above. It is clear to see that $\mathcal{S}(f)$ does not go to $-\infty$. We can see from the definition of $\mathcal{S}$ and $\mathcal{S}^2$ that they are the difference of a finite convex functional and a quadratic term and thus the continuity follows by Theorem 2.10 ∎

Theorem 2.23. *Let f be a finite valued continuous functional on $\mathcal{V}$. Then*

$$\mathcal{L}\Big(f(x)+\frac{1}{2}\|x-d\|^2\Big)(y)=\mathcal{S}(f)\big(y+d\big)+\frac{1}{2}\|y+d\|^2-\frac{1}{2}\|d\|^2$$

and

$$\mathcal{L}\Big(S(f)\big(y+d\big)+\frac{1}{2}\|y+d\|^2-\frac{1}{2}\|d\|^2\Big)(x)=\mathcal{S}^2(f)(x)+\frac{1}{2}\|x-d\|^2.$$

In particular, $\mathcal{S}^2(f)(x)+\frac{1}{2}\|x-d\|^2$ is the convex envelope of $f(x)+\frac{1}{2}\|x-d\|^2$.

Proof.

$$\mathcal{L}\Big(f(x)+\frac{1}{2}|x-d|^2\Big)(y)=\sup_x -f(x)-\frac{1}{2}\|x-(y+d)\|^2+\frac{1}{2}\|y+d\|^2-\frac{1}{2}\|d\|^2$$

from which the first identity follows. Similarly

$$\mathcal{L}\Big(S(f)\big(y+d\big)+\frac{1}{2}\|y+d\|^2-\frac{1}{2}\|d\|^2\Big)(x)=\sup_y -\mathcal{S}(f)\big(y+d\big)-\frac{1}{2}\|y+d-x\|^2+\frac{1}{2}\|x-d\|^2$$

$$=\mathcal{S}^2(f)(x)+\frac{1}{2}\|x-d\|^2$$

The statement about the convex envelope can be seen from Theorem 2.10 and Corollary 2.17. ■

Corollary 2.24. *For a function $f:\mathbb{R}^n\mapsto\mathbb{R}$*

$$\mathcal{S}^2(f)\leq f(x)$$

Proof. Given that $\mathcal{S}^2(f)(x)+\|x-d\|^2$ is the convex envelope of $f(x)+\|x-d\|^2$, we have that

$$\mathcal{S}^2(f)(x)+\|x-d\|^2\leq f(x)+\|x-d\|^2\Rightarrow$$

$$\mathcal{S}^2(f)(x)\leq f(x)$$

■

3. Computing $\mathcal{S}^2$ transform of $\mu\mathsf{rank}$

Recalling that in the opening section we had been tackling the problem

$$\min_X \mathcal{S}^2(\mu rank)(X)+\|X-H_f\|^2, \text{ subject to } X\in\mathcal{H}. \tag{12}$$

Using the results from Section 2 we can now calculate the $\mathcal{S}^2$ transform of the rank function.

$$\mathcal{S}(\mu\mathsf{rank})(X)=\mathcal{L}\Big(\mu\mathsf{rank}(X)+\frac{1}{2}\|X\|^2\Big)(Y)-\frac{1}{2}\|Y\|^2$$

$$=\sup_X -\mu\mathsf{rank}(X)-\frac{1}{2}\Big(\|X\|^2-2\langle X,Y\rangle+\|Y\|^2\Big),\ X,Y\in\mathbb{M}_n \tag{13}$$

Immediately an issue arises as finding the supremum of a matrix can be difficult. For this reason we will demonstrate how to decompose the problem so that we can work with vectors. First we will show that (13) can be expressed as the sum of $\mathcal{S}$ transforms of the characteristic function (defined in 2.5) on the singular values of X and then we will compute the $\mathcal{S}^2$ transform of that function.

3.1. Decomposing $\mathcal{S}(\mu\mathsf{rank})$

To begin we take an SVD of $X = U_{x}\sigma(X)V_{x}^{*}$ and using Proposition 2.3 we see that

$$\|X\| = \|U_{x}^{*}XV_{x}\| = \|U_{x}^{*}U_{x}\sigma(X)V_{x}^{*}V_{x}\| = \|\sigma(X)\|$$

Similarly for $Y = U_{y}\sigma(Y)V_{y}^{*}$, $\|Y\| = \|\sigma(Y)\|$. We can then write (13) as

$$\sup_{X} -\sum_{i=1}^{d}\chi_{\dot{\mathbb{R}}}\big(\sigma_i(X)\big) - \frac{1}{2}\Big(\|\sigma(X)\|^2 - 2\langle X, Y\rangle + \|\sigma(Y)\|^2\Big). \tag{14}$$

It is clear to see that the largest value of X will be achieved when $\langle X, Y\rangle$ is at a maximum. Thus recalling Theorem 2.4 we know the maximum value of $\langle X, Y\rangle$ is $\langle\sigma(X), \sigma(Y)\rangle$. This happens when $U_{x} = U_{y}$ and $V_{x} = V_{y}$. Given that an SVD is not a unique decomposition we can choose U_{x} and V_{x} so this happens, (14) then becomes

$$\sup_{\sigma(X)}\left(-\sum_{i=1}^{d}\chi_{\dot{\mathbb{R}}}\big(\sigma(X)\big)\right) - \frac{1}{2}\Big(\|\sigma(X)\|^2 - 2\langle\sigma(X), \sigma(Y)\rangle + \|\sigma(Y)\|^2\Big)$$

$$= \sup_{\sigma_i(X)} -\sum_{i=1}^{d}\chi_{\dot{\mathbb{R}}}\big(\sigma_i(X)\big) - \frac{1}{2}\Big(\|\sigma_i(X)\|^2 - 2\langle\sigma_i(X), \sigma_i(Y)\rangle + \|\sigma_i(Y)\|^2\Big).$$

We see that we have a sum of $\mathcal{S}$ transforms of the characteristic function

$$\mathcal{S}(\mu\mathsf{rank})(X) = \sup\sum_{i=1}^{d} -\chi_{\dot{\mathbb{R}}}(\sigma_i(X)) - \frac{1}{2}\|\sigma_i(X) - \sigma_i(Y)\|^2 = \sum_{i=1}^{d}\mathcal{S}\big(\mu\chi_{\dot{\mathbb{R}}}\big)\big(\sigma_i(X)\big) \tag{15}$$

The whole argument can now be iterated to show that

$$\mathcal{S}^2(\mu\mathsf{rank})(X) = \sum_{i=1}^{d}\mathcal{S}^2\big(\mu\chi_{\dot{\mathbb{R}}}\big)\big(\sigma_i(X)\big), \tag{16}$$

we omit the details. Naturally we will now compute the $\mathcal{S}$ transform of the characteristic functional and then use (16) to easily compute $\mathcal{S}^2(\mu\mathsf{rank})(X)$.

3.2. $\mathcal{S}^2$ transform of the characteristic functional

For notational purposes we set $g(x) = \mu\chi_{\dot{\mathbb{R}}}(x)$. Taking the $\mathcal{S}$ transform of g we get

$$\mathcal{S}(g)(y) = \sup_{x} -\mu\chi_{\dot{\mathbb{R}}}(x) - \frac{1}{2}\big(x - y\big)^2$$

It is clear to see that the maximum is found at either $x = y$ or $x = 0$. This gives

$$\mathcal{S}(g)(y) = \max\Big\{\frac{-y^2}{2}, -\mu\Big\} = -\min\Big\{\frac{y^2}{2}, \mu\Big\}. \tag{17}$$

Repeating this process we compute $\mathcal{S}^2(g)$.

$$\mathcal{S}^2(g)(x) = \sup_{y} -\Big(-\min\Big\{\frac{y^2}{2}, \mu\Big\}\Big) - \frac{1}{2}\big(x - y\big)^2$$

$$= \sup_y \left(\min\left\{ \frac{y^2}{2}, \mu \right\} \right) - \frac{1}{2}(x-y)^2$$

We split our analysis according to whether $|x| \geq \sqrt{2\mu}$ or not and seek to find the best maximum in each case. Defining the function

$$f(y) = \left(\min\left\{ \frac{y^2}{2}, \mu \right\} \right) - \frac{1}{2}(x-y)^2,$$

and noting that for $|y| \geq \sqrt{2\mu}$, $\min\left\{\frac{y^2}{2}, \mu\right\} = \mu$, and for $|y| \leq \sqrt{2\mu}$, $\min\left\{\frac{y^2}{2}, \mu\right\} = \frac{y^2}{2}$. We only focus on $x \geq 0$ for values of $y \geq 0$ as it is easy to show that f is symmetric about the y axis.

$$\text{Case } ① : x \geq \sqrt{2\mu}.$$

For $y \geq \sqrt{2\mu}$ the maximum can be attained at $y = x$ and returns $\mathcal{S}^2(g)(x) = \mu$. For $0 \leq y \leq \sqrt{2\mu}$ given that f is an increasing linear function we know that it's maximum lies at it's most right endpoint, i.e. $y = \sqrt{2\mu}$, which gives $\mathcal{S}^2(g)(x) = \frac{-x^2}{2} + x\sqrt{2\mu}$. Clearly $\mu \geq \frac{-x^2}{2} + x\sqrt{2\mu}$, since $x \geq \sqrt{2\mu}$.

$$\text{Case } ② : 0 \leq x \leq \sqrt{2\mu}.$$

Again for $0 \leq y \leq \sqrt{2\mu}$ we have that the maximum lies at it's most right endpoint, giving $\mathcal{S}^2(g)(x) = \frac{-x^2}{2} + x\sqrt{2\mu}$. For $y \geq \sqrt{2\mu}$ the maximum lies as close to x as possible, thus the maximum is at the left endpoint, i.e $y = \sqrt{2\mu}$. Plugging this in gives $\mathcal{S}^2(g)(x) = \frac{-x^2}{2} + x\sqrt{2\mu}$, where the maximums are the same.

We see that the $\mathcal{S}^2$ transform of the characteristic functional can be expressed as follows

$$\mathcal{S}^2(g)(x) = \mu - \frac{1}{2}\Big(|x| - \sqrt{2\mu}\Big)^2 \chi_{[-\sqrt{2\mu}, \sqrt{2\mu}]}(x) = \mu - \left(\max\left\{ \sqrt{\mu} - \frac{|x|}{\sqrt{2}}, 0 \right\} \right)^2 \tag{18}$$

Now using (17) and (18) alongside (15) and (16), we see that for $\mathsf{rank}(X) = \sum_{j=1}^{d} \chi_{\dot{\mathbb{R}}}(\sigma_j(X))$ we get

$$\mathcal{S}(\mu\mathsf{rank})(Y) = \sum_{j=1}^{d} S\big(\chi_{\dot{\mathbb{R}}}\big)\big(\sigma_j(X)\big) = \sum_{j=1}^{d} \max\left\{ \frac{-\sigma_j(Y)^2}{2}, -\mu \right\} \tag{19}$$

and

$$\mathcal{S}^2(\mu\mathsf{rank})(X) = \sum_{j=1}^{d} \mu - \left(\max\left\{ \sqrt{\mu} - \frac{\sigma_j(X)}{\sqrt{2}}, 0 \right\} \right)^2 \tag{20}$$

3.3. Applying $\mathcal{S}^2$ transform with a Hankel structure

Returning to the problem

$$\arg\min_{X \in \mathcal{H}} \ \mu\mathsf{rank}(X) + \frac{1}{2}\|X - Y\|^2 \tag{21}$$

which we then tweaked to be the convex problem

$$\arg\min_{X \in \mathcal{H}} \ \mathcal{S}^2(\mu\mathsf{rank})(X) + \frac{1}{2}\|X - Y\|^2 \tag{22}$$

The solutions for (21) and (22) are not always the same. However we can be sure their solutions are the same whenever

$$\mathcal{S}^2(f)(x^*) = f(x^*)$$

where x^* is a solution to (21), because we always have that $f(x) \geq \mathcal{S}^2(f)(x)$ by Corollary 2.24. Finding solutions for (22) is a smoother option as we can implement convex optimization routines which are highly effective and generally easier to work with. However, as currently stated (22) has no closed form solution due to the subspace constraint, and has to be solved by an iterative method. As presented in [3] we can work around this subspace restriction by making a simple yet significant change. We reformulate the problem as

$$\underset{X=Y}{\arg\min}\ \mathcal{S}^2(\mu\mathsf{rank})(X) + \frac{1}{2}\|X - Y\|^2 + \iota_{\mathcal{H}}(Y) \tag{23}$$

where $\iota_{\mathcal{H}}$ is the indicator function discussed in Section 2.19. Here X is encouraged to be of low rank without the restrictions of the subspace, yet the Hankel structure is enforced on Y, but upon convergence $X = Y$, so solving this will result in a low rank Hankel matrix. (23) can be solved using the convex optimization algorithm Alternating Direction of Multipliers Method (ADMM).

The next section will give a brief review of ADMM before explaining why and how it is applicable here.

4. Alternating Direction Method of Multipliers

ADMM is a method for solving convex optimization problems by dissecting them into smaller pieces which are easier to work with. It was developed in the 1970s and since then has been a place-holder method in convex optimization for its widespread applications.

ADMM is renowned for being very versatile, and with very little conditions in place the method guarantees convergence. For our application, more optimal solutions could exist, however to keep the level of abstraction high in this paper, ADMM serves as a perfect benchmark to show that a solution for our problem can be achieved.

We will now give a quick overview of how ADMM is set up in the general case and then apply it to our calculations from the previous section.

4.1. An overview of ADMM

For two convex functions g_1 and g_2 we want to minimize

$$g_1(x) + g_2(y), \text{ subject to } Ax + By = C$$

Here we have two sets of variables where there objective is separable. Using this we can form the Lagrangian

$$L_\rho(x, y, \Lambda) = g_1(x) + g_2(y) + \langle Ax + By - C, \Lambda\rangle + \frac{\rho}{2}\|Ax + By - C\|_2^2$$

To follow the algorithm we want to first minimize our Lagrangian w.r.t x, for y, Λ fixed. Then using our new x^{k+1} we want to minimize our Lagrangian w.r.t y for x, Λ fixed. After which we compute the dual step to find Λ^{k+1}. This can be expressed as follows:

$$x^{k+1} = \arg\min_x L_\rho(x, y^k, \Lambda^k)$$

$$y^{k+1} = \arg\min_y L_\rho(x^k, y, \Lambda^k)$$

$$\Lambda^{k+1} = \Lambda^k + \rho(Ax^{k+1} + By^{k+1} - C)$$

For more information about ADMM see [7].

4.2. Setting up ADMM

Returning to section 3 we will now set up our problem to be solved using ADMM. Setting $g_1 = \mathcal{S}^2(\mu\mathsf{rank})(X) + \frac{1}{2}\|X - H_f\|^2$, $g_2 = \iota_{\mathcal{H}}(Y)$ we get the Lagrangian

$$L_\rho(X, Y, \Lambda) = \mathcal{S}^2(\mu\mathsf{rank})(X) + \frac{1}{2}\|X - \mathcal{H}_f\|^2 + \iota_{\mathcal{H}}(Y) + \langle X - Y, \Lambda\rangle + \frac{\rho}{2}\|X - Y\|^2$$

4.2.1. X update set-up

On this update step we can ignore terms independent of X as we are searching for X which minimizes its respective step, and constants won't impact this value. The X update step becomes

$$X^{k+1} = \arg\min_X \ \mathcal{S}^2(\mu\mathsf{rank})(X) + \frac{1}{2}\|X - H_f\|^2 + \iota_{\mathcal{H}}(Y^k) + \langle X - Y, \Lambda^k\rangle + \frac{\rho}{2}\|X - Y^k\|^2$$

$$= \arg\min_X \ \mathcal{S}^2(\mu\mathsf{rank})(X) + \frac{1}{2}\|X\|^2 - \langle X, H_f\rangle + \frac{1}{2}\|H_f\|^2 + \langle X, \Lambda^k\rangle + \frac{\rho}{2}\|X\|^2 - \frac{\rho}{2}\langle X, Y^k\rangle$$

$$= \arg\min_X \ \mathcal{S}^2(\mu\mathsf{rank})(X) + \frac{1+\rho}{2}\|X - D_1\|^2 \tag{24}$$

where $D_1 = \frac{H_f - \Lambda^k + \rho Y^k}{1+\rho}$. The X update step can be seen as the proximal operator of $\frac{\mathcal{S}^2(\mu\mathsf{rank})(X)}{1+\rho}$

$$\mathsf{prox}_{\frac{\mathcal{S}^2(\mu\mathsf{rank})(X)}{1+\rho}}(D_1) = \arg\min_X \ \mathcal{S}^2(\mu\mathsf{rank})(X) + \frac{1+\rho}{2}\|X - D_1\|^2$$

4.2.2. Y update set-up

Applying the same logic as the X update step, the Y update step becomes:

$$Y^{k+1} = \arg\min_Y \ \iota_{\mathcal{H}}(Y) - \langle Y, \Lambda^k\rangle + \frac{\rho}{2}\|X^{k+1} - Y\|^2$$

$$= \arg\min_Y \ \iota_{\mathcal{H}}(Y) - \rho\langle Y, X^{k+1} + \frac{\Lambda^k}{\rho}\rangle + \frac{\rho}{2}\|Y\|^2$$

$$= \arg\min_Y \ \iota_{\mathcal{H}}(Y) + \frac{\rho}{2}\|Y - D_2\|^2 \tag{25}$$

where $D_2 = X^{k+1} + \frac{\Lambda^k}{\rho}$. The third equality holds as adding constants will also not affect the value of Y. The Y update step can be seen as the proximal operator of $\iota_{\mathcal{H}}(Y)$

$$\mathsf{prox}_{\iota_{\mathcal{H}}(Y)}(D_2) = \arg\min_Y \iota_{\mathcal{H}}(Y) + \frac{\rho}{2}\|Y - D_2\|^2$$

4.3. Computing ADMM for the $\mathcal{S}^2$ transform

4.3.1. Computing X update

Applying (20) and (24) to compute the X update step we get the following

$$\arg\min_X \sum_{j=1}\mu - \left(\max\left\{\sqrt{\mu}-\frac{\sigma_j(X)}{\sqrt{2}},0\right\}\right)^2 + \frac{1+\rho}{2}\|X-D_1\|^2$$

$$= \arg\min_X \sum_{j=1}\mu - \left(\max\left\{\sqrt{\mu}-\frac{\sigma_j(X)}{\sqrt{2}},0\right\}\right)^2 + \frac{1+\rho}{2}\left(\|X\|^2 - \langle X, D_1\rangle + \|D_1\|^2\right)$$

Using a similar argument to Section 3.1 we can simplify this expression to

$$\arg\min_{\sigma(X)} \sum_{j=1}\mu - \left(\max\left\{\sqrt{\mu}-\frac{\sigma_j(X)}{\sqrt{2}},0\right\}\right)^2 + \frac{1+\rho}{2}\left(\|\sigma(X)\|^2 - \langle \sigma(X), \sigma(D_1)\rangle + \|\sigma(D_1)\|^2\right)$$

$$= \arg\min_{\sigma_j(X)} \sum_{j=1}\mu - \left(\max\left\{\sqrt{\mu}-\frac{\sigma_j(X)}{\sqrt{2}},0\right\}\right)^2 + \sum_{j=1}\frac{1+\rho}{2}\left(\|\sigma_j(X)\|^2 - \langle \sigma_j(X), \sigma_j(D_1)\rangle + \|\sigma_j(D_1)\|^2\right)$$

$$= \sum_{j=1}\arg\min_{\sigma_j(X)} \mu - \left(\max\left\{\sqrt{\mu}-\frac{\sigma_j(X)}{\sqrt{2}},0\right\}\right)^2 + \frac{1+\rho}{2}\left(|\sigma_j(X)-\sigma_j(D_1)|^2\right)$$

Which in the last expression we are working with single values and thus it is quite easy to solve at each index of summation. For purely aesthetical reasons we use the notation $\sigma_j(X) = t$, $\sigma_j(D_1) = d$ so our problem looks as follows:

$$\arg\min_t \mu - \left(\max\left\{\sqrt{\mu}-\frac{|t|}{\sqrt{2}},0\right\}\right)^2 + \frac{1+\rho}{2}\left(|t-d|^2\right) \tag{26}$$

We define the function

$$f(t) = \mu - \left(\max\left\{\sqrt{\mu}-\frac{|t|}{\sqrt{2}},0\right\}\right)^2 + \frac{1+\rho}{2}\left(|t-d|^2\right). \tag{27}$$

We know extrema exist given that f is continuous on $[0,\infty)$ and $\lim_{t\to\infty} = \infty$. It is clear to see that no singular points exist so we only need to compare critical points and endpoints. To find t which minimizes this expression we have to handle the max function. This can be done by considering the cases depending on the given fixed value of d. We only need to focus on values for $d \geq 0$ where $t \geq 0$ as it is easily verified that the expression is symmetric about the y axis.

We denote the sets $S = [0, \sqrt{2\mu})$ and $S' = [\sqrt{2\mu}, \infty)$. There are two cases for the values of d; Case ① where $d \in S$, and Case ② where $d \in S'$. We split (26) into two expressions (a) and (b) where

$$(a): \quad \arg\min_t \mu - \left(\sqrt{\mu}-\frac{t}{\sqrt{2}}\right)^2 + \frac{1+\rho}{2}\left(t-d\right)^2,\ t \in S$$

$$(b): \quad \arg\min_t \ \mu + \frac{1+\rho}{2}\Big(t-d\Big)^2,\ t \in S'$$

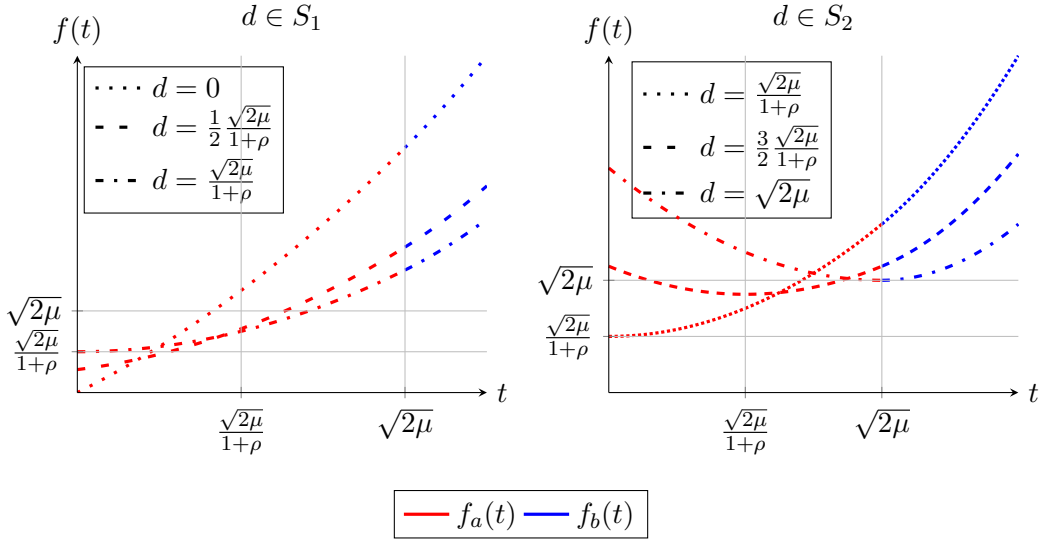
and want to explore which of (a) and (b) contains a more optimal t for each case. We define $f_a(t) = \mu - \big(\sqrt{\mu} - \frac{t}{\sqrt{2}}\big)^2 + \frac{1+\rho}{2}\big(t-d\big)^2$, where $f_a'(t) = \sqrt{2\mu} - t + (1+\rho)(t-d)$ and $f_a''(t) = \rho$. Similarly, $f_b(t) = \mu + \frac{1+\rho}{2}\big(t-d\big)^2$, where $f_b'(t) = (1+\rho)(t-d)$ and $f_b''(t) = 1+\rho$.

Case ①: $d \in S$.

For $t \in S$ we set $f_a'(t) = 0$ and see that a minimum occurs at $t = \frac{(1+\rho)}{\rho}(d) - \frac{\sqrt{2\mu}}{\rho}$. Since we are only looking for values of $t \geq 0$, this value can only be achieved for $d \in (\frac{\sqrt{2\mu}}{1+\rho}, \sqrt{2\mu})$. For this reason we split S into two subsets, $S_1 = [0, \frac{\sqrt{2\mu}}{1+\rho})$ and $S_2 = [\frac{\sqrt{2\mu}}{1+\rho}, \sqrt{2\mu})$, and want to find the optimum minimum of $d \in S_2$.

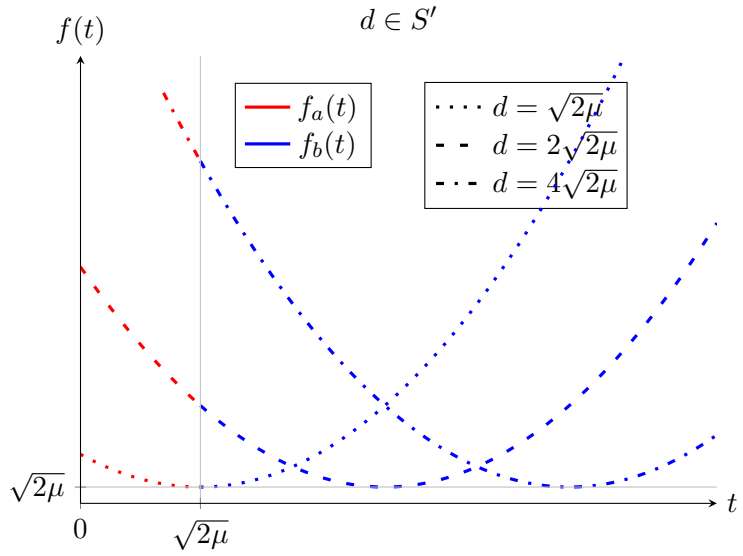
In the interval where $d \in S_1$ it is clear to see the minimum of f_a is at its left endpoint where $t = 0$, given that is an increasing function. For $t \in S'$ we see that $f_b'(t) = (1+\rho)(t-d)$ is always positive as $t > d$. f_b is an increasing function, so its minimum value is also at its left end point. This value is achieved when $t = \sqrt{2\mu}$.

Comparing these values we see that for $d \in S_1$, $\min f_a \leq \min f_b$ and $d \in S_2$, $\min f_a \leq \min f_b$. Hence, for $d \in S_1$, $t = 0$ and for $d \in S_2$, $t = \frac{(1+\rho)}{\rho}(d) - \frac{\sqrt{2\mu}}{\rho}$.



Case ②: $d \in S'$.

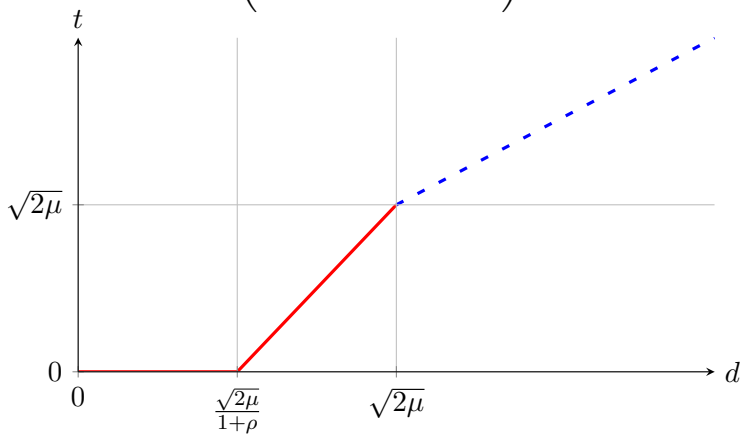
Here $f_a'(t)$ is always negative as $d > t$, from this we know that $f_a(t)$ is non-increasing and attains its minimum value at its most right end-point $t = \sqrt{2\mu}$. The minimum value of f_b is attained at the critical point $t = d$. Comparing these values we see that $\min f_a = \mu + (1+\rho)(\sqrt{2\mu} - d)^2$ and $\min f_b = \mu$, where clearly the latter is smaller.



To recap the solution for (26) is

$$\arg\min_t \; f(t) = \begin{cases} 0, & 0 \le |d| \le \frac{\sqrt{2\mu}}{1+\rho}, \\ \frac{d(1+\rho)-\sqrt{2\mu}}{\rho}, & \frac{\sqrt{2\mu}}{1+\rho} < |d| \le \sqrt{2\mu}, \\ d, & \sqrt{2\mu} < |d| < \infty. \end{cases} \tag{28}$$

$$\arg\min_t \; \mu - \left(\max\left\{ \sqrt{\mu} - \tfrac{t}{\sqrt{2}}, 0 \right\} \right)^2 + \tfrac{1+\rho}{2}(t-d)^2$$



From this plot of t against d we can see that the X update step removes small singular values d, reducing the rank of the matrix in question. The user has control of this parameter by means of ρ. Relatively small singular values are averaged out, which smooths out the range of singular values. The X update step seeks to best fit the data while encouraging a low-rank matrix.

4.3.2. Computing Y update

As mentioned at the end of Section 4.2.2 the Y update step can be expressed as the proximal operator of the indicator function. Recalling Theorem 2.20 we see that we need to

work out the orthogonal projection of D_2 onto $\mathcal{H}$. Projecting onto a subspace where the basis is over the set of Hankel matrices requires a few steps, but first we will give a brief review of how to get the orthogonal projection for a n dimensional vector onto a standard subspace.

Let $\mathcal{V}$ be a subspace of R^n. Then each $x \in R^n$ can be uniquely represented in the form $x = \hat{x} + y$, where $\hat{x} \in \mathcal{V}$ and $y \in \mathcal{V}^{\perp}$. If $\{e_1, ..., e_n\}$ is an orthonormal basis of $\mathcal{V}$ then $\hat{x} = \sum_{j=1}^{n} \langle x, e_j \rangle e_j$ and the vector $\hat{x}$ is the orthogonal projection of x onto $\mathcal{V}$. We can denote this projection as

$$P_{\mathcal{V}}(x) = \sum_{j=1}^{n} \langle x, e_j \rangle e_j.$$

However in this instance we want our basis to be composed of Hankel matrices so we have to set up our space accordingly. Let the square $N+1$ matrices $F_0, F_1, ..., F_{2N} \in \mathcal{H}$ be defined, where

$$F_k(i,j) = \begin{cases} 1, & i+j = k+2, \\ 0, & else. \end{cases}$$

Note that it forms a basis for $\mathcal{H}$ since given H_a, $a = (a_0, a_1, ..., a_{2N}) \in \mathbb{R}^{2N+1}$, we have

$$\begin{pmatrix} a_0 & a_1 & \dots & a_N \\ a_1 & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & a_{2N-1} \\ a_N & \dots & a_{2N-1} & a_{2N} \end{pmatrix} = \underbrace{\begin{pmatrix} a_0 & 0 & \cdots & 0 \\ 0 & 0 & & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}}_{a_0 F_0} + \underbrace{\begin{pmatrix} 0 & a_1 & \cdots & 0 \\ a_1 & 0 & & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}}_{a_1 F_1} + \cdots = \sum_{k=0}^{2N} a_k F_k$$

An orthonormal basis for $\mathcal{H}$ is $\left(\frac{F_0}{\|F_0\|}, \frac{F_1}{\|F_1\|}, ..., \frac{F_{2n}}{\|F_{2n}\|} \right)$. We denote our orthonormal basis as $E_j = \frac{F_j}{\|F_j\|} = F_j \times \frac{1}{\sqrt{t_j}}$, where $t_j = N + 1 - |j - N|$ to handle the triangle like structure of the Hankel matrices. So then our projection of D_2 onto our subspace $\mathcal{H}$ is given by

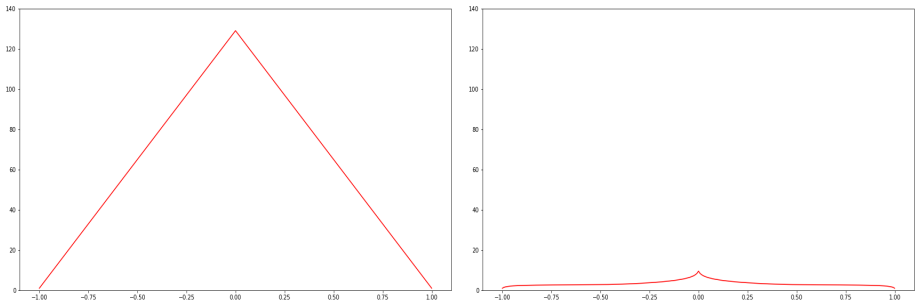
$$P_{\mathcal{H}}(D_2) = \sum_{j=0}^{2N} \langle D_2, E_j \rangle E_j = \sum_{j=0}^{2N} \langle D_2, F_j \rangle \frac{F_j}{t_j} = \mathsf{prox}_{\iota_{\mathcal{H}}}(Y). \tag{29}$$

The Y update step ensures the matrix has a Hankel structure. For our particular problem the X and Y update step compliment each other perfectly as X will eventually converge to Y resulting in a low rank Hankel matrix.

5. Improvements working in a weighted Hilbert space

As mentioned at the beginning of the previous section, ADMM is one of many applicable iterative methods that can be used to solve (22). Moreover, there is a whole area specialised in the optimization of ADMM in specific situations. The results so far in this paper suggest that the $\mathcal{S}$ transform at a base level will improve upon previously in place methods, however the yield of the $\mathcal{S}^2$ transform itself can be improved upon.

Figure 2: left: the weight for (22). right: W as per (31)



For the quadratic term in (21) we are trying to find a Hankel matrix $X = H_g$ that best fits the matrix H_f formed by samples of our input signal. This corresponds to a weighted misfit term of the form

$$\|H_f - H_g\| = \sum_{j=1}^{2N+1} (N + 1 - |j - N|)|g_j - f_j|^2.$$

It is clear to see from the left image of Figure 2 above that the triangle-like structure of the Hankel matrix amplifies the norm, in the sampling for values closer to the central anti diagonals. Hence measurements on the edges of the interval make a limited impact. To remedy this problem we will introduce some theoretical structure on how the $\mathcal{S}^2$ transform can be computed in a weighted Hilbert space and then provide an example of a weight which will significantly reduce this problem.

We notate the weighted Hilbert Space as $\mathbb{M}_n^W$ and introduce the norm

$$\|X\|_W^2 = \sum_{i,j} w_{i,j}|x_{i,j}|^2,$$

where $w_{i,j}$ are the entries of $W \in M_{n\times n}$ with strictly positive entries. To this point we have been dealing with the case $W = 1$, whereby each component of W is equal to one. The $\mathcal{S}$ transform in a weighted space is denoted by $\mathcal{S}_{\mathbb{M}_n^W}(f)(y) = \mathcal{S}_W(f)(y)$ and similarly $\mathcal{S}^2_{\mathbb{M}_n^W}(f)(x) = \mathcal{S}^2_W(f)(x)$. To continue we want to compute $\mathcal{S}_{\mathbb{M}_n^W}(\mathsf{rank})$, where W must be a direct tensor, i.e. of the form

$$w_{i,j} = u_i v_j$$

where u, v are n dimensional vectors. This restriction is necessary as left-and right multiplication of diagonal matrices by invertible matrices does not change the rank of the diagonal matrix. Introducing some new notation we let $\sqrt{W}$ be the pointwise square root of W and define the mapping $\mathcal{J} : \mathbb{M}_n^W \mapsto \mathbb{M}_n$ by

$$(\mathcal{J}(X))_{i,j} = \sqrt{w_{i,j}}x_{i,j}$$

and note that this linear operator is unitary, i.e isometric and bijective. We can see this as

$$\|\mathcal{J}X\|^2_{\mathbb{M}_n} = \sum_{i,j} |\sqrt{w_{i,j}}x_{i,j}|^2 = \|X\|^2_W.$$

Further, we define the set of Hankel matrices in the weighted space by $\mathcal{H}^W$, where the basis elements are of the form $\frac{F_j}{\|F_j\|_W}$.

5.1. Computing the $\mathcal{S}^2$ transform in weighted spaces

We now wish to calculate $\mathcal{S}^2_{\mathbb{M}_n^W}(\mu\mathsf{rank})(X)$. We begin by calculating the $\mathcal{S}$ transform of the rank function in a weighted space.

$$\mathcal{S}_{\mathbb{M}_n^W}(\mu\mathsf{rank})(Y) = \sup_{X \in M_n^W} -\mu\mathsf{rank}(\mathcal{J}^{-1}(\mathcal{J}(X))) - \frac{1}{2}\|\mathcal{J}(X-Y)\|^2$$

$$= \sup_{Z \in M_n} -\mu\mathsf{rank}(\mathcal{J}^{-1}(Z)) - \frac{1}{2}\|Z - \mathcal{J}(Y)\|^2$$

$$= \mathcal{S}(\mu\mathsf{rank} \circ \mathcal{J}^{-1})(\mathcal{J}Y) = \mathcal{S}(\mu\mathsf{rank})(\mathcal{J}Y)$$

where the last equality holds since W is a direct tensor and hence does not alter the rank. By repeating this process we get that

$$\mathcal{S}^2_{\mathbb{M}_n^W}(\mu\mathsf{rank})(X) = \mathcal{S}^2(\mu\mathsf{rank})(\mathcal{J}(X)) = \sum_{j=1}^{n} 1 - \Big(\max\{1 - \frac{\sigma_j(\mathcal{J}X)}{\sqrt{2}}, 0\})^2.$$

5.2. Proximal operators in weighted spaces

As mentioned at the end of section 4.2.1 and 4.2.2 the X and Y update steps in ADMM can be expressed as $\mathsf{prox}_{\frac{\mathcal{S}^2_W(\mu\mathsf{rank})}{1+\rho}}$ and $\mathsf{prox}_{\iota_{\mathcal{H}^W(Y)}}$ respectively. The generalization of these proximal operators in the weighted space is given by

$$\mathsf{prox}_{\frac{\mathcal{S}^2_{\mathbb{M}_n^W}(\mu\mathsf{rank})}{1+\rho}}(D_1) = \operatorname*{arg\,min}_{X \in \mathbb{M}_n^W} \ \mathcal{S}^2_{M_n^W}(\mu\mathsf{rank})(X) + \frac{1+\rho}{2}\|X - D_1\|^2_W$$

$$= \operatorname*{arg\,min}_{X \in \mathbb{M}_n^W} \ \mathcal{S}^2(\mu\mathsf{rank} \circ \mathcal{H}^{-1})(\mathcal{J}(X)) + \frac{1+\rho}{2}\|\mathcal{J}(X - D_1)\|^2_{\mathbb{M}_n}$$

$$= \operatorname*{arg\,min}_{Z \in \mathbb{M}_n} \ \mathcal{S}^2(\mu\mathsf{rank})(Z) + \frac{1+\rho}{2}\|Z - \mathcal{J}(D_1)\|^2_{\mathbb{M}_n}$$

$$= \mathcal{J}^{-1}\mathsf{prox}_{\frac{\mathcal{S}^2(\mu\mathsf{rank})}{1+\rho}}(\mathcal{J}(D_1)).$$

The orthogonal projection of the characteristic function in the weighted Hilbert space can be expressed as

$$\mathsf{prox}_{\iota_{\mathcal{H}^W(Y)}}(D_2) = \sum_{j=0}^{2N} \langle D_2, F_j\rangle_W \frac{F_j}{\|F_j\|_W} = \sum_{j=0}^{2N} \langle D_2, F_j\rangle \frac{F_j}{t_j} \tag{30}$$

where t_j generalizes to $t_j = \sum_{i=1}^{J} w_{(J+1-i),(i)}$, where $J = N - |j - N|$

5.3. A concrete weighted space

As suggested in [3] a weighted space which yields a more uniform weight for the quadratic term of $X = H_g, H_f \in \mathcal{H}^W$ is given by

$$\|H_g - H_f\|_W^2 = \sum_{i,j=1}^{N} w_{i,j}|x_{i,j} - f_{i,j}|^2 \tag{31}$$

where W is composed by $w_{i+1,j+1} = u_i u_j$ for $u_i = (\frac{N}{2}+1-|i-\frac{N}{2}|)^{\frac{-1}{2}}$, $i = 0, ..., N$. From Figure 2 we see that the norm $\|H_g - H_f\|_W$ yields a norm much closer to the standard un-weighted l^2 norm.

This concludes the theoretical details of this thesis, to continue we will give an overview of how the whole algorithm is set up before discussing test results.

6. Algorithm Recap in the weighted case

We now present the algorithm in the weighted case. Create a vector $f = (f_0, ..., f_{2N})$ from equidistant samples of input signal $F(x)$. Use this vector to create a $N+1$ square Hankel matrix via the formula $f_{m+n-2} = H(m, n)$.

Set up **ADMM** for the Lagrangian

$$L_\rho(X, Y, \Lambda) = \mathcal{S}_W^2(\mu\mathsf{rank})(X) + \frac{1}{2}\|X - \mathcal{H}_f\|_W^2 + \iota_{\mathcal{H}^W}(Y) + \langle X - Y, \Lambda\rangle_W + \frac{\rho}{2}\|X - Y\|_W^2,$$

with initial values $(X^0, Y^0, \Lambda^0) = (0, 0, 0)$ so that the update steps are achieved by

$$\begin{aligned}
X^{k+1} &= \underset{X\in\mathbb{M}_n^W}{\arg\min}\ \mathcal{S}_W^2(\mu\mathsf{rank})(X) + \frac{1+\rho}{2}\|X - D_1\|_W^2,\\
Y^{k+1} &= \underset{Y\in\mathbb{M}_n^W}{\arg\min}\ \iota_{\mathcal{H}^W}(Y) + \frac{\rho}{2}\|Y - D_2\|_W^2,\\
\Lambda^{k+1} &= \Lambda^k + \rho(X^{k+1} - Y^{k+1}),
\end{aligned}$$

where $D_1 = \frac{H_f - \Lambda^k + \rho Y^k}{1+\rho}$, and $D_2 = X^{k+1} + \frac{\Lambda^k}{\rho}$.

For the X update step we take an SVD of $Z = \mathcal{J}(D_1) = U_Z \Sigma_Z V_Z^*$ and set $X = U_Z\, \Sigma_T\, V_Z^*$, where Σ_T is the diagonal matrix formed by applying (28) to each singular value of Σ_Z, after which we perform the mapping $\mathcal{J}^{-1}$ on the returned matrix. To get Y^{k+1} we find the orthogonal projection of $\mathcal{J}(D_2)$ onto $\mathcal{H}$ using (30). The Λ update stage is clear from its definition above. Repeat ADMM until X converges and $X = Y$.

Then run the **ESPRIT** Algorithm on $X \in \mathcal{H}$, essentially the steps are as follows:

· Compute an SVD of $X = U\Sigma V^*$.
· Take U to be a $N + 1 \times \hat{K}$ matrix, where $\hat{K}$ is the number of significant singular values of Σ, i.e. where the magnitude between singular values is within some user-defined threshold.
· Form U^+ and U^- by deleting the first and last rows from U, respectively.
· Form $A = \Big(\big((U^-)^*\big)U^-\Big)^{-1}\big((U^-)^*\big)(U^+)$.
· Take an eigenvalue decomposition of $A = B^{-1}diag(\lambda_1, ..., \lambda_{\hat{K}})B$.
· Recover $\hat{\zeta_j} = \ln(\lambda_j) = N \cdot \ln(e^{\hat{\zeta_j}/N})$.

The next step is to recover $\hat{c} = (\hat{c}_1, .., \hat{c}_K)$. We can do this by performing a **least squares approximation** of the matrix formed from samples of $\hat{\zeta}$ against the vector $\hat{f}$, where $\hat{f}$ is the vector formed from average values of the anti-diagonals of X, e.g $\hat{f}_2 = \frac{X(3,1)+X(2,2)+X(3,1)}{3}$.

We seek to solve for $\hat{c}$ in the equation $A\hat{c} = \hat{f}$, where $A \in \mathbb{C}^{2N+1 \times K}$ is a matrix formed from taking $2N+1$ equidistant samples over the same domain as f with our new vector $\hat{\zeta}$. The entries of A will be of the form

$$A = \begin{pmatrix} e^{\hat{\zeta}_1 x_0} & e^{\hat{\zeta}_2 x_0} & \dots & e^{\hat{\zeta}_{\hat{K}} x_0} \\ e^{\hat{\zeta}_1 x_1} & \dots & \dots & e^{\hat{\zeta}_{\hat{K}} x_1} \\ \vdots & \dots & \dots & \vdots \\ e^{\hat{\zeta}_1 x_{2N}} & e^{\hat{\zeta}_2 x_{2N}} & \dots & e^{\hat{\zeta}_{\hat{K}} x_{2N}} \end{pmatrix},$$

where $x = x_0, ..., x_{2N}$ is a vector of equidistant points of the domain of f. This process will seek a vector $\hat{c}$ that minimizes the norm $\|A\hat{c} - \hat{f}\|^2$. Putting it all together we now have

$$\hat{F}(x) = \sum_{j=1}^{\hat{K}} \hat{c}_j e^{\hat{\xi}_j x}.$$

6.1. Additional parameters

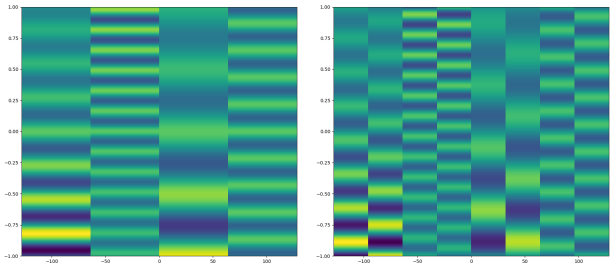
μ acts as a penalty parameter in our algorithm, encouraging a low rank matrix. We can see from the X update step of ADMM that singular values below the threshold $\sqrt{2\mu}$ will eventually converge to 0. If we set this threshold as the midpoint between the $\hat{K}th$ and $(\hat{K}+1)th$ singular values of H_f we can increase the likelihood that ADMM returns a matrix of desired rank. For this reason it is encouraged to use $\mu = \frac{1}{2}\left(\frac{\sigma_{\hat{K}}(H_f)+\sigma_{\hat{K}+1}(H_f)}{2}\right)^2$.

ρ can be viewed as the step-size of the ADMM algorithm. ρ appears in multiple points of this set-up of ADMM where it seems likely that it counter acts itself, meaning the point of convergence will not change and ρ only acts as a catalyst to the rate of convergence. In various tests we have seen that ρ has no effect on the point of convergence, but the rate of convergence appears to have an optimal value.

7. Testing

The process of testing is itself of intrigue, before setting up our tests we must consider what hypotheses to construct. Initially, it seems intuitive that we are trying to approximate a signal as close to the ground truth as possible i.e. we want $\hat{F}(x) = F(x) - e(x)$. However, in real-life applications we can presume the noiseless signal never exists and it becomes unclear what we are trying to approximate. Ultimately, we are trying to approximate a signal formed by K complex exponentials without noise that act as a best fit for the original signal with noise. For this reason it is more applicable to compare distances to the corrupted signal, while paying attention to other factors like the rate of convergence and the rank/number of complex exponentials used.

Figure 3: Real (left) and imaginary (right) values of ζ



7.1. Setting up tests

We decided to implement testing with a similar structure to [9] to keep the testing in line with previous research on this topic. The results of the $\mathcal{S}^2$ transform are heavily dependent on the rank penalty parameter μ. As explained at the end of the previous section we choose $\mu = \frac{1}{2}\left(\frac{\sigma_{\hat{K}}(H_f)+\sigma_{\hat{K}+1}(H_f)}{2}\right)^2$. Experimentation suggests that the optimal value of $\rho \approx 1.25$ where the convergence rate tends to increase moving away from this value. For the tests we used the following eight values presented in the table below.

Signal values	
c_j	ζ_j
0.94737646+0.57831985i	0.96607614 + 22.94274544i
-0.93834136+0.80723275i	0.96607614 - 22.94274544i
0.93370208-0.74539468i	-0.4+38.39425339i
0.04446746-0.61503949i	-0.4-38.39425339i
0.68611731+0.94646068i	0.70902471+12.43801711i
-0.82304000+0.39125639i	0.70902471-12.43801711i
0.233645-0.36850826i	0+ 28.85837933i
-0.99867433+0.95581703i	0- 28.85837933 i

The values of c and the real parts of ζ were randomly generated by uniform distribution between -1 and 1. The imaginary parts of ζ were randomly generated with uniform distribution between 10 and 40 and we then used their negative complement for the following value. This range was used to ensure there was at least three to four revolutions of each individual complex exponential for $t \in (-1, 1)$. The real and imaginary values of ζ are plotted in Figure 3 below, where it is clear to see that each function has multiple revolutions, providing a modest sample space where the functions have little cross-over.

For the testing process we decided to compare optimality between the $\mathcal{S}^2$ transform against the ESPRIT method using the two types of weights discussed in this paper. For $W = 1$ testing was done by comparing

$$\frac{\|H_{\mathcal{S}^2} - H_f\| - \|H_{\text{ESPRIT}} - H_f\|}{\|H_f\|} \tag{32}$$

and for testing with weights as per (31), we compared

$$\frac{\|f_{\mathcal{S}^2_W} - f\| - \|f_{\text{ESPRIT}} - f\|}{\|f\|} \tag{33}$$

Results from tests categorised by the type of noise.

Random Gaussian noise

Scale	$\mathcal{S}^2$ transform	Accuracy of rank	Converged	$\mathcal{S}^2_W$-transform	Accuracy of rank	Converged
×1	1000/1000	100%	100%	955/1000	100%	100%
×5	1000/1000	100%	100%	960/1000	93.5%	98.3%
×20	1000/1000	99.1%	99.7%	990/1000	91%	90.3%
×40	785/1000	63.8%	98.7%	833/1000	50%	85%

Noise from music samples

Scale	$\mathcal{S}^2$ transform	Accuracy of rank	Converged	$\mathcal{S}^2_W$-transform	Accuracy of rank	Converged
×1	1000/1000	100%	100%	973/1000	100%	100%
×5	1000/1000	99.8%	100%	985/1000	93.5%	98%
×10	1000/1000	95.7%	99.5%	994/1000	56%	92.3%
×20	965/1000	48%	98.6%	590/1000	47.6%	91%

Columns 1 & 4 display the number of times $\mathcal{S}^2$ outperformed ESPRIT in tests (32) & (33) respectively.

For the noise we made 1,000 samples generated from pieces of music and another 1,000 samples generated from random Gaussian noise. Storing the noise like this allowed us to test both types of noise with four different strengths of signal to noise ratio on each sample, making it clearer to see how the scale of noise affects the methods. To ensure convergence of X and Y, ADMM was ran until $\|X - Y\| < 10^{-12}$, with a cut off of 10,000 iterations. Also noting on each test whether the rank of the returned matrix X was of correct rank, or that the method converged.

7.2. Results

With Gaussian noise the $\mathcal{S}^2$ transform overall outperforms ESPRIT, the magnitude of this improvement does increase for a higher scale of noise, which suggest the yield of the $\mathcal{S}^2$ transform is better for such situations. This result is very promising, however the magnitude of this difference is so slight that some further adjustments have to be made before any real benefit comes from applying the $\mathcal{S}^2$ transform. This could potentially be with a better choice of weight W or with an improvement on the iterative method used to solve (22).

When noise is taken from the samples of music the $\mathcal{S}^2$ transform again outperforms ESPRIT in both the weighted and un-weighted case. Similarly the magnitude of improvement on each test is generally growing as the scale of noise increases however the limit of this improvement is a lot lower, perhaps as the signal type from samples of music are closer to that of the type we are looking for, causing more interference.

As the scale of noise increases the accuracy of the rank of the output matrix decreases. However, in the tests returning an incorrect rank, the majority of these cases is where the rank is greater than eight, and upon inspecting the singular values of the returned matrix X, the magnitude still drastically decreases between the eight and ninth singular values, which indicates that the method needs more time to converge to the correct rank or a more optimal choice of ρ is needed. It should be highlighted that there are some cases which have a lower rank, and this suggests the algorithm is more turbulent as the noise increases.

On a similar note, the rate of convergence of the weighted case is generally slower and again it seems a more optimum choice of ρ could improve this, particularly as discussed in section 3.3 of [13]. Although not shown in the test results if we test (33) for $W = 1$ we see that ESPRIT out performs the $\mathcal{S}^2$ transform which suggests that working in a weighted Hilbert space makes a big improvement. Upon inspecting individual cases it appears that the end points are generally better for the weighted space.

Conclusion: The test results suggest that the theory behind the $\mathcal{S}^2$ transform hold, however some further optimization of the routine seems necessary. The magnitude of improvement

is generally so mild that for sheer simplicity ESPRIT is more effective. A strong characteristic of the $\mathcal{S}^2$ transform is that it provides convergence with a very high success and speed rate. Further, the theoretical framework for the $\mathcal{S}^2$ transform is much clearer then that of ESPRIT.

The Python code used has been uploaded to Github at : github.com/J-Son89/Stransform

8. Appendix

8.1. ESPRIT

Referring to Section 1, we assume that our input signal is noiseless and of the form (2). As per Section 1.1 we take equidistant samples of our input $f(j) = F(\frac{j-N}{N}), j \in 0, 1, ..., 2N$ and form the $N+1$ square Hankel matrix H_f, where H_f has rank K by Kronecker's Theorem for Hankel matrices. We assume that $N > K$ to ensure enough samples are taken of our input signal.

Letting Λ denote the Vandermonde matrix (each row contains the terms of a geometric progression), where $\Lambda(m, n) = e^{\zeta_n m}$ and let Λ_n denote the columns of Λ.

The following relation then holds for the entries of H_f,

$$H_f(m, n) = f_{m+n-2} = \sum_{j=1}^{K} c_j e^{\zeta_j (m+n-2)} = \sum_{j=1}^{K} c_j e^{\zeta_j (m-1)} e^{\zeta_j (n-1)}.$$

Using this we see that H_f can be denoted by

$$H_f = \sum_{j=1}^{K} c_j \Lambda_j \Lambda_j^T = \Lambda diag(c) \Lambda^T,$$

where $c = (c_1, ..., c_K)$. For an SVD of $H_f = U \Sigma \mathcal{V}^*$, from Ch4.4 of [10] , it holds that the singular vectors U are linear combinations of the columns of Λ, i.e.

$$U = \Lambda B$$

where $B \in \mathbb{M}^K$ is invertible. Note that U is a $N + 1 \times K$ matrix, where the excess columns of U have been removed as we are only interested in the columns which have data relevant to our non-zero singular values.

Introducing some new notation, we let $D = diag(e^{\zeta_1}, ..., e^{\zeta_K})$ and Λ^+ be the matrix that yields from removing the first row of Λ, and similarly Λ^- be the matrix that yields from removing the last row of Λ. There is a clear relation between Λ^+ and Λ^-, namely that

$$\Lambda^+ = \begin{pmatrix} e^{\zeta_1 \cdot 2} & e^{\zeta_2 \cdot 2} & ... & e^{\zeta_K \cdot 2} \\ e^{\zeta_1 \cdot 3} & ... & ... & e^{\zeta_K \cdot 3} \\ \vdots & ... & ... & \vdots \\ e^{\zeta_1 \cdot N+1} & ... & ... & e^{\zeta_K \cdot N+1} \end{pmatrix} = \begin{pmatrix} e^{\zeta_1 \cdot 1} & e^{\zeta_2 \cdot 1} & ... & e^{\zeta_K \cdot 1} \\ e^{\zeta_1 \cdot 2} & ... & ... & e^{\zeta_K \cdot 2} \\ \vdots & ... & ... & \vdots \\ e^{\zeta_1 \cdot N} & ... & ... & e^{\zeta_K \cdot N} \end{pmatrix} \begin{pmatrix} e^{\zeta_1} & 0 & ... & 0 \\ 0 & e^{\zeta_2} & 0 & \vdots \\ \vdots & ... & ... & 0 \\ 0 & ... & 0 & e^{\zeta_K} \end{pmatrix} = \Lambda^- D.$$

Also it holds that

$$U^+ = \Lambda^+ B = \Lambda^- D B$$

and

$$U^- = \Lambda^- B$$

which can be easily verified. Performing a number of manipulations on the matrix $A = (U^-)^* U^+$ returns a promising result.

$$\begin{aligned} A = (U^-)^* U^+ &= \big((U^-)^*(U^-)\big)^{-1}\big((U^-)(U^+)\big) \\ &= \big(B^*(\Lambda^-)^*(\Lambda^- B)\big)^{-1}\big(B^*(\Lambda^-)^*(\Lambda^-)DB\big) \\ &= B^{-1}\big((\Lambda^-)^*(\Lambda^-)\big)^{-1}(B^*)^{-1}B^*(\Lambda^-)^*(\Lambda^-)DB \\ &= B^{-1}DB \end{aligned}$$

Given that D is a diagonal matrix it is clear to see from this result that the columns of B^{-1} are eigenvectors of A, where $e^{\zeta_j/N}$ are the corresponding eigenvalues. Thus, the exponentials $e^{\zeta_j/N}$ can be recovered from H_f by diagonalizing the matrix A by performing an eigenvalue decomposition of A. Finally we take the ln of the values and given that the initial ζ have been sampled to a scale of $\frac{\zeta}{N}$ we have to rescale them to a factor of N to retrieve the initial values.

9. References

[1] Fredrik Andersson and Marcus Carlsson. *Alternating projections on nontangential manifolds.* 38(3):489–525, 2013.

[2] Fredrik Andersson and Marcus Carlsson. *ESPRIT for multimendsional general grids.* 2010 MSC: 15B05, 41A63, 42A10

[3] Marcus Carlsson. *On convexification/optimization of functionals including an l^2-misfit term.* 1609.09378v2, 2017.

[4] Viktor Larsson and Carl Olsson. *Convex low rank approximation.* International Journal of Computer Vision, pages 1–21, 2016.

[5] Fredrik Andersson and Marcus Carlsson. *On the Structure of Positive Semi-Definite Finite Rank General Domain Hankel and Toeplitz Operators in Several Variables* Complex Anal. Oper. Theory (2017) 11:755–784 DOI 10.1007/s11785-016-0596-6

[6] John von Neumann. *Some matrix inequalities and metrization of matric-space* Tomsk Univ. Rev. 1:286-300 (1937); in Collected Works, Pergamon, Oxford, 1962, Vol. IV, pp. 205-218.

[7] Stephen Boyd and Neal Parikh. *Proximal Algorithms* Foundations and Trends in Optimization Vol. 1, No. 3 (2013) 123–231

[8] Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces.*

[9] Fredrik Andersson, Marcus Carlsson, Carl Olsson *Convergence of dual ascent in non-convex/non-differentiable optimization* arXiv:1609.06576

[10] Roger A Horn and Charles R Johnson. *Matrix analysis.* Cambridge university press, 2012.

[11] Peter D. Lax. *Functional Analysis* (Wiley Pure and Applied Mathematics: A Series of Texts, Monographs and Tracts), 2002, ISBN: 978-0-471-55604-6.

[12] Jonathan Gillard, Anatoly Zhigljavsky. *Optimization challenges in the structured low rank approximation problem* J Glob Optim (2013) 57:733–751 DOI 10.1007/s10898-012-9962-8

[13] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato and Jonathan Eckstein. *Foundations and Trends in Machine Learning* Vol. 3, No. 1 (2010) 1–122 DOI: 10.1561/2200000016