UiO : Department of Mathematics
University of Oslo

Coherence Estimates Between Hadamard Matrices and Daubechies Wavelets

Vegard Antun

Master's Thesis, Spring 2016

Abstract

Traditionally the compressive sensing theory have been focusing on the three principles of *sparsity*, *incoherence* and *uniform random subsampling*. Recent years research have shown that these principles yield insufficient results in many practical setups. This has lead to the development of the principles of *asymptotic sparsity*, *asymptotic incoherence* and *multilevel random subsampling*.

As a result of these principles, the current theory is limited to unitary sampling and sparsifying operators. For large scale reconstruction, the theory is further restricted to operators whose product can be computed in $\mathcal{O}(N \log_2 N)$ operations, due to memory constraints of computers. Accordingly this has increased the popularity of the Fourier and Hadamard sampling operators, for applications where these operators can model the underlying sampling structure. As the sparsifying operator the wavelet transform have proven to yield satisfactory results in most setups. Since all of these operators needs to be unitary, this have restricted us to only consider Daubechies compactly supported orthonormal wavelets.

By using wavelets as the sparsifying transform it has been proven that a Fourier sampling basis will be asymptotically incoherent to a unitary wavelet basis. The same result can easily be calculated numerically between a Hadamard sampling basis and a Daubechies wavelet basis. However, any theoretical result of this fact have been lacking. The purpose of this text is to provide such a theoretical result.

Preface

Overview of the thesis

In this thesis we will shortly review the limitations and the most important results of the standard theory in compressive sensing. We will then turn to the theory proposed by Adcock, Hansen, Poon & Roman in [2] to motivate the need for a coherence estimate between Hadamard matrices and Daubechies wavelets.

We will then give a short introduction to wavelets and the Hadamard transform in chapter 3 and 4. These chapters will also contain most of the results needed in the derivation of the new coherence result. In chapter 5 we will state most of the formal theory known from [2], and conduct some practical experiments verifying their accuracy. In particular these results indicate that both a Fourier and Hadamard sampling basis can benefit from this asymptotic theory. It also suggests that both of these bases are asymptotically incoherent when they are combined with wavelets. This property will then be proven to be true for Hadamard matrices combined with wavelets in chapter 6.

The purpose of this thesis has been to create this new coherence estimate between Hadamard matrices and Daubechies wavelets. The focus has therefore been to construct a proof of this, rather than rewriting proofs of well known theorems. Most of the theorems stated in this thesis will therefore be given without any proof. For theorems which have a particularly short and simple proof, some of them have been provided, to give the reader a better understanding of the text.

Code

The code produced during the work with this thesis is to extensive to be included as a part of the text. Most notable of all this code is a C++ implementation of the Hadamard transform, with bindings to Python and MATLAB. This code extends NumPy's module with lacking functionally and outperform MATLAB's own `fwht(...)` function. This can be seen in figure 0.1, where the computational time between the two have been time measured for arrays of different sizes. As this code may be of independent interest, it has been created as a project on its own under the name *Fastwht*. The project also provide an implementation in C++ of the WAL and PAL functions with an interface to Python. As the author never needed these in MATLAB an interface have not yet been provided. The code is found at https://bitbucket.org/vegarant/fastwht.

Most of the figures included in this text have been created using Python scripts, while all of the sketches have been written in Tikz. All of this code have been published in the git repository https://bitbucket.org/vegarant/code-thesis. This repository also includes all the MATLAB scripts used in the reconstruction process of compressive sensing. These scripts use the SPGL1 software package [5, 6] to solve the basis pursuit denoise problem. All of these scripts also rely on the Python and MATLAB wavelet library developed by Øyvind Ryan. It is found at http://folk.uio.no/oyvindry/matinf2360/code/. For interested readers I have also included five different solvers of the basis pursuit problem, written in Python. These solvers are however not suitable for large scale problems, and have been written for readability rather than performance.

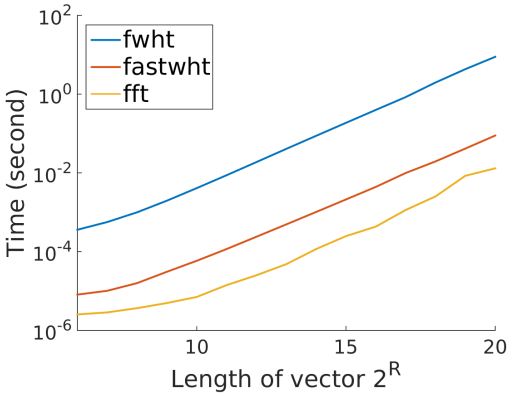


Figure 0.1: The difference in performance between MATLAB's Hadamard transform, the authors Hadamard transform and MATLAB's implementation of the fast Fourier transform for various array lengths.

Due to the authors personal interests in convex optimization solvers used in compressive sensing, there have also been developed a C++ implementation of the SPGL1 solver. This have been published as its own project on https://bitbucket.org/vegarant/spgl1. This code also include my own C++ implementation of a Daubechies wavelet transform for various number of vanishing moments. All of the code in this project is working correctly, but it is poorly documented, and have not yet been optimized. As a result it is harder to maintain and order of magnitude slower than the MATLAB package. To reduce development time, the MATLAB package have therefore been preferred.

Acknowledgements

First and foremost I would like to thank my two supervisors Øyvind Ryan and Anders Hansen. In this processes Øyvind have been the one to help me with my weekly challenges, proof reading parts of my code and guiding me through the world of wavelets. Anders on the other hand have guided me through much of the newly developed theory, suggesting relevant reading material and possible solutions. Without their help and supervision, this thesis would never have become what it is today.

I would also like to thank my friend and former fellow student Anders Matheson, for lighting my passion for computer science during our days as college freshmen. Without your interest to share knowledge about Arch Linux, C++ and other good practical solutions, I would not have the computational experience I have today. Thank you for many enlightening discussions.

Additionally I would like to thank my friend and fellow student Paul A. Maugesten for lending me a camera capable of capturing RAW images, and for all the cheerful morning meetings we have had the past year.

For the past two years I have been seated at the study hall B601 in Niels Henrik Abels building at Blindern, Oslo. I would like to thank all of whom I have shared the room with, for accompanying me during my master studies.

Contents

List of acronyms

BP	Basis pursuit
BT	Basic thresholding
CoSaMP	Compressive sampling matching pursuit
CS	Compressive sensing
CT	Computerized tomography
DBx	Daubechies wavelet with x vanishing moments
DFT	Discrite Fourier transform
DWT	Discrete wavelet transform
FFT	Fast Fourier transform
HTP	Hard thresholding pursuit
IDWT	Inverse discrete wavelet transform
IHT	Iterative hard thresholding
LASSO	Least absolute shrinkage and selection operator
MRA	Multiresolution analysis
MRI	Magnetic resonance imaging
NSP	Null space property
OMP	Orthogonal matching pursuit
QCBP	Quadratically constrained basis pursuit
RIP	Restricted isometry property
SPGL1	Spectral projected gradient ℓ_1

CHAPTER 1

Introduction

1.1 Motivation

For decades the sampling theory have been dictated by the famous theorem of *Shannon* and *Nyquist*. This theorem states that in order to obtain perfect signal recovery of a uniformly sampled signal, whose maximum frequency is γ, the signal must be sampled at a rate which exceeds 2γ. Any sampling below this rate would cause *aliasing*, and thus suboptimal results. A consequence of this theorem has therefore been high sampling rates for high-frequent signals. Such sampling rates will thus result in an enormous amount of data.

As an example consider a sound whose highest frequency is 20 000 Hz. A recoding of this sound would require more than 40 000 samples/sec. Sampling at this rate for one hour, using say 16 bit/samples would result in at least 288 Mb of data. Recoding on multiple channels would increase this size further.

This amount of data would be impractical to most applications, so to overcome this difficulty one usually compress the collected data before storage. This has lead to the development of a swarm of different compression formats which are able to extract and store only the important parts of all the data, created by the high sampling rates.

A general strategy for many of these compression formats is to transform the sampled signal into a different basis, where most of the signal coefficients are close to zero, while only a few are large in magnitude. The key observation used by these compression algorithms is that all the small coefficients contain very little "information", in the sens that if we neglect all coefficients below a certain threshold by setting them to zero, and transform the signal back to the original domain, the signal would be nearly identical to the original one. Hence, in order to reduce the size of the data, only the significant coefficients in the transformed signal are stored

The important observation, exploited by these algorithms is the empirical fact that most real life signals are *sparse*, if they are written in the right domain. It is this observation which gives rise to the field of *compressive sensing* and enables us to beat the traditional sampling rate introduced by Shannon and Nyquist.

The idea behind this revolutionary recovery technique, is to use a linear non-adaptive signal acquisition process, so that the acquisition can be modeled as a sensing matrix $\mathbf{A} \in \mathbb{C}^{m \times N}$. By letting $\boldsymbol{b} \in \mathbb{C}^m$ denote the set of measurements,

the signal $\boldsymbol{x} \in \mathbb{C}^N$ can be found as a solution of the equation

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{b}. \tag{1.1}$$

As the idea is to use fewer measurements than the traditional sampling rate, we implicitly assume that $m < N$. This leaves us with the redundant system $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$, which have infinitely many solutions, making it impossible to pick the right one.

To overcome this difficulty the compressive sensing technique has to add some further assumptions on the measured signal $\boldsymbol{x}$, in order to obtain a unique solution of equation (1.1). It is here the important observation of sparsity comes into play. By doing our measurements in a smart way – which will be clarified later – and assuming that the original signal $\boldsymbol{x}$ is sparse, any vector $\boldsymbol{z} \in \mathbb{C}^N$ with the same number of non-zero coefficients as $\boldsymbol{x}$ will be the unique solution of $\mathbf{A}\boldsymbol{z} = \boldsymbol{b}$, i.e., $\boldsymbol{z} = \boldsymbol{x}$.

To clarify the key idea of compressive sensing, we will present it through a classical example of a counterfeit coin. Before we start on the example, we note that we will use the usual convention of letting $\boldsymbol{e}_j$ be a column vector with only zero entries, except for a 1 at position j. We will also let

$$\operatorname{sgn}(z) := \begin{cases} \frac{z}{|z|} & \text{if } z \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

denote the sign of the number z. If $\boldsymbol{z}$ is a vector we let it be the component wise sign of $\boldsymbol{z}$.

Example 1.1. You are given 12 golden coins, one of which is a counterfeit. All the coins are of equal shape and color, but the counterfeit's mass differ from the others. Using a balance scale, how many measurements are needed to detect the counterfeit?

As it turns out this can be solved non-adaptively, using only three measurements. The solution reads as follows: Label the 12 coins from $1, 2, \ldots 12$, and let each coin correspond to a column in the sensing matrix $\mathbf{A}$. Let each row of the matrix correspond to a weighting of a subset of the coins. Any entry (i, j) of the sensing matrix should then be either -1, 1, or 0, if coin j was placed either on the left scale, on the right scale or off the scales, respectively, during weighting i. A sensing matrix which could solve the problem, would place 4 coins on either side of the scales in each weighting. It could for instance look like this

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & -1 & 1 & -1 & 0 & 1 & -1 & 1 \\ 0 & 1 & 0 & -1 & -1 & 0 & -1 & 0 & 1 & 1 & 1 & -1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 & 1 & -1 & -1 & 1 & 1 \end{bmatrix}.$$

We know that each of the authentic coins have the same mass μ, and that the counterfeit have a different mass μ'. This implies that the mass vector $\boldsymbol{x}$ must have the following form

$$\boldsymbol{x} = \mu \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}^T + (\mu' - \mu)\boldsymbol{e}_j \in \mathbb{R}^{12}.$$

As we weight four coins on either side in all weightings, all rows have four 1's, four 0's and four -1's. This will make the first constant vector term in $\boldsymbol{x}$ cancel

in each weighting. As a result we get the following relation

$$\mathbf{A}\boldsymbol{x} = (\mu' - \mu)\mathbf{A}\boldsymbol{e}_j.$$

Because a balance scale is used, it will only tell us which of the two sides are heavier. We will therefore not have access to the difference $\mu' - \mu$, but rather the sign of this difference i.e., $\operatorname{sgn}(\mu' - \mu)$. To overcome this difficulty we could write the result of each weighting using one of the numbers -1, 0 and 1, depending on which of the scales that fell to the ground. Doing so we are searching for a solution of

$$\operatorname{sgn}(\mathbf{A}\boldsymbol{x}) = \operatorname{sgn}\left((\mu' - \mu)\mathbf{A}\boldsymbol{e}_j\right).$$

The measurement vector $\boldsymbol{b} = \operatorname{sgn}(\mu' - \mu)\mathbf{A}\boldsymbol{e}_j$ would then equal one of the columns $\mathbf{A}$ times the sign of $\operatorname{sgn}(\mu' - \mu)$. As we know that there is only one single counterfeit coin, we can now easily extract coin j from the authentic ones [9]. ♣

In the example above, the sparsity was simply created as a part of the problem. Similar examples can be found in error correction, machine learning [21, pp. 19–21] or facial recognition [18, ch. 12]. In general however, we can not assume that the problem at hand will have a sparse solution $\boldsymbol{x}$ without imposing a sparsifying transform $\boldsymbol{\Psi} \in \mathbb{C}^{N\times N}$, similar to the ones used by the compression algorithms. Consequently we have to modify the system in equation (1.1), into

$$\mathbf{A}\boldsymbol{\Psi}^{-1}\boldsymbol{z} = \boldsymbol{b} \tag{1.2}$$

with $\boldsymbol{z} = \boldsymbol{\Psi}\boldsymbol{x}$. This means that we first have to find the sparse solution $\boldsymbol{z}$ of the underdetermined system in equation (1.2), and then reconstruct $\boldsymbol{x}$ from $\boldsymbol{z}$ using $\boldsymbol{\Psi}^{-1}$.

1.2 Random sensing matrices

An open question within the field of compressive sensing, have been to construct good sensing matrices $\mathbf{A}$, so the reconstruction of $\boldsymbol{z}$ can be done with as few measurements as possible. Traditionally many of these sensing matrices have been random matrices, where each entry is drawn from some probability distribution. Typical distributions would be the standard normal Gaussian distribution or a Bernoulli distribution with equal probability of the entry 1 and -1. For such random matrices one can guarantee reconstruction of a sparse vector $\boldsymbol{x}$ with high probability provided that the number of measurements meets the famous lower bound of

$$m \geqslant Cs\log(N/s) \tag{1.3}$$

measurements [21, p. 6]. Where C is a constant which only depends on the probability distribution, N is the length of the vector, and s is the number of non-zero coefficients of $\boldsymbol{x}$. To simplify notation a vector $\boldsymbol{x}$ will be called *s-sparse* if it contains at most s non-zero coefficients.

These random matrices do, however, have two major limitations. The first of these are all of the applications where the sampling operator is given due

to physical constraints of the imaging system. These constraints are found in almost any medical imaging system, such as *magnetic resonance imaging* (MRI) and *computerized tomography* (CT) [2].

To see how such an operator is imposed consider magnetic resonance imaging. In this application a specimen is exposed to different frequencies while lying in a magnetic field. This will cause the specimen to emit frequencies whose wavelength depends linearly on the strength of the magnetic field. By imposing a varying magnetic field around the specimen, one can construct an image based on the strength and wavelength of the emitted frequencies.

To model this setup, one must apply a sampling operator, whose rows corresponds to different frequencies. Otherwise the sampling operator would have no physical interpretation. In these types of applications a popular choice have therefore been to select the rows from the Fourier matrix

$$\mathbf{V}_{\text{dft}} = \left[\exp(-2\pi ijk/N)/\sqrt{N}\right]_{j,k} \in \mathbb{C}^{N\times N}, \quad j,k = 1,\dots,N$$

whose frequencies coincides with the sampled frequency.

Other examples are found in computerized tomography where one is unable to illuminate the specimen from all angles, and the coin example above where any matrix entry which is not equal to one of the numbers -1, 0 and 1 would be impractical at best. Other similar examples are single-pixel imaging and fluorescence microscopy, were the matrix entires are either 1 or 0. Thus a sampling matrix whose entries are drawn from some probability distribution can not always model the underlying sampling pattern.

The second problem with random matrices is that one needs to store the entire matrix $\mathbf{A}$. As an example assume we have an image with 1024×1024 pixels, written as a column vector $\boldsymbol{x} \in \mathbb{C}^{2^{20}}$, and lets say we use a subsampling rate of 10%. Using double floating-point arithmetic this would require us to store a $\frac{1}{10} \cdot 2^{20} \times 2^{20}$ random matrix, resulting in a memory consumption of approximately 880 GB! This makes the problem incomputable without a supercomputer.

1.3 A new sampling scheme

To overcome the computational bottleneck of large, densely stored matrices, we will only address a sampling scheme similar to the one proposed by Adcock, Hansen, Poon & Roman in [2]. In this paper one restricts the search for sampling operators to unitary matrices which do not need to be stored in memory, but whose matrix product can be computed in-place with $\mathcal{O}(N \log N)$ operations, rather than the usual $\mathcal{O}(N^2)$. As our sampling operator $\mathbf{V} \in \mathbb{C}^{N\times N}$ we will therefore only consider the Fourier transform and Hadamard transform. Similarly we will only consider the wavelet transform as the sparsifying operator. This enable us to model the sampling process as

$$\mathbf{P}_\Omega \mathbf{V}\boldsymbol{x} = \mathbf{P}_\Omega \mathbf{V}\boldsymbol{\Psi}^{-1}\boldsymbol{z} = \boldsymbol{b} \tag{1.4}$$

where $\mathbf{P}_\Omega \in \mathbb{R}^{m\times N}$ is a projection matrix selecting which of the m rows we would like to sample. The subset $\Omega \subset \{1,\dots,N\}$, $|\Omega| = m$ will contain the indices of the chosen rows. Traditionally one of the fundamental concepts in compressive sensing have been to use *random subsampling* of the indices in

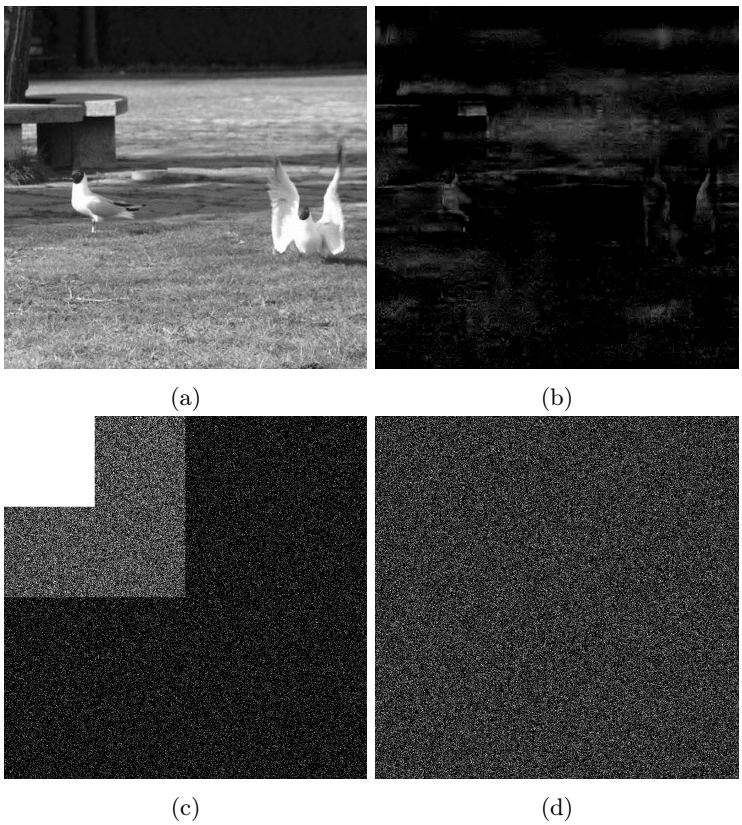


Figure 1.1: The two images seen in (a) and (b) are reconstructions of an image which have been sampled by a Hadamard matrix and sparesified by a two dimensional Daubechies 4 (DB4) wavelet. Image (a) used the structured sampling scheme seen in (c), while image (b) used the uniformly random subsampling scheme seen in (d). Both sampling schemes used the same number of samples.

Ω. This have, however, proven to produce suboptimal results. The reason for this is, as we shall see in chapter 5, that the signal $\boldsymbol{z}$ is not sparse, it is *asymptotically sparse*. This means that most of the large magnitude coefficients are stored at the beginning of $\boldsymbol{z}$, while the vector becomes more sparse as one moves towards the end. A more optimal sampling scheme Ω therefore takes this structure into account by favoring the rows in the sampling matrix.

Another principle known from this traditional theory is the use of an *incoherent* sampling operator.

Definition 1.2 (coherence)**.** Let $\mathbf{V}, \mathbf{\Psi} \in \mathbb{C}^{N \times N}$ be unitary matrices with columns $\boldsymbol{v}_i$ and $\boldsymbol{\psi}_j$, respectively. The *coherence* of the matrix $\mathbf{U} = \mathbf{V}\mathbf{\Psi}^*$ is defined as

$$\mu(\mathbf{U}) \coloneqq \max_{i,j=1,\dots,N} |\langle \boldsymbol{v}_i, \boldsymbol{\psi}_j \rangle|^2 \in \left[\tfrac{1}{N}, 1\right]$$

where $\mathbf{\Psi}^*$ denotes the conjugate transpose of the matrix $\mathbf{\Psi}$. We say that $\mathbf{U}$ is perfectly incoherent if $\mu(\mathbf{U}) = N^{-1}$.

Due to the result of Candes & Plan [10] and Adcock & Hansen [1], we know that the setup in equation (1.4) can recover $\boldsymbol{x}$ exactly with a probability of at least $1 - \epsilon$ if

$$m \geqslant C \cdot \mu(\mathbf{V}\boldsymbol{\Psi}^*) N s (1 + \log(1/\epsilon)) \log(N). \tag{1.5}$$

Hence by using an incoherent basis $\mu(\mathbf{V}\boldsymbol{\Psi}^*) = N^{-1}$ we see that the inequality (1.5) reduces to the bound of (1.3).

The sampling bases proposed earlier do not possess this property. For a zero frequency Fourier measurement, it is well known [29, Thm. 7.2] that using a orthonormal wavelet scaling function ϕ will result in a coherence[1] $\mu(\mathbf{V}_{\text{dft}}, \boldsymbol{\Psi}^*) = |\langle e^0, \phi_0 \rangle|^2 = \mathcal{O}(1)$. As the first row of any Hadamard matrix only consists of 1's, a similar result can be derived for this basis. By the inequality (1.5) this implies that using these sampling bases should result in a sampling rate $m > N$ i.e., the opposite of compressive sensing.

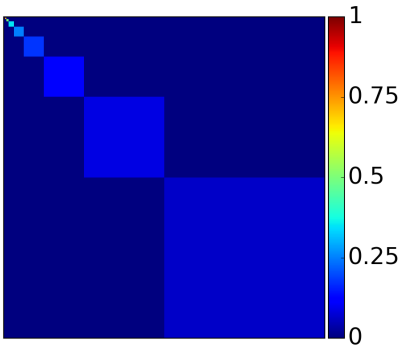
In practice these bases works perfectly fine with $m < N$ measurements, given the right sampling strategy Ω. This can bee seen in figure 1.1 where two different sampling strategies results in two completely different reconstructions. A mathematical justification for these empirical results will be reviewed in chapter 5. As we shall see in that chapter a necessary condition for the success of the structured sampling scheme seen in figure 1.1 is the principle of *asymptotic incoherence* between the sampling basis $\mathbf{V}$ and the sparsifying basis $\boldsymbol{\Psi}$. This is defined as follows

Definition 1.3 (Asymptotic incoherence)**.** Let $\{\mathbf{U}_N\}$ be a sequence of isometries with $\mathbf{U}_N \in \mathbb{C}^{N \times N}$ and let $\mathbf{P}_N^K$ denote the projection onto $\text{span}\{\boldsymbol{e}_j : j = K+1, \ldots, N\}$. Then $\{\mathbf{U}_N\}$ is *asymptotically incoherent* if $\mu(\mathbf{P}_N^K \mathbf{U}_N) \to 0$ and $\mu(\mathbf{P}_N^K \mathbf{U}_N) \to 0$ when $K \to \infty$, with $N/K = c$ for all $c \geqslant 1$.

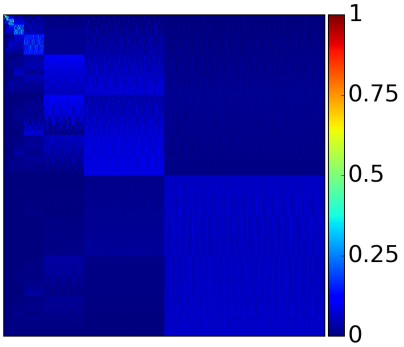
In short, this definition tells us that if K first rows or columns are removed from the isometry $\mathbf{U}_N$ and the coherence of the resulting matrix is small, we shall call the isometry asymptotically incoherent.

Due to the success of the structured sampling scheme in figure 1.1c we would expect the Hadamard sampling basis to be asymptotically incoherent to a Daubechies wavelet matrix. In figure 1.2 we have plotted the absolute values of these two matrices multiplied together. From this figure we see that the large magnitude coefficients are stored in the upper left corner, and that the magnitudes decreases as one moves away from this corner. This suggests that these operators could be asymptotically incoherent. It is asymptotic structure we will find a theoretical justification for in chapter 6.

[1] See remark 3.3 for an explanation of notation.



(a) DB1



(b) DB4

Figure 1.2: The magnitude of the matrix entries of $|\mathbf{V}_{\text{had}}\mathbf{\Psi}^{-1}|$, where $\mathbf{V}_{\text{had}}$ is the Hadamard matrix, and $\mathbf{\Psi}$ is a Daubechies wavelet matrix.

CHAPTER 2

Elementary compressive sensing

2.1 Solving underdetermined systems

The fundamental problem of compressive sensing is to solve the underdetermined system

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{b} \tag{2.1}$$

where $\mathbf{A} \in \mathbb{C}^{m\times N}$, $\boldsymbol{x} \in \mathbb{C}^N$ and $\boldsymbol{b} \in \mathbb{C}^m$ with $m < N$. Hence, if there exists at least one solution of equation (2.1), this system will contain infinitely many solutions. To overcome this problem we introduced the assumption of a sparse solution $\boldsymbol{x}^\sharp$ of equation (2.1). By using this assumption one can reformulate the problem into

$$\text{minimize } ||\boldsymbol{x}||_0 \quad \text{subject to} \quad \mathbf{A}\boldsymbol{x} = \boldsymbol{b} \tag{P_0}$$

where the notation $||\cdot||_0$ refer to the ℓ_0-"norm". That is $||\boldsymbol{x}||_0 := |\text{supp}(\boldsymbol{x})|$, with $\text{supp}(\boldsymbol{x}) := \{j : x_j \neq 0\}$. The use of the word "norm" in the sentence above can be misleading as $||\cdot||_0$ does not satisfy the triangle inequality. It is, however, customary to denote it as a norm, so we will continue this practice.

The system (P_0) can be solved uniquely according to the following theorem

Theorem 2.1 ([15, Corollary 1.1])
Let $\mathbf{A} \in \mathbb{C}^{m\times N}$. Every s-sparse vector $\boldsymbol{x} \in \mathbb{C}^N$ is the unique solution of $\mathbf{A}\boldsymbol{z} = \mathbf{A}\boldsymbol{x}$, that is if both $\boldsymbol{x}$ and $\boldsymbol{z}$ is s-sparse then $\boldsymbol{x} = \boldsymbol{z}$, if and only if every set of $2s$ columns of $\mathbf{A}$ is linearly independent.

Proof. Assume $\boldsymbol{x}$ is the unique s-sparse solution of $\mathbf{A}\boldsymbol{x} = \mathbf{A}\boldsymbol{z}$, and let $\boldsymbol{v} \in \ker \mathbf{A}$ be $2s$-sparse. Then we can write $\boldsymbol{v} = \boldsymbol{x} - \boldsymbol{z}$ for some s-sparse vector $\boldsymbol{z}$ with $\text{supp}(\boldsymbol{z}) \cap \text{supp}(\boldsymbol{x}) = \varnothing$. But as every s-sparse solution of $\mathbf{A}\boldsymbol{x} = \mathbf{A}\boldsymbol{z}$ is unique, and the support of $\boldsymbol{x}$ and $\boldsymbol{z}$ are disjoint, it follows that $\boldsymbol{x} = \boldsymbol{z} = 0$.

Conversely assume every set of $2s$ columns of $\mathbf{A}$ are linearly independent, further let both $\boldsymbol{x}$ and $\boldsymbol{z}$ be s-sparse. Then $\boldsymbol{x} - \boldsymbol{z}$ is $2s$-sparse and the solution of $\mathbf{A}(\boldsymbol{x} - \boldsymbol{z}) = 0$ is unique. □

Unfortunately the (P_0) problem is NP-hard. This means that even a moderately sized problem is impossible to solve within a reasonable time-constraint. To overcome this difficulty one therefore applies the standard technique of relaxing the formulation of (P_0) into a convex optimization problem. For a

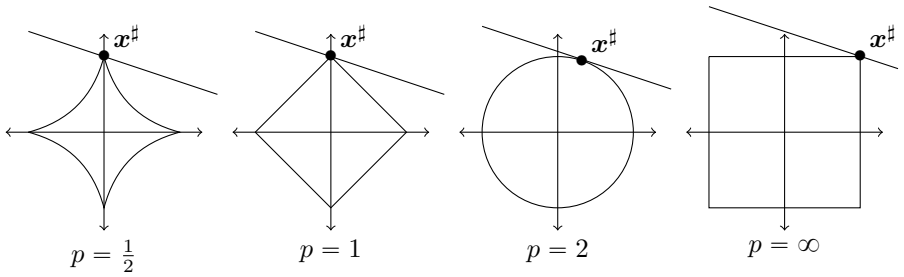


Figure 2.1: The solution of maximize $[1 \ -1/3][x_1 \ x_2]^T$ subject to $||\boldsymbol{x}||_p = 1$, for various values of p.

problem involving only real variables, the default relaxation is known as *basis pursuit* (BP) and can be formulated as

$$\text{minimize } ||\boldsymbol{x}||_1 \quad \text{subject to} \quad \mathbf{A}\boldsymbol{x} = \boldsymbol{b}. \tag{BP}$$

This can be reformulated into a *linear optimization problem* and solved efficiently. For the complex case we relax the problem further into the *quadratically constrained basis pursuit* (QCBP) problem. That is

$$\text{minimize } ||\boldsymbol{x}||_1 \quad \text{subject to} \quad ||\mathbf{A}\boldsymbol{x} - \boldsymbol{b}||_2 \leqslant \eta \tag{$P_{1,\eta}$}$$

which can be reformulated into a *second-order cone problem* and solved by appropriate software.

For an intuitive justification for our use of the ℓ_1-norm we refer to figure 2.1. Here one can easily see that any ℓ_p-norm is non-convex for $0 < p < 1$. This makes the problems intractable to solve. For the choice $p = 1$ we see that the norm consists of four extreme faces, all of which corresponds to sparse solutions, as opposed to $p = \infty$. The easiest choice would be $p = 2$, which could be solved using a least squares solution. This norm does, however, create a convex set with infinitely many extreme faces, which implies that no sparse solutions will be favoured.

As any ℓ_1-solution of (BP), is a relaxation of the ℓ_0-solution of (P_0), our main interest is when these solutions coincide. To create such results we need to impose certain restrictions on the sensing matrix $\mathbf{A}$. One such restriction ensuring such results is called the *null space property* (NSP). To introduce this concept, we first need to settle on some new notation.

Let $S \subset \{1, \dots, N\}$, then its compliment S^c will be with respect to the set $\{1, \dots, N\}$. For a vector $\boldsymbol{x} \in \mathbb{C}^N$, we shall let the vector $\boldsymbol{x}_S \in \mathbb{C}^{|S|}$ consist of the entries of $\boldsymbol{x}$ indexed in S. Similarly for a matrix $\mathbf{A} \in \mathbb{C}^{m \times N}$ we shall let $\mathbf{A}_S \in \mathbb{C}^{m \times |S|}$ be the matrix consisting of the $|S|$ columns in $\mathbf{A}$ labeled in S.

Definition 2.2 (Null space property)**.** A matrix $\mathbf{A} \in \mathbb{C}^{m \times N}$ is said to satisfy the *null space property* relative to a set $S \subset \{1, \dots, N\}$ if

$$||\boldsymbol{v}_S||_1 < ||\boldsymbol{v}_{S^c}||_1 \quad \text{for all } \boldsymbol{v} \in \ker \mathbf{A} \backslash \{0\}.$$

It is said to satisfy the null space property of order s if it satisfy the null space property for any set $S \subset \{1, \dots, N\}$ with $|S| \leqslant s$.

Applying this property, one can derive the following theorem

Theorem 2.3 ([21, p. 79])
Given a matrix $\mathbf{A} \in \mathbb{C}^{m\times N}$*, every* s*-sparse vector* $\boldsymbol{x} \in \mathbb{C}^N$ *is the unique solution of* (BP) *if and only if* $\mathbf{A}$ *satisfy the null space property of order* s*.*

Using this theorem, one can easily derive that any solution $\boldsymbol{x}^\star$ of (P_0) will coincide with a solution $\boldsymbol{x}^\sharp$ of (BP). To see this, remember that $||\boldsymbol{x}^\star||_0 \leqslant ||\boldsymbol{x}^\sharp||_0$, which implies that $\boldsymbol{x}^\star$ is s-sparse. By the theorem above any s-sparse solution of a matrix satisfying the null space property of order s is unique i.e., $\boldsymbol{x}^\star = \boldsymbol{x}^\sharp$.

Other similar requirements on the sensing matrix $\mathbf{A}$, which ensure a unique solution of the ℓ_1-optimization problem, have also been derived. One of these is the *restricted isometry property* (RIP).

Definition 2.4 (Restricted isometry property)**.** The s^{th} restricted isometry property constant $\delta_s = \delta_s(\mathbf{A})$ of a matrix $\mathbf{A} \in \mathbb{C}^{m\times N}$ is the smallest $\delta \geqslant 0$ such that

$$(1-\delta)||\boldsymbol{x}||_2^2 \leqslant ||\mathbf{A}\boldsymbol{x}||_2^2 \leqslant (1+\delta)||\boldsymbol{x}||_2^2$$

for all s-sparse vectors $\boldsymbol{x} \in \mathbb{C}^N$.

Theorem 2.5 ([21, pp. 142–143])
Suppose that the $2s^{th}$ *restricted isometry constant of the matrix* $\mathbf{A} \in \mathbb{C}^{m\times N}$ *satisfy*

$$\delta_{2s} < \frac{1}{3}.$$

Then every s*-sparse vector* $\boldsymbol{x} \in \mathbb{C}^N$ *is the unique solution of* (BP)

To see the resemblance between the constant $\delta = \frac{1}{3}$ for the ℓ_1-problem, and the ℓ_0-problem, consider the latter for a moment. For this problem we could require that $\delta_{2s} < 1$. For such a δ, any subset $S \subset \{1,\dots,N\}$ with $|S| \leqslant 2s$ will provide a matrix $\mathbf{A}_S^T\mathbf{A}_S$ with non-zero singular values. Any such matrix is non-singular, which again implies that any subset of $2s$ columns of $\mathbf{A}$ are linearly independent. Hence, due to theorem 2.1 it would provide a unique ℓ_0-minimizer. The condition $\delta_{2s} < 1/3$, yields the same condition for an ℓ_1-minimizer.

By considering the definition of the RIP and the theorem above, one could see that it would be advantageous to have almost equal size of the entries in the sensing matrix. In fact it is not difficult to construct a system where the ℓ_0-solution does not coincide with the ℓ_1-solution, if this property is not present. As the following example shows, all one has to do is to stretch some of the columns.

Example 2.6. Consider the underdetermined system

$$\begin{bmatrix} 2 & 1 & 3 \\ 3 & 1 & 2 \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}. \tag{2.2}$$

If we denote the columns of the matrix as $\boldsymbol{a}_1, \boldsymbol{a}_2$ and $\boldsymbol{a}_3$, we can see that the columns $\boldsymbol{a}_1$ and $\boldsymbol{a}_3$ are longer than $\boldsymbol{a}_2$, in an ℓ_2-sense. This means than any linear combination of these two columns yields a smaller ℓ_1-norm than a stretched version of $\boldsymbol{a}_2$. The sparsest solution of this system is, however, the vector $[0\ 2\ 0]^T$. ♣

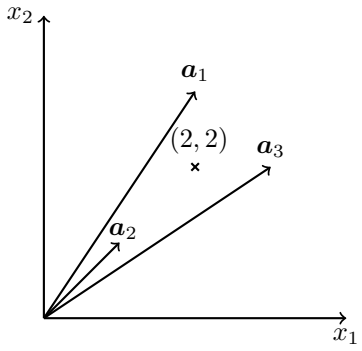


Figure 2.2: The system in equation (2.2). Due to the two long column vectors $\boldsymbol{a}_1$ and $\boldsymbol{a}_3$, the solution of (P_0) and (BP) does not coincide.

In the introduction we learned that most compression algorithms exploit the fact that real life signals can be sparsified by a transform $\boldsymbol{\Psi}$, so that most of the coefficients become close to zero. Such signals are typically called *compressible* rather than sparse, as few of the coefficients are exactly zero. To model such a compressible signal we will add a random perturbation or noise $\boldsymbol{\epsilon} \in \mathbb{C}^m$ to our original model. This gives the equation

$$\mathbf{A}\boldsymbol{x} + \boldsymbol{\epsilon} = \boldsymbol{b}.$$

To make the problem solvable, we shall assume that the amount of perturbations is limited by $||\boldsymbol{\epsilon}||_2 \leqslant \eta$. One can then employ the formulation $(P_{1,\eta})$ to find a sparse solution $\boldsymbol{x} \in \mathbb{C}^N$.

For this relaxed optimization problem, there exists similar results as described above. In these setups one then requires the matrix $\mathbf{A}$ to satisfy the *robust null space property* [21, def. 4.17] or the matrix could have a RIP constant $\delta_{2s} < \frac{4}{\sqrt{41}} \approx 0.62$ [21, thm. 6.12]. For these estimates we are not guaranteed to recover $\boldsymbol{x}$, but we are guaranteed to recover a vector which is close to $\boldsymbol{x}$, which is the best we can do, due to the relaxation. All error bounds in these theorems will therefore involve the *error of best s-term approximation*, denoted as

$$\sigma_s(\boldsymbol{x})_p \coloneqq \inf\{||\boldsymbol{x} - \boldsymbol{z}||_p, ||\boldsymbol{z}||_0 \leqslant s\}$$

This is an important measure of performance, since no s-sparse vector can obtain a better ℓ_p-approximation to $\boldsymbol{x}$, than $\sigma_s(\boldsymbol{x})_p$. Hence, when we evaluate any s-sparse solution $\boldsymbol{x}^\sharp$ of $(P_{1,\eta})$, this will be an excellent lower bound for the error.

2.2 Convex optimization algorithms

The problem found in $(P_{1,\eta})$ is a conic optimization problem, which can be solved using generalised convex optimization software. For the basis pursuit problem, and more generally the quadratically constrained optimization problem, there have, however, been developed several specific algorithms for these problem formulations.

Selected ℓ_1-optimization solvers

Among the greedy algorithms one finds the *orthogonal matching pursuit* (OMP) and *compressive sampling matching pursuit* (CoSaMP). These algorithms work iteratively, and include new entries in the set of support $S = \text{supp}(\boldsymbol{x})$ to the solution vector $\boldsymbol{x}$ at each iteration. The new entries in the set S is at each iteration chosen among the larges entries of the product $\mathbf{A}^*(\mathbf{A}\boldsymbol{x}^n - \boldsymbol{b})$, where $\boldsymbol{x}^n$ denotes the intermediate solution at iteration n. When the new set S have been chosen, one computes a new intermediate solution as

$$\boldsymbol{x}^{n+1} = \text{argmin}\{||\mathbf{A}\boldsymbol{x} - \boldsymbol{b}||_2, \text{supp}(\boldsymbol{x}) \subseteq S\}. \tag{2.3}$$

This process continues until a stopping criterion is met.

Other methods include thresholding-based methods, such as *basic thresholding* (BT), *iterative hard thresholding* (IHT) and *hard thresholding pursuit* (HTP). The simplest of these is the BT algorithm, which chooses the set of support $S = \text{supp}(\boldsymbol{x})$ as the s largest entries of $\mathbf{A}^*\boldsymbol{b}$, and then finds the final solution by solving (2.3). The two other algorithms work iteratively by choosing a new set of support S, as the largest entries of $\boldsymbol{x}^n + \mathbf{A}^*(\boldsymbol{b} - \mathbf{A}\boldsymbol{x}^n)$. The IHT chooses the s largest of these entries as the next intermediate solution $\boldsymbol{x}^{n+1}$, while HTP solves equation (2.3).

All of these algorithms, except the IHT, solves (2.3). This is done by solving the least squares problem seen in the equation, and restricting the solution to have support in S. Hence, to solve this equation we must store the entire matrices in memory. This will make all four of these algorithms unsuitable for large scale calculations.

Another drawback with these algorithms, except the OMP algorithm, is that they require an estimate of the sparsity s. This can be hard to estimate for an unknown signal. The advantage with such an estimate is of course the ability to recover a signal which is truly sparse. By iteratively adding more non-zero entries, or by setting a fixed number of non-zero entries, one can guarantee that the final solution will be sparse. This will not be the case for the SPGL1 algorithm considered in the next subsection.

To explain all of these algorithms in more detail is beyond the scope of this text. Interested readers are refer to [21, Ch. 3] for an easy introduction. For actual implementations of these algorithms, the author has provided this in Python.

Spectral projected gradient ℓ_1 algorithm

Recovery of large signals using convex optimization software will often be constrained by memory limitations on computers. Thus, if the signal is large enough, it will simply be impractical to store the sampling and sparsifying operators as matrices. To overcome this difficulty, we have indicated that we will apply linear operators on $\mathbb{C}^N$ which can be computed in-place using $\mathcal{O}(N \log_2 N)$ operations. One challenge with this approach is that most conic solvers require a densely stored matrix to solve $(P_{1,\eta})$, making it impossible to bypass the memory bottle-neck.

An algorithm which handles these issue is the spectral projected gradient ℓ_1 (SPGL1) algorithm [5, 7]. This algorithm works iteratively, by only using operations which require the matrices $\mathbf{A}$ and $\mathbf{A}^T$ to be linear operators. Hence,

it does not involve any computations of the pseudo-inverse, as required by most of the algorithms above. This enables us to save enormous amounts of memory, by using matrices whose products can be computed in-place.

The SPGL1 algorithm tries to solve the $(P_{1,\eta})$ problem for a fixed $\eta \in [0, ||\boldsymbol{b}||_2]$ by finding a sequence of solutions to an equivalent problem formulation [21, prop. 3.2]. This equivalent formulation is often denoted the *least absolute shrinkage and selection operator* (LASSO), and uses a one-norm constraint. That is

$$\text{minimize } ||\mathbf{A}\boldsymbol{x} - \boldsymbol{b}||_2 \quad \text{subject to} \quad ||\boldsymbol{x}||_1 \leqslant \tau. \tag{LS_τ}$$

As both of these formulations fall into the category of vector optimization problems with no unique optimal value, any solution of either one of them will be Pareto optimal [8, Sec. 4.7.3]. A general problem with these formulations is that it is impossible to know a priori for what values of η and τ these solutions coincide.

Next, let $\boldsymbol{x}_\tau$ be a solution of (LS_τ) and let τ_η denote the value of τ for which the solution of $(P_{1,\eta})$ and (LS_τ) coincide. In [5] it was proved that the optimal solutions of (LS_τ) for all values of $\tau \in [0, \tau_\eta]$ defines a continuously differentiable curve. As all points along this curve is Pareto optimal it is denote as the *Pareto curve*. It was further proven that for $\tau \in [0, \tau_\eta]$ the single parameter function

$$\phi(\tau) = ||\boldsymbol{r}_\tau||_2 \quad \text{with} \quad \boldsymbol{r}_\tau \coloneqq \boldsymbol{b} - \mathbf{A}\boldsymbol{x}_\tau$$

was strictly decreasing. One was also able to find an explicit expression for its derivative using the solution $\boldsymbol{x}_\tau$ of (LS_τ),

$$\phi'(\tau) = -\left|\left|\frac{\mathbf{A}^T\boldsymbol{r}_\tau}{||\boldsymbol{r}_\tau||_2}\right|\right|_\infty.$$

In total, this enable us to use a Newton-based root-finding algorithm to find the final solution $\phi(\tau) = \eta$. Hence, start by setting $\tau_0 = 0$ and solve (LS_τ). This gives us a solution $\boldsymbol{x}_{\tau_0}$. Next, evaluate $\Delta\tau_0 \coloneqq (\eta - \phi(\tau_0))/\phi'(\tau_0)$ for this $\boldsymbol{x}_{\tau_0}$ and compute $\tau_1 = \tau_0 + \Delta\tau_0$. Repeat this process until $\phi(\tau) = \eta$

For this algorithm to work, a critical requirement is to solve the (LS_τ) problem in an efficient manner. This is done by the *spectral projected-gradient* (SPG) algorithm. This algorithm consists of two potentially costly operations. First, one need to implement an efficients projection operator

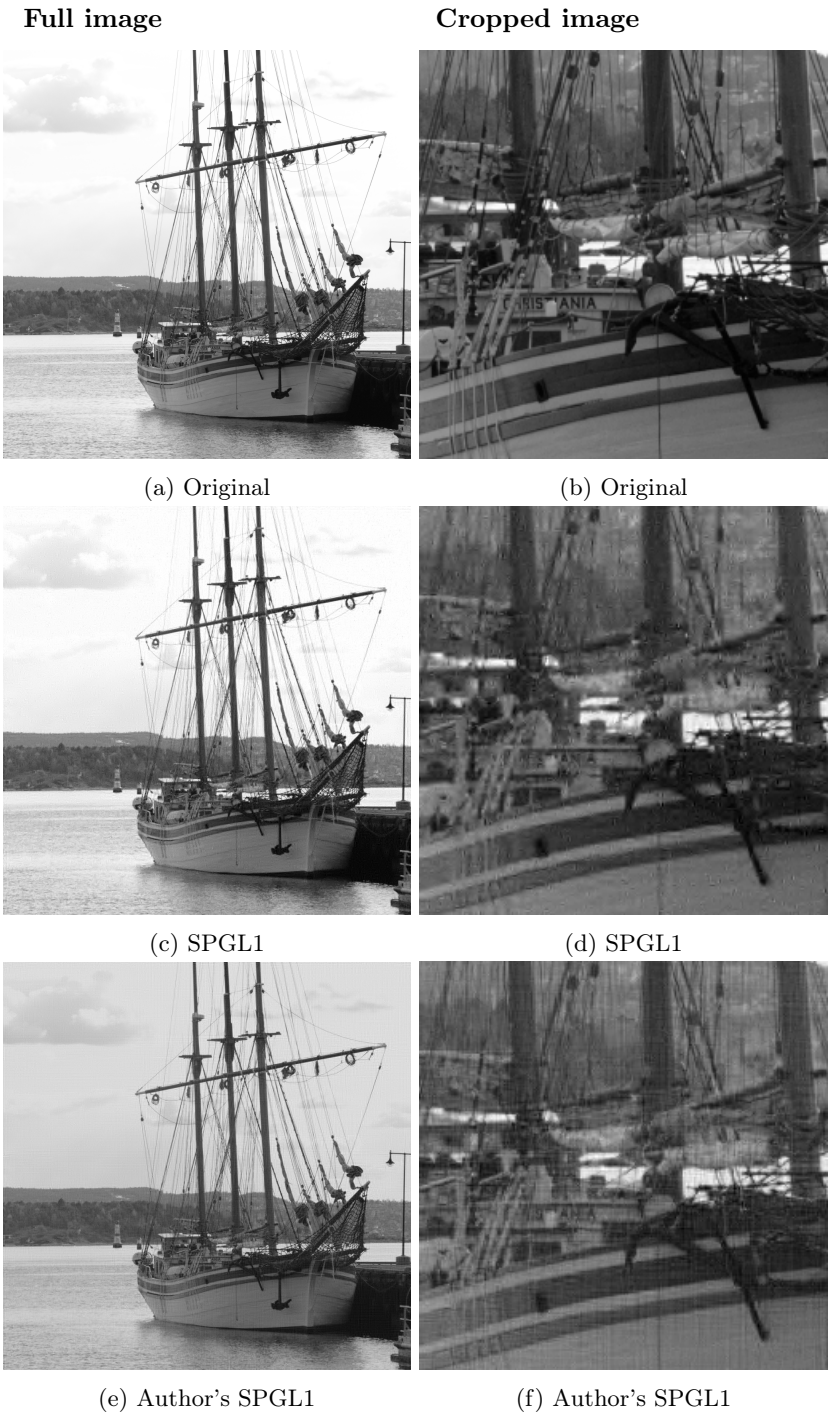
$$P_\tau(\boldsymbol{c}) \coloneqq \left\{ \underset{\boldsymbol{x}}{\operatorname{argmin}} ||\boldsymbol{c} - \boldsymbol{x}||_2 \;\text{ subject to }\; ||\boldsymbol{x}||_1 \leqslant \tau \right\}$$

to project any vector onto the feasible set $\{\boldsymbol{x} : ||\boldsymbol{x}||_1 \leqslant \tau\}$. This can be done using at most $\mathcal{O}(N \log_2 N)$ operations by following an algorithm given in [5].

The second critical step is to compute the gradient of the function $f(x) = \frac{1}{2}||\mathbf{A}\boldsymbol{x} - \boldsymbol{b}||_2^2$ i.e., $\nabla f(\boldsymbol{x}) = \mathbf{A}^*(\mathbf{A}\boldsymbol{x} - \boldsymbol{b})$. The matrix-vector products found in this gradient could potentially have a cost of $\mathcal{O}(N^2)$, but with our use of operators we know that it can be computed using only $\mathcal{O}(N \log_2 N)$ operations. As a result, both of the potentially expensive operations can be computed efficiently.

The idea of the SPG algorithm is to start with an initial solution $\boldsymbol{x}_0$ and search for a new solution along the gradient $\nabla f(\boldsymbol{x}_0)$. This is done by choosing a step length α and compute a new solution $\boldsymbol{x}_1 = P_\tau(\boldsymbol{x}_0 - \alpha \nabla f(x_0))$. This process is then repeated until some stopping criterion is met.

There are more technicalities concerning this algorithm, which have been omitted here. Interested readers are referred to [5] for the mathematical details, while the authors C++ implementation can be found online for technical details. In figure 2.3 we have compared this implementation with the default MATLAB package [6]. As we can see from the figure, both of these implementations yield sufficient results, but the computational time of the former is higher. All other figures in this text have therefore been computed with the MATLAB algorithm.



(a) Original (b) Original

(c) SPGL1 (d) SPGL1

(e) Author's SPGL1 (f) Author's SPGL1

Figure 2.3: The images shows reconstruction using Hadamard sampling and Daubechies wavelets with various SPGL1 algorithms. (Top) The original images. (Middle) The reconstruction using MATLAB's SPGL1 algorithm. (Bottom) The reconstruction using the author's SPGL1 implementation.

CHAPTER 3

Wavelets

Compressive sensing rely on the assumption of sparsity to solve an underdetermined system of linear equations. To impose the required sparsity on the signal $\boldsymbol{x} \in \mathbb{C}^N$, there have been proposed several sparsifying operators $\boldsymbol{\Psi}$ on $\mathbb{C}^N$. Typical choices could be the curvelet transform [29, Sec. 5.5.2] for two dimensional signals, the cosine transform for smooth real signals or total variation transform [29, Sec. 2.3.3], if an inverse transform in not necessary.

In this text we have chosen to study the wavelet transform, since this transform can be defined as an unitary operator and computed in $\mathcal{O}(N)$ operations. In addition, it has an excellent sparsifying effect on most signals. This is one of the reasons for its popularity in image compression [36]. Due to the unitary requirement of the current theory within compressive sensing, our main focus in this chapter will be on Daubechies' compactly supported orthonormal wavelets.

3.1 Multiresolution analysis

From Fourier theory we know that any function $f \in \mathcal{L}^1([0,1])$ can be represented in the Fourier basis $\{e^{2\pi ikx}\}_{k\in\mathbb{Z}}$ through the Fourier series

$$f(x) = \sum_{k=-\infty}^{\infty} \widehat{f}(k) e^{2\pi ikx}$$

with

$$\widehat{f}(k) = \left\langle f(x), e^{2\pi ikx} \right\rangle = \int_0^1 f(x) e^{-2\pi ikx} \, \mathrm{d}x$$

provided that $\widehat{f} \in \ell^1(\mathbb{Z})$ [30, Thm. 15.2]. This gives a signal representation of f in the *frequency domain*, by using the complex exponentials as basis functions. Each of these exponentials have a frequency k and support on all of $[0,1]$. Hence, if f consists of any irregularity, this will cause all of the coefficients $\widehat{f}(k)$ to have a slow decay. It will also be impossible to localize any irregularity, since all the exponentials have support on $[0,1]$.

Wavelets, on the other hand, which we will define on $\mathbb{R}$ rather than $[0,1]$, will have a narrow support in both the time and frequency domain. This is conducted by replacing the complex exponentials $e^{2\pi ikx}$ found in the Fourier basis, by a scaling function $\phi\colon \mathbb{R} \to \mathbb{R}$, in the wavelet basis. By translating and dilating the function ϕ we will create an orthonormal basis for $\mathcal{L}^2(\mathbb{R})$ where

not all the basis functions have the same support. Hence if we define

$$\phi_{j,k} := 2^{j/2}\phi(2^j x - k)$$

we are seeking a orthonormal basis of $\mathcal{L}^2(\mathbb{R})$ on the form $\{\phi_{j,k}\}_{(j,k)\in\mathbb{Z}^2}$. To create such a basis we need to construct ϕ such that it satisfies the conditions of a *multiresolution analysis* (MRA).

Definition 3.1 (Multiresolution [27, 29])**.** A sequence $\{V_j\}_{j\in\mathbb{Z}}$ of closed subspaces of $\mathcal{L}^2(\mathbb{R})$ is a *multiresolution analysis* if the following five properties are satisfied

$$\begin{aligned}
&V_j \subset V_{j+1} && \text{for all } j \in \mathbb{Z};\\
&f \in V_j \text{ if and only if } f(2(\cdot)) \in V_{j+1} && \text{for all } j \in \mathbb{Z};\\
&\bigcap_{j\in\mathbb{Z}} V_j = \{0\};\\
&\text{Closure}\left(\bigcup_{j\in\mathbb{Z}} V_j\right) = \mathcal{L}^2(\mathbb{R});
\end{aligned}$$

and there exists a $\phi \in V_0$ such that $\{\phi(x-k) : k \in \mathbb{Z}\}$ is an orthonormal basis of V_0.

For any set of functions $V_j \subset V_{j+1}$, there exists a nonempty orthogonal set of functions $W_j = V_{j+1} \perp V_j$, usually denoted as the *detail space* of V_{j+1}. This could alternatively be written as

$$V_j \oplus W_j = V_{j+1}$$

where $\oplus$ denotes the direct sum of the two spaces. As $V_j \to \{0\}$ when $j \to -\infty$ one can apply this relation recursively and obtain

$$V_{j+1} = V_j \oplus W_j = \bigoplus_{\ell=-\infty}^{j} W_\ell. \tag{3.1}$$

Similarly we can apply this relation when $j \to \infty$ as $V_j \to \mathcal{L}^2(\mathbb{R})$, thus

$$\mathcal{L}^2(\mathbb{R}) = \bigoplus_{j=-\infty}^{\infty} W_j. \tag{3.2}$$

Hence, if we find another function $\psi \in W_0$ so that $\{\psi(x-k) : k \in \mathbb{Z}\}$ is an orthonormal basis for W_0 then, due to equation (3.1) and (3.2) W_j constitutes a multiresolution analysis of $\mathcal{L}^2(\mathbb{R})$. As $V_0 \subset V_1$ we know that $\phi(x)$ can be written as a linear combination of $\sqrt{2}\phi(2x-k)$. That is

$$\phi(x) = \sum_{k=-\infty}^{\infty} h[k]\sqrt{2}\phi(2x-k) \tag{3.3}$$

with

$$h[k] = \left\langle \phi(x), \sqrt{2}\phi(2x-k) \right\rangle = \sqrt{2}\int_{-\infty}^{\infty} \phi(x)\phi^*(2x-k)\,\mathrm{d}x,$$

where ϕ^* denotes the complex conjugate of ϕ. As the function $\phi \in \mathcal{L}^2(\mathbb{R})$ we know that the convergence in (3.3) is in $\mathcal{L}^2(\mathbb{R})$ and $\sum_{k\in\mathbb{Z}} |h[k]|^2 < \infty$. Taking the Fourier transform of the above equation, we obtain

$$\widehat{\phi}(\omega) = \frac{1}{\sqrt{2}}\widehat{\phi}\left(\tfrac{\omega}{2}\right) \sum_{k=-\infty}^{\infty} h[k]e^{2\pi ik\omega} = \frac{1}{\sqrt{2}}\widehat{\phi}\left(\tfrac{\omega}{2}\right)\widehat{h}(\omega) \tag{3.4}$$

where

$$\widehat{h}(\omega) = \sum_{k=-\infty}^{\infty} h[k]e^{2\pi ik\omega}$$

is denoted as the *conjugate mirror filter* of the scaling function ϕ, with filter coefficients $\{h[k]\}_{k\in\mathbb{Z}}$ [27, p. 53]. As we shall see from the following theorem, this filter h is a *low-pass* filter due to the equality $\widehat{h}\left(\frac{1}{2}\right) = 0$.

Theorem 3.2 ([29, thm. 7.2])
Let $\phi \in \mathcal{L}^2(\mathbb{R})$ be an integrable scaling function. The Fourier series of $h[k] = \left\langle \phi(x), \sqrt{2}\phi(2x-k)\right\rangle$ satisfies for all $\omega \in \mathbb{R}$

$$\left|\widehat{h}(\omega)\right|^2 + \left|\widehat{h}\left(\omega + \tfrac{1}{2}\right)\right|^2 = 2$$

and

$$\widehat{h}(0) = \sqrt{2}.$$

Remark 3.3. In the theorem above, none of the equations are equal to 1. In the literature, however, both of these equations are often normalized. To obtain this effect, one would need to remove the factor $\sqrt{2}$ in equation (3.3). The derivation of an MRA would have been the same, but the factors would be different. Both forms are therefore found in the literature. As a result, we refer to the coherence at frequency $\omega = 0$ as $\mathcal{O}(1)$, meaning that it is equal to 1 or a small constant factor away from 1.

As $W_0 \subset V_1$, the same derivation can be made for ψ as we did for ϕ above. One then finds the following relation between ϕ and ψ

Theorem 3.4 ([29, thm. 7.3])
Let ϕ be a scaling function and h the corresponding conjugate mirror filter. Let ψ be the function having a Fourier transform

$$\widehat{\psi}(2\omega) = \frac{1}{\sqrt{2}}\widehat{\phi}(\omega)\widehat{g}\,(\omega)$$

with

$$\widehat{g}(\omega) := e^{-2\pi i\omega}\widehat{h}^*\left(\omega + \tfrac{1}{2}\right).$$

Let us denote

$$\psi_{j,k} := 2^{j/2}\psi\left(2^j x - k\right).$$

For any scale 2^j, $\{\psi_{j,k}\}_{k\in\mathbb{Z}}$ is an orthonormal basis of W_j. For all scales $\{\psi_{j,k}\}_{(j,k)\in\mathbb{Z}^2}$ is an orthonormal basis for $\mathcal{L}^2(\mathbb{R})$.

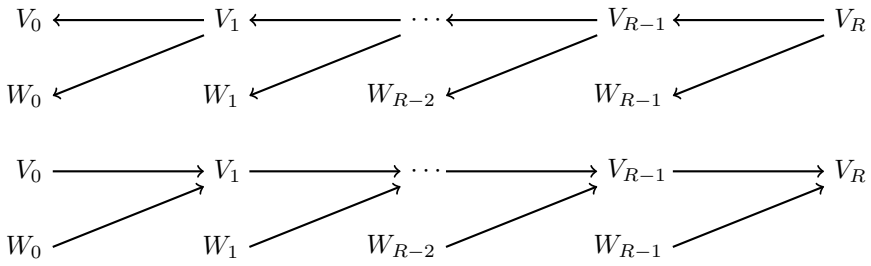


Figure 3.1: Above: The wavelet transform of a function $f \in V_R$. Below: The inverse wavelet transform of a function $f \in V_R$

3.2 Vanishing moments

The idea of the discrete wavelet transform is to assume the that signal $f \in V_R$, $R > 0$ and decompose f into the low resolution space V_0 and the detail spaces W_j, for $0 \leqslant j < R$. This is possible since

$$V_R = V_0 \bigoplus_{j=0}^{R-1} W_j.$$

In order to obtain a more compressible representation of f in this new domain, we need to construct the spaces $\{W_j\}_{0 \leqslant j < R}$ such that most of the coefficients $\langle f, \psi_{j,k} \rangle \approx 0$. To obtain this effect, one usually requires ψ to have a certain number of vanishing moments ν. That is, ψ needs to be orthogonal to all polynomials of degree less than ν.

Definition 3.5 (Vanishing moments)**.** A wavelet ψ has ν vanishing moments if

$$\int_{-\infty}^{\infty} x^k \psi(x)\, \mathrm{d}x = 0 \quad \text{for } 0 \leqslant k < \nu.$$

The magnitude of the inner product $\langle f, \psi_{j,k} \rangle$ will depend on the smoothness of f, in addition to the support and number of vanishing moments of ψ. A smooth signal can be approximated well by a lower degree polynomial, which implies that the inner product becomes close to zero. If the support of ψ is large, any singularity in f would create large amplitude coefficients for many of the $\psi_{j,k}$ functions. We would therefore like to minimize the support of ψ, while maximizing its number of vanishing moments.

The above argument can be formalized by specifying the regularity – which will be defined later – of f and the support of $\psi_{j,k}$. Interested readers are referred to [29, thm. 6.3]

According to theorem 7.4 – 7.9 in [29] one can derive that any orthonormal wavelet ψ has ν vanishing moments if and only if $\widehat{\psi}(\omega)$ and its first $\nu - 1$ derivatives are zero at $\omega = 0$. Further, we know from these theorems that if the scaling function ϕ has compact support, then so does h, and their support is equal. A similar results holds for ψ.

The orthonormal wavelet which minimizes the support of ψ for a given number of vanishing moments ν, are called *Daubechies wavelets*. We summarize this as a theorem.

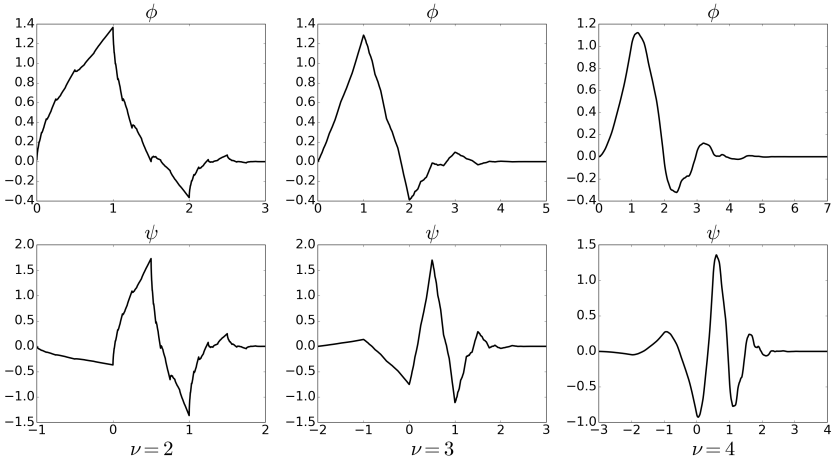


Figure 3.2: The Daubechies scaling function and wavelet for three different vanishing moments.

Theorem 3.6 ([29, pp. 293 – 294])
If ψ is a wavelet with ν vanishing moments, that generates an orthonormal basis for $\mathcal{L}^2(\mathbb{R})$, then its support size is larger than or equal to $2\nu - 1$. A Daubechies wavelet has a minimum-size support equal to $[-\nu + 1, \nu]$. The support of the corresponding scaling functions ϕ is $[0, 2\nu - 1]$. The filters h and g corresponding to these functions both have 2ν filter coefficients.

The Daubechies scaling function ϕ is defined through the conjugate mirror filter h in equation (3.3). To create an MRA this filter h must satisfy the conditions of theorem 3.2. Further $\widehat{h}(\omega)$ and its $\nu - 1$ first derivative must equal zero at $\omega = 0$, to achieve the required number of vanishing moments. That is

$$\widehat{h}(\omega) = \sqrt{2} \left(\frac{1 - e^{-2\pi i \omega}}{2} \right)^{\nu} R(e^{-2\pi i \omega}) \tag{3.5}$$

where $R(e^{-2\pi i \omega})$ is some polynomial. To obtain a minimum support of ϕ, one needs to minimize the degree of R under the constraint of

$$|\widehat{h}(\omega)|^2 + \left| \widehat{h}\left(\omega + \tfrac{1}{2}\right) \right|^2 = 2.$$

One can then show using Bezout's theorem [14, Thm. 6.1.1] that the minimum degree of this polynomial is $\nu - 1$, and use a computer to calculate its roots. Due to this construction, one usually finds the filter coefficients h to the Daubechies scaling function in lookup-tables. For a table of all the coefficients up to 10 vanishing moments, we refer to [14, p. 195].

3.3 The discrete wavelet transform

The cascade algorithm

The Daubechies wavelets are, as we have seen above, created in the Fourier domain, with filter coefficients which are computed to machine precision. The

functions ϕ and ψ do therefore not have a closed analytic form. To compute any of their function values, one have to apply the cascade algorithm to perform an inverse discrete wavelet transform.

To use this algorithm, we start by finding the filter coefficients of the filter g, defined in theorem 3.4. From this theorem we know that

$$\begin{aligned}\widehat{g}(\omega) &= e^{-2\pi i\omega}\widehat{h}^*\left(\omega+\tfrac{1}{2}\right)\\ \widehat{g}(\omega) &= e^{-2\pi i\omega}\widehat{h}\left(-\omega+\tfrac{1}{2}\right)\\ \widehat{g}(\omega) &= \widehat{h}\left(-(\omega+1)+\tfrac{1}{2}\right).\end{aligned}$$

By converting this to the time domain, one obtains the following formula for the g-coefficients,

$$g[k] = (-1)^{1-k}h[1-k].$$

To use these coefficients in an MRA we recall that $V_{j-1} \subset V_j$ and that

$$\phi_{j-1,k} = \sum_{n=-\infty}^{\infty} \langle \phi_{j-1,k}, \phi_{j,n}\rangle \phi_{j,n}.$$

By using the change of variable $x' = 2^j x - 2k$ we obtain

$$\begin{aligned}\langle \phi_{j-1,k}, \phi_{j,n}\rangle &= \int_{-\infty}^{\infty} 2^{j-1/2}\phi\left(2^{j-1}x-k\right)\phi^*\left(2^j x-n\right)\,\mathrm{d}x\\ &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2}}\phi\left(\frac{x}{2}\right)\phi^*\left(x-n+2k\right)\,\mathrm{d}x\\ &= h[n-2k]\end{aligned}$$

where the same derivation can be made for $\psi_{j,k}$. By using vector notation for the inner products

$$a_j[k] = \langle f, \phi_{j,k}\rangle, \qquad d_j[k] = \langle f, \psi_{j,k}\rangle \qquad \text{for all } k \in \mathbb{Z} \tag{3.6}$$

and the convention $\bar{h}[k] = h[-k]$, one can derive the following theorem, known as the *cascade algorithm*.

Theorem 3.7 ([29, thm. 7.10])
A function $f \in V_j$ can be decomposed into V_{j-1} and W_{j-1} using

$$\begin{aligned}a_{j-1}[k] &= \sum_{n=-\infty}^{\infty} h[n-2k]a_j[n] = a_j * \bar{h}[2k]\\ d_{j-1}[k] &= \sum_{n=-\infty}^{\infty} g[n-2k]a_j[n] = a_j * \bar{g}[2k]\end{aligned}$$

where $*$ denotes the convolution of the two sequences. Similarly one can convert f back to its original domain V_j using coefficients from the two spaces V_{j-1} and W_{j-1}. That is

$$a_j[k] = \sum_{n=-\infty}^{\infty} h[k-2n]a_{j-1}[n] + \sum_{n=-\infty}^{\infty} g[k-2n]d_{j-1}[n]$$

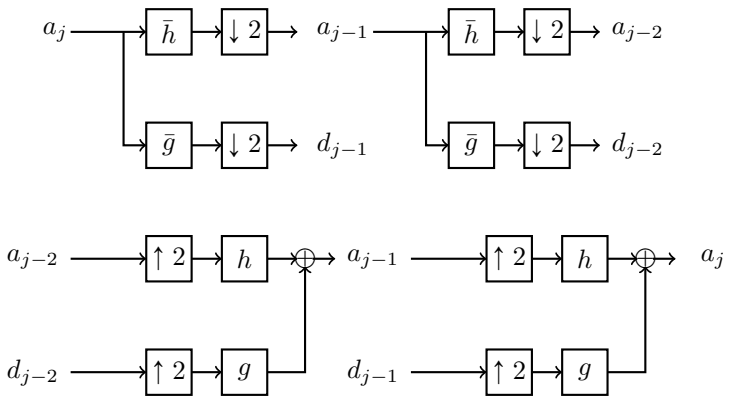


Figure 3.3: Illustration of composition and decomposition found in theorem 3.7. The notation "$\uparrow 2$" means upsampling by a factor 2 i.e., one insert a zero between all samples. Similarly the notation "$\downarrow 2$" means downsampling by a factor 2 i.e., remove every second sample.

By applying the above theorem to a function $f \in V_R$, $R > 0$, we are able represent f in both of the vector spaces V_R and $V_{R-1} \oplus W_{R-1}$. By decomposing the vector space V_{R-1} further, we can obtain a representation of f in the vector space $V_0 \oplus_{j=0}^{R-1} W_j$. The change of coordinates from the vector space V_R to $V_0 \oplus_{j=0}^{R-1} W_j$ is known as the *discrete wavelet transform* (DWT) of f. Similarly the *inverse discrete wavelet transform* (IDWT) is the change of coordinates from the vector space $V_0 \oplus_{j=0}^{R-1} W_j$ back to V_R.

Computational complexity

From theorem 3.6 we know that the filters h and g both have length $K = 2\nu$. One step with the cascade algorithm would therefore require approximately $2KN$ floating-point operations. As the next iteration only applies this transform to half of the remaining coefficients there are approximately $2KN/2$ flops at this iteration. Continuing in this manner one obtains the sequence

$$2KN + \frac{2KN}{2} + \frac{2KN}{2^2} + \cdots + \frac{2KN}{2^{R-1}} \approx 4KN$$

where we assume $K \ll N$. This gives an overall complexity of $\mathcal{O}(N)$.

The wavelet crime

In any application of the wavelet transform we do never have a function in $f \in \mathcal{L}^2(\mathbb{R})$, but rather a function $f \in \mathcal{L}^2([0,1])$, where we for simplicity have normalized any interval $[a,b]$ into $[0,1]$. This function is usually observed using N equispaced samples. For our purposes we shall always assume $N = 2^R$, since the Hadamard transform is only defined for signals of this length. A welcome side effect of this choice, is that the DWT becomes particularly simple for signals of this length. In the following we will drop any concerns regarding the boundary of $[0,1]$, as these will be dealt with in the next subsection.

To compute the DWT of a signal $f \in \mathcal{L}^2([0,1])$, we start by assuming that the samples of f are uniformly distributed. Any sample of f can then be written as $f(k/N)$, $k = 0, \ldots, N-1$. We shall assume that all of these samples $f(k/N)$ are approximately equal to $2^{R/2}\langle f, \phi_{R,k}\rangle$.

To justify this assumption, notice that the scaling function is orthonormal. We can therefore interpret the inner product

$$2^{R/2}\langle f, \phi_{R,k}\rangle = \int_0^1 f(x)2^R\phi(2^R x - k)\,\mathrm{d}x$$

as a weighted average of f around k/N. Thus the approximation

$$f(k/N) \approx 2^{R/2}\langle f, \phi_{R,k}\rangle \tag{3.7}$$

is fair for smooth functions f [29, p. 301]. As the constant $2^{R/2}$ is the same for all samples, we usually omit it. The approximation found in equation (3.7) is the standard way of approximating discrete signals, and it is usually referred to as the *wavelet crime* [33] due to its inaccuracy. The approximation (3.7) can however cause large errors if the samples does not match the corresponding inner products.

Boundary wavelets

One of the problems with the model above is of course that the scaling function $\phi_{R,k}$ does not have support on $[0,1]$ for all k. This problem can be addressed by several different approaches, three of which are presented here.

Periodic extension The simplest approach is to extend f periodically to $\mathbb{R}$ with period 1. In any case where $f(0) \neq f(1)$ this will cause the extended function to be discontinuous at the boundary points. As polynomials always give a poor approximation to any discontinuous function, this will remove all vanishing moments around the discontinuity. Thus, any wavelet coefficient whose wavelet have support at one of the endpoints, will have a large magnitude. This removes any compressibility among all of these coefficients. The advantage with this method is, of course, that it is particularly simple to implement. One simply replace the convolution operator found in theorem 3.7 by a circular convolution operator.

Extension by folding The second approach is to fold the function f around the point $x = 0$, by extending the support of f to $[-1,1]$ and setting $f(x) = f(-x)$. This signal is then extended periodically to $\mathbb{R}$. Using this method the folded function obtains the desired continuity $f(-1) = f(1)$. The only problem is that we can not not assume $f'(0) = f'(1) = 0$. Hence, even if f is continuously derivative on $[0,1]$, its folded extension will have a discontinuous first derivative when it is extended to $\mathbb{R}$. This approach will therefore only preserve one vanishing moment.

Constructing boundary wavelets The last and most demanding approach is to construct a new orthonormal wavelet basis on the interval $[0,1]$. This is done by first translating the scaling function ϕ such that it has support

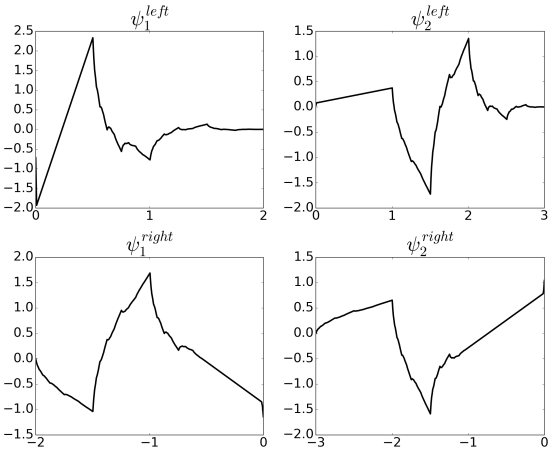


Figure 3.4: The Daubechies boundary wavelets with $\nu = 2$ vanishing moments.

$[-\nu + 1, \nu]$. One then choose a j_0 such that $2\nu \leqslant 2^{j_0}$. This ensures that there are $2^{j_0} - 2\nu$ scaling functions with support inside $(0, 1)$. Next, one constructs ν orthonormal *boundary scaling functions* with support inside $[0, 1]$ on both sides of this interval.

Using this approach one construct a new space we shall denote V_j^{int} for $j \geqslant j_0$, consisting of the functions

$$\begin{aligned}
\phi_{j,k}^{\text{int}}(x) &= 2^{j/2}\phi_k^{\text{left}}(2^j x) && \text{for } 0 \leqslant k < \nu; \\
\phi_{j,k}^{\text{int}}(x) &= 2^{j/2}\phi(2^j x - k) && \text{for } \nu \leqslant k < 2^j - \nu; \\
\phi_{j,k}^{\text{int}}(x) &= 2^{j/2}\phi_k^{\text{right}}(2^j x) && \text{for } 2^j - \nu \leqslant k < 2^j.
\end{aligned}$$

Each of these new boundary functions are created as solutions of the corresponding scaling equation

$$\frac{1}{\sqrt{2}}\phi_n^{\text{left}}\left(\tfrac{x}{2}\right) = \sum_{k=0}^{\nu-1} H_{n,k}^{\text{left}}\phi_k^{\text{left}}(x) + \sum_{k=\nu}^{\nu+2n} h_{n,k}^{\text{left}}\phi(x-k) \tag{3.8}$$

and similarly for ϕ_k^{right}, ψ_k^{left} and ψ_k^{right} [29, Sec. 7.5.3]. Hence, to decompose a function $f \in V_R^{\text{int}}$ into $V_{R-1}^{\text{int}} \oplus W_{R-1}^{\text{int}}$ one can then be perform the usual cascade algorithm to interior samples, while one applies equations similar to (3.8) for the boundary samples. For the corresponding inverse equations we refer to [29, thm. 7.19].

Pre- and postconditioning By using the boundary wavelet algorithm described above, one would expect that a one level DWT of the vector $\begin{bmatrix}1 & 1 & \dots & 1\end{bmatrix}^T$ would result in a vector where the $N/2$ last elements are 0. This is, however, not the case. To obtain such an effect, one need to impose some *pre- and postconditions* on the corresponding input and output vectors. Next, we will shortly explain these concepts.

Recall that we in equation (3.6) introduced the short hand notation $a_j[k] = \langle f, \phi_{j,k}\rangle$ and $d_j[k] = \langle f, \psi_{j,k}\rangle$. Let $\pi_{\nu-1}$ denote the set of polynomials of degree less than or equal to $\nu-1$. The following discussion considers wavelets in $\mathcal{L}^2(\mathbb{R})$.

If ψ is a wavelet with ν vanishing moments we know that $\int_{-\infty}^{\infty} x^k \psi(x)\,\mathrm{d}x = 0$ for $k = 0, \ldots, \nu - 1$. This implies that $\phi \in \pi_{\nu-1}$ as ϕ is orthogonal to ψ. Hence, if $p \in \pi_{\nu-1}$ we know that $a_0[k]$ can be written as a linear combination of the monomial basis. That is

$$a_0[k] = \int_{-\infty}^{\infty} p(x)\phi(x-k)\,\mathrm{d}x = \sum_{i=0}^{\nu-1} \alpha_i k^i.$$

Thus, if $a_0[k]$ is some polynomial sequence, say for simplicity $a_0[k] = 1$ for all $k \in \mathbb{Z}$, $\nu \geqslant 1$, this implies that $p(x) = 1$ and $d_{-1}[k] = 0$ for all $k \in \mathbb{Z}$.

On the interval $[0, 1]$ the $d_{-1}[k] = 0$ equality does not hold for the boundary functions because the integrals

$$\int_{-\infty}^{\infty} \phi_k^{\text{right}}\,\mathrm{d}x \neq 1 \quad \text{and} \quad \int_{-\infty}^{\infty} \phi_k^{\text{left}}\,\mathrm{d}x \neq 1,$$

while $\int_{-\infty}^{\infty} \phi_{0,k}(x)\,\mathrm{d}x = 1$. Hence, if we would like the DWT of vectors like

$$\begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}^T \quad \text{or} \quad \begin{bmatrix} 1 & 2 & \ldots & N \end{bmatrix}^T$$

to have boundary wavelet coefficients which equal 0, one must multiply the endpoints of the vectors with some precondition matrices $\mathbf{A}_{\text{left}} \in \mathbb{R}^{\nu\times\nu}$ and $\mathbf{A}_{\text{right}} \in \mathbb{R}^{\nu\times\nu}$ before one apply the DWT. This will account for the boundary scaling functions the lack of orthonormality. Similarly, one needs to multiply with the postconditioning matrices $\mathbf{A}_{\text{left}}^{-1}$ and $\mathbf{A}_{\text{right}}^{-1}$ after one have applied the IDWT. [12].

Note the boundary wavelet matrix with no pre- or postconditioning will be a unitary matrix. The corresponding boundary wavelet matrix with this preconditioning included as a part of the operator will, on the other hand, not be unitary, since $\mathbf{A}^{-1} \neq \mathbf{A}^T$. In any calculations involving boundary wavelets found in this text, these pre- and postconditions have therefore not been included as part of the operator.

3.4 Wavelet regularity

As there does not exists any analytical expression for the functions ψ and ϕ, it is difficult to state if these functions are continuous or if the derivative exits for all points. A first glance of these functions in figure 3.2 reveal that they might be continuous. To verify this analytically, we need to explore the *Lipschitz regularity* of the functions.

Definition 3.8 ([29, def. 6.1])**.**

- A function f is pointwise Lipschitz $\alpha \geqslant 0$ at s if there exists $K > 0$ and a polynomial p_s of degree $n = \lfloor \alpha \rfloor$ such that

$$|f(x) - p_s(x)| \leqslant K|x - s|^\alpha. \tag{3.9}$$

- A function f is uniformly Lipschitz α over $[a, b]$ if it satisfies (3.9) for all $x \in [a, b]$ with a constant K that is independent of x.

- The Lipschitz regularity of f at s or over $[a, b]$ is the supremum of the α such that f is Lipschitz α.

It is an easy exercise to show that any function f which is uniformly Lipschitz $\alpha > 0$ is continuous. Similarly for $\alpha \geqslant 1$, f is $\lfloor \alpha \rfloor$ times continuously differentiable. To estimate the regularity of the Daubechies scaling function ϕ and wavelet ψ, one may use the Fourier transform formulation of this regularity. It says that a function f is bounded and uniformly Lipschitz α over $\mathbb{R}$ if

$$\int_{\mathbb{R}} |\widehat{f}(\omega)|(1 + |\omega|^{\alpha}) \, \mathrm{d}\omega < \infty.$$

By using the recurrence relation in (3.4) and the expression for $\widehat{h}$ in (3.5) one can estimate these α's with various accuracy. As ϕ and ψ are both linear combinations of $\phi(2x-k)$ they obtain the same Lipschitz regularity. In Daubechies work [14, Ch. 7] she describes several of these approaches to estimate the α's. The best of these estimates are seen in table 3.1. Further, we know that for large ν, α increases approximately as 0.2ν [12]. This means that for $\nu \geqslant 3$ the functions ϕ and ψ are continuously differentiable.

ν	α
2	0.55
3	1.08
4	1.61

Table 3.1: The Lipschitz constant α for ν number of vanishing moments for the Daubechies wavelets.

3.5 Numerical implementation

In any numerical algorithm, one would always like to reduce the number of operation as much as possible. For the orthonormal wavelet transform, this can be done by rewriting the cascade algorithm using elementary *lifting* operations [29, sec. 7.8]. Using these liftings, one can reduce the number of operations by a factor of 2 and reduce the memory consumption by performing all operations in-place. Thus, when one choose a modern wavelet library, it is this algorithm one would find under the hood. This is also the case for the wavelet library provided by Ryan, which is used extensively for all computations related to this text.

The lifting algorithm makes it particularly simple to implement the periodic or folded wavelets to handle the boundary problem. Most wavelet libraries therefore implements these two extensions, and omit the harder approach of constructing dedicated boundary wavelets. In fact, a thorough search for orthonormal wavelet libraries, showed that most libraries simply omit any comments about boundary handling, or implements the periodic or folded approach.

To construct a DWT with boundary wavelets, one would necessarily need the boundary wavelet coefficients. In [12] Cohen, Daubechies and Vial provide a FTP connection where one can retrieve these coefficients. This connection did naturally turned out to be unresponsive when the author tested it 23 years

after the paper was first published. A quick google search for these coefficients did also turned out empty.

Eventually the author was able to find one wavelet library where the boundary wavelets had been implemented. This was found as a part of the *Wavelab* package [16] for MATLAB. Unfortunately this package only support 2 and 3 vanishing moments for this particular boundary handling. The package does also only support the boundary wavelet decomposition into the space V_3 rather than V_2, which should be possible for $\nu = 2$ vanishing moments. It is also worth noting that due to the boundary wavelet functions use of the cascade algorithm, rather than a lifting factorization, results in a notable round off error.

In addition, for the case with $\nu = 2$, the left sided pre- and postcondition matrices were wrong. The author therefore had to debug the DWT and IDWT functions and provide a fix to make them work correctly. The author have tried to file a bug report on the issue, but the Wavelab mail address turned out to be dead, so nothing have been reported. For a thorough bug report, see the repository with the code related to this thesis. In total this suggest that boundary wavelets are not very common in applications.

Most recently the author have also found an extension of the Wavelab package provided by Clarice Poon and Milana Gataric at http://www.damtp.cam.ac.uk/research/afha/code/gs_wavelets-1.1.zip. In this extension one have rewritten the function responsible for providing the boundary wavelet coefficients in the Wavelab package, so that one can retrieve the coefficients up to 10 vanishing moments. Due to the late discovery of this extension, it has not been used in any of our computations.

In this text we have applied the periodic extensions of the wavelet basis in all figures involving wavelets. The only exception is a few figures found in chapter 6, where it is explicitly stated that the boundary wavelets found in the Wavelab package have been used. This approach have been chosen both for the numerical accuracy provided by the lifting factorization and for simplicity, as Ryan's wavelet library provided a unified interface to both MATLAB and Python. For any computations made in C++, the author's own wavelet library was preferred.

CHAPTER 4

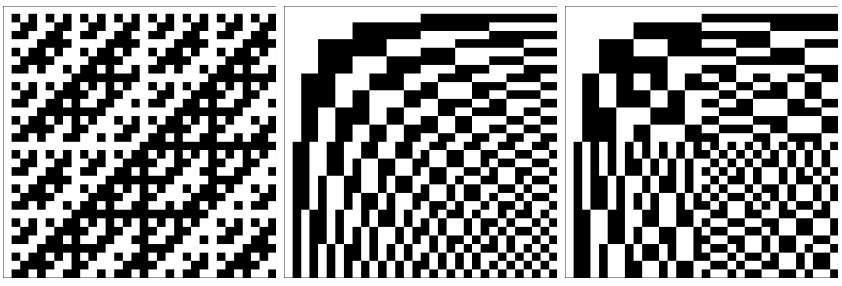
Hadamard transform

In chapter 1, we saw the need for unitary matrices whose matrix product could be computed in-place using $\mathcal{O}(N \log_2 N)$ operations. One such type of matrix is the orthogonal Hadamard matrix, whose entries consists of either 1 or -1. To create an orthogonal matrix with these entries for any N is, in general, impossible, just consider any odd number N. The Hadamard matrix is therefore only defined for $N = 2^R$ for some positive integer R.

For the size $N = 2^R$, there are at least three different ways to construct an orthogonal matrix with the entries -1 and 1, all of which are referred to as the Hadamard matrix. The set of rows in these matrices are necessarily the same, but the ordering of the rows are different. To distinguish between them we shall use the usual convention of denoting them as the *ordinary*, *sequency* and *Paley* enumerated Hadamard matrix.

4.1 The ordinary Hadamard matrix

The ordinary Hadamard matrix is, as the name suggests, the ordering of the matrix rows which is often found in sources where only one of the orderings is mentioned. It is therefore often also referred to as the *Hadamard order*. The reason for its popularity is probably due to its simple definition through the



(a) (b) (c)

Figure 4.1: The three different Hadamard matrices for $N = 32$. (a) Ordinary order, (b) Sequency order, (c) Paley order. The color white represents 1's, while black represents -1's.

recurrence relation

$$\mathbf{H}_N = \mathbf{H}_2 \otimes \mathbf{H}_{N/2} = \begin{bmatrix} \mathbf{H}_{N/2} & \mathbf{H}_{N/2} \\ \mathbf{H}_{N/2} & -\mathbf{H}_{N/2} \end{bmatrix} \tag{4.1}$$

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{4.2}$$

where $\otimes$ denotes the Kronecker product. As this relation has a close connection to the fast Fourier transform , it is often proposed as a poor man's FFT.

Remark 4.1. The Hadamard matrix as we have defined it in equation (4.1) and (4.2) is orthogonal and not orthonormal. We will, however, often state that all three of the Hadamard matrices are unitary. It is then implicitly understood that one scale $\mathbf{H}_N$ by $1/\sqrt{N}$. The same effect could also be obtained by scaling the $\mathbf{H}_2$ matrix in (4.2) by $1/\sqrt{2}$.

Computational complexity In general, a matrix multiplication would require $\mathcal{O}(N^2)$ floating point operations. For the ordinary ordered Hadamard matrix, this amount can be reduced to $\mathcal{O}(N \log_2 N)$. To see this, note that we can decompose the $N \times N$ matrix product

$$\mathbf{H}_N \boldsymbol{x} = \begin{bmatrix} \mathbf{H}_{N/2}\boldsymbol{x}_{\text{top}} + \mathbf{H}_{N/2}\boldsymbol{x}_{\text{bottom}} \\ \mathbf{H}_{N/2}\boldsymbol{x}_{\text{top}} - \mathbf{H}_{N/2}\boldsymbol{x}_{\text{bottom}} \end{bmatrix}$$

into two matrix products, each of size $N/2 \times N/2$. Here $\boldsymbol{x}_{\text{top}}$ are the $N/2$ first entries of $\boldsymbol{x}$ while $\boldsymbol{x}_{\text{bottom}}$ are the $N/2$ last entries. Next let q_i denote the number of operations required for the multiplication $\mathbf{H}_{2^i}\boldsymbol{x}$ of size $2^i \times 2^i$. By applying the matrix decomposition recursively, we can describe the number of operation through the difference equation $q_i = 2q_{i-1} + 2^i$. For $q_0 = 1$, this difference equation have the solution $q_i = (i+1)2^i$, which implies that the overall complexity of the matrix product is $\mathcal{O}(N \log_2 N)$ [13, p. 505].

Iterative algorithm To apply the recursive strategy presented above directly, would in any case outperform the usual matrix product for sufficiently large N. Recursive algorithms are however known to be suboptimal compared to any equivalent iterative algorithm. Any recursive algorithm will always require a large amount of function calls, which all together will increase the computational time of the algorithm. [28, pp. 87–89].

Iterative algorithms are therefore the preferable choice. To compute the Hadamard matrix product in an iterative manner we will decompose the matrix into a sequence of matrix products $\mathbf{H}_{2^R} = \mathbf{A}_{R-1} \cdots \mathbf{A}_1$. In this product each of the matrices $\mathbf{A}_i$ are the block diagonal matrix

$$\mathbf{A}_i = \text{diag}(\underbrace{\mathbf{B}_{2^i}, \ldots, \mathbf{B}_{2^i}}_{r}), \qquad r = 2^{R-i}$$

with

$$\mathbf{B}_{2^i} = \begin{bmatrix} \mathbf{I}_{2^{i-1}} & \mathbf{I}_{2^{i-1}} \\ \mathbf{I}_{2^{i-1}} & -\mathbf{I}_{2^{i-1}} \end{bmatrix}$$

where $\text{diag}(\cdots)$ denotes the matrices along the diagonal and $\mathbf{I}_K$ denotes the identity matrix of size K. Using this decomposition one can easily write the desired for-loops, implementing each of these matrix products in turn [13, p. 508].

4.2 The sequency ordered Hadamard matrix

The sequency and Paley enumerated Hadamard matrix is defined through the use of Walsh functions, as opposed to the simple recursive definition of the ordinary matrix. To define these functions, we will need some notation for the dyadic expansion of the real positive numbers. We therefore start by reviewing some elementary theory of the dyadic numeral system. Recall that any number $x \in [0,1)$ have a dyadic expansion

$$x = x_1 2^{-1} + x_2 2^{-2} + \cdots + x_j 2^{-j} + \ldots$$

with $x_i \in \{0,1\}$.

For a rational number x this expansion is not unique, as we may either choose the finite expansion, or the infinite expansion with $x_i = 1$ for all $i \geqslant k$, for some $k \in \mathbb{N}$. To avoid this ambiguity we will always choose the finite expansion. This means that we in practice have removed countably many singletons from the interval $[0,1)$. As the removed set have Lebesgue measure zero, we will see later on that it will not cause any problems for us. The gain on the other hand, is an isomorphic mapping between any $x \in [0,1)$ and a dyadic sequence $\{x_1, x_2, \ldots\}$.

In a similar manner, one can find a unique dyadic expansion of any number $n \in \mathbb{Z}_+$, as

$$n = n_1 2^0 + n_2 2^1 + \cdots + n_j 2^{j-1} + \ldots$$

This expansion is necessarily 0 from some point on, as the number n is finite. Using this setup we can define the *sequency ordered Walsh function.*

Definition 4.2. Let $n \in \mathbb{Z}_+$ and $x \in [0,1)$. The Walsh function for sequency ordered Hadamard transform is

$$\mathrm{WAL}(n,x) := (-1)^{\sum_{i=1}^{\infty}(n_i+n_{i+1})x_i}.$$

The sequency ordered Hadamard matrix is then initialized at the (i,j) entry using the value $\mathrm{WAL}(i-1,(j-1)/2^R)$, for $i,j = 1,\ldots,N$. Using this ordering, each of the matrix rows will contain one more sign change than the previous row. Thus the "frequency" of the WAL function will be strictly increasing for increasing n [4, pp. 7,17-18].

Gray code

In computer science there have been developed several elementary operators on binary sequences. Typical operators are NOT, AND, OR, XOR and the bit shift operators $\ll, \gg$. In our work with the Walsh functions the XOR operator will be used extensively on binary sequences. We therefore introduce it as the operator $\oplus$.

Definition 4.3. Let $x = \{x_i\}_{i=1}^{\infty}$ and $y = \{y_i\}_{i=1}^{\infty}$, be sequences consisting of only binary numbers. That is $x_i, y_i \in \{0,1\}$ for all $i \in \mathbb{N}$. The operator $\oplus$ of these sequences is then

$$x \oplus y := \{|x_i - y_i|\}_{i=1}^{\infty}.$$

For two binary numbers $x_i, y_i \in \{0,1\}$ we let $x_i \oplus y_i = |x_i - y_i|$.

Using this definition we can now introduce the *gray code* of a binary sequence.

Definition 4.4 (Gray code)**.** Let $n \in \mathbb{Z}_+$ have a binary representation $\{n_1, n_2, \ldots\}$, $n_i \in \{0, 1\}$ where $n = n_1 2^0 + n_2 2^1 + \cdots$. The gray code of n is

$$g(n) := n \oplus 2n = \{n_i \oplus n_{i+1}\}_{i=1}^{\infty}$$

The gray code of a finite binary sequence, is an alternative representation where only one bit is changed every time the value of the corresponding decimal number increases by one. This structure can be seen in table 4.1, where the first 16 decimals are written in both systems.

In definition 4.2 of the sequency ordered Walsh function, we used the usual '+' operator in the expression $n_i + n_{i+1}$. This operator could also be replaced by the $\oplus$ operator, since $(-1)^0 = (-1)^2$. Using this operator we recognize this as the gray code representation of n. This fact will be used extensively later, in the computations of the $\mathcal{O}(N \log_2 N)$ matrix product with the sequency ordered Hadamard transform [4, p. 23].

Decimal	Binary	Gray		Decimal	Binary	Gray
0	0	0		8	1000	1100
1	1	1		9	1001	1101
2	10	11		10	1010	1111
3	11	10		11	1011	1110
4	100	110		12	1100	1010
5	101	111		13	1101	1011
6	110	101		14	1110	1001
7	111	100		15	1111	1000

Table 4.1: The ordinary representation of the binary number, and their gray code representation. Notice that only one bit changes in the gray code representation each time the decimal number increases by one.

4.3 The Paley ordered Hadamard matrix

The sequency ordered Hadamard matrix can be defined by the sequency ordered Walsh function WAL. In a similar manner, the Paley enumerated Hadamard matrix can be defined through another Walsh function, which we will denote as PAL. Both of these functions can be defined in a variety of ways, for instance through the use of difference equations [26, p. 19], as product of Rademacher functions [19] or by using binary sequences as we did above [23, Eq. 1.2.5]. We have chosen the latter approach as much of the theory of Walsh functions have been derived for this definition.

Definition 4.5. Let $n \in \mathbb{Z}_+$ and $x \in [0, 1)$. The Walsh function for the Paley ordered Hadamard transform is

$$\mathrm{PAL}(n, x) := (-1)^{\sum_{i=1}^{\infty} n_i x_i}.$$

This function will also be denoted as $w_n(x) := \mathrm{PAL}(n, x)$ to shorten the notation.

From this definition it is easy to see the equality

$$\mathrm{WAL}(n, x) = \mathrm{PAL}(g(n), x). \tag{4.3}$$

Hence the WAL function is the gray code permuted PAL function.

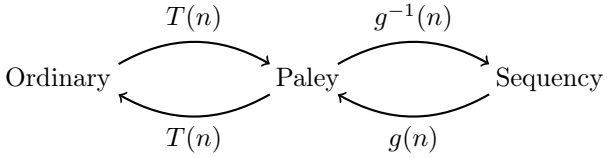
To initialize the Paley ordered Hadamard matrix, we can evaluate the function $\mathrm{PAL}((i-1), (j-1)/2^R)$ for $i, j = 1, \ldots, N$, as we did for the sequency ordered matrix. This approach will, however, require that we compute and store all the N^2 elements. This creates a densely stored matrix, whose matrix product will have a cost of $\mathcal{O}(N^2)$ operations.

Preserving the $\mathcal{O}(N \log_2 N)$ requirement

For our applications to work satisfactory, the critical requirement of a transform that can be computed using $\mathcal{O}(N \log_2 N)$ operations, can not be relaxed. The two previous orderings do not possess this property, using their usual definition. To obtain these transforms within the desired amount of floating point operations, we have to rely on the fact that all of these matrices consist of the same set of rows, ordered in different ways. As all of the coefficients from the ordinary ordering transform corresponds to the inner product with the matrix's rows, a reordering of these coefficients is equivalent to a reordering of the rows.

Hence, by applying the ordinary ordering transform using $\mathcal{O}(N \log_2 N)$ operations, the computational bottleneck of the transform have been bypassed. Afterwards, one permutes these coefficients into the desired order, at a cost of $\mathcal{O}(N)$ operations. The ordering between the sequency and Paley enumeration is as we have seen in equation (4.3), given by the gray code of the row number. Similarly, one can show that the ordinary ordering of the coefficients corresponds to a bit reversed Paley enumeration [4, p. 18].

Let the permutation which reverses the finite bit sequences be denoted as $T(n) := \{n_R, n_{R-1}, \ldots, n_1\}$ for $n = \{n_1, \ldots, n_R\}$. We can then write this permutation system as we have done in the following figure.



Unfortunately, the gray code permutations of a sequence with 2^R elements contain many cycles. This can be seen in table 4.1, which contains the five cycles

$$\begin{array}{lllllllll}
1 & \to & 1 & & & & & & \\
2 & \to & 3 & \to & 2 & & & & \\
4 & \to & 6 & \to & 5 & \to & 7 & \to & 4 \\
8 & \to & 15 & \to & 10 & \to & 12 & \to & 8 \\
9 & \to & 13 & \to & 11 & \to & 14 & \to & 9
\end{array}$$

Thus to compute the gray code permutation of an array in-place one must detect all of the cycles and permute with respect to these. Such a setup can be seen in e.g., [3, pp. 128–133, 474–481]. In the *fastwht* library, a simpler approach was chosen, by creating an intermediate array to store all the permuted elements.

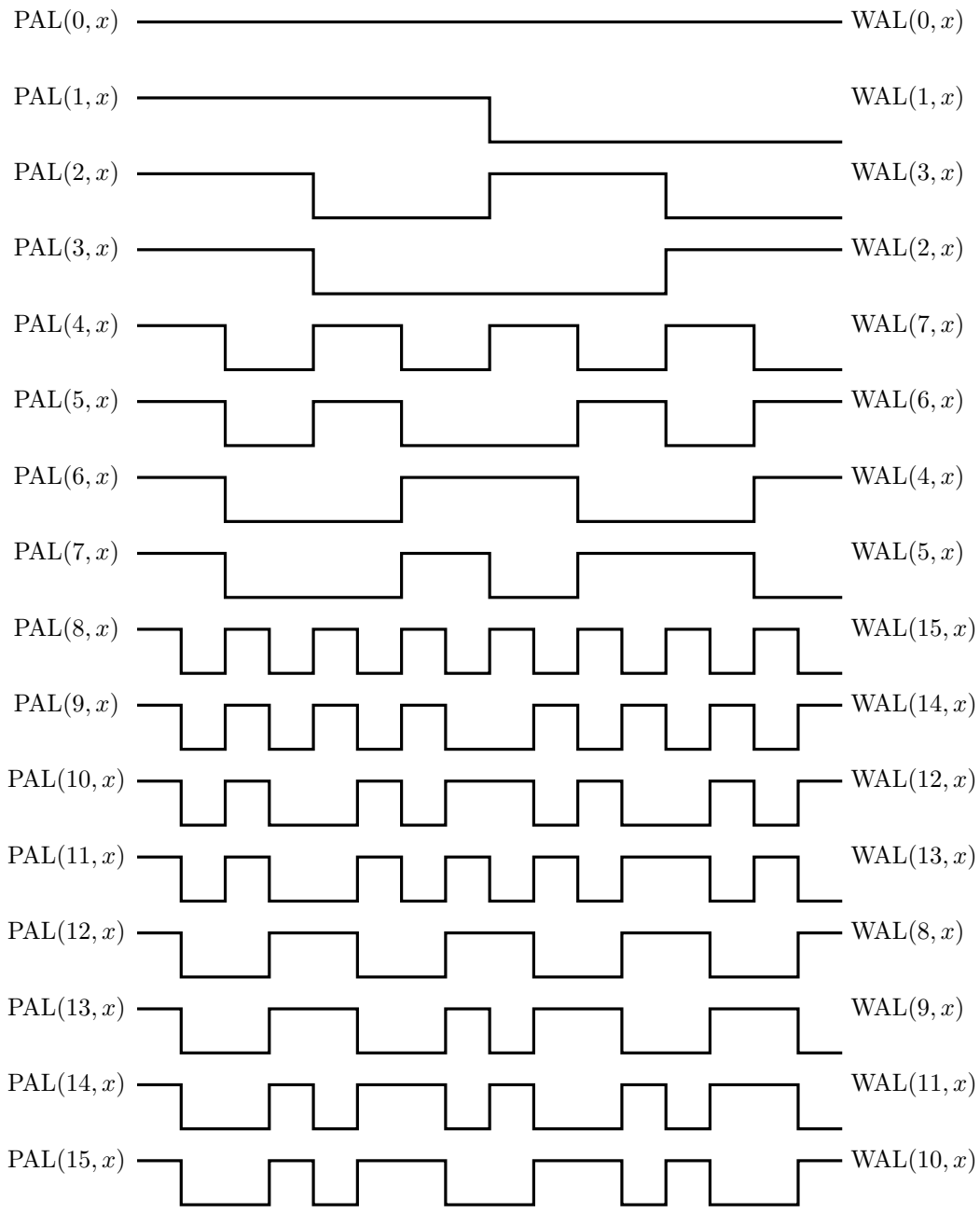


Figure 4.2: The 16 first PAL and WAL functions.

Choice of ordering

In compressive sensing, a critical requirement is the asymptotic incoherence property between the sampling basis and sparsifying basis. In chapter 5 we will see numerically that this property is not present for the ordinary ordered Hadamard matrix. For our applications we will therefore not consider this ordering.

In chapter 6 we will prove that the incoherence property holds for the sequency and Paley enumerated Hadamard matrix. To do this, the following lemma is needed.

Lemma 4.6
Let the rows of a $2^R \times 2^R$ matrix be labeled from 0 (top row) to $2^R - 1$ (bottom row). The set of rows between $2^p \leqslant n < 2^{p+1}$ is the same in both the Paley and sequency ordered Hadamard matrix for $p = 0, 1, \dots R-1$.

Proof. For $2^p \leqslant n < 2^{p+1}$ we know that the binary sequence representation of n is $\{n_1 \dots n_p, 1, 0, \dots\}$. If we let $m = g(n)$ be the gray code of n, we also know that m will be represented in the same way, $\{m_1, \dots, m_p, 1, 0, \dots\}$ since $m_{p+1} = n_{p+1} \oplus n_{p+2} = 1 \oplus 0 = 1$. This means that if $n \in [2^p, 2^{p+1})$ then $m \in [2^p, 2^{p+1})$. As g is injective, the result follows. □

4.4 Walsh transform

In section 3.1 we saw that a function $f \in \mathcal{L}^1([0,1])$ could be written in the Fourier domain by applying the Fourier transform, and writing the function as a Fourier series. In this section we will state a similar result for the Walsh functions.

First we will, however, investigate the Walsh-functions more closely, by deriving some well known properties. All of these properties will turn out to be useful when we later investigate the coherence between the Hadamard matrices and Daubechies wavelets. The lemmas we present below can be found in [23, ch. 1], however, most of them is not be explicitly stated as we have done below. As most of the theory of the Walsh transform and Walsh series have been derived for the PAL function, we will continue this practice. To shorten notation we use the abbreviation $w_n(x) = \mathrm{PAL}(n, x)$.

Lemma 4.7
Let $x, y \in [0,1)$ have disjoint dyadic support i.e., $x_i = 1 \implies y_i = 0$ and $y_j = 1 \implies x_j = 0$. Then

$$x + y = x \oplus y$$

Proof. The results follows directly from the definition of $\oplus$. □

Lemma 4.8
Let n and $p \geqslant 0$ be integers such that $2^p \leqslant n < 2^{p+1}$ and let

$$\Delta_k^{p+1} = \left[k/2^{p+1}, (k+1)/2^{p+1}\right)$$

for $k \in \{0, \dots, 2^{p+1} - 1\}$. Then w_n is constant on each of the intervals Δ_k^{p+1}. Each of the intervals Δ_k^p can be decomposed into the intervals Δ_{2k}^{p+1} and Δ_{2k+1}^{p+1}, where w_n is equal to 1 on exactly one of them, and equal to -1 on the other.

Proof. Since $2^p \leqslant n < 2^{p+1}$ we know that an dyadic expansion of n will have the following form $n = \{n_1, \ldots, n_p, 1, 0, \ldots\}$ i.e., $n_{p+1} = 1$, $n_k = 0$ for $k \geqslant p+2$. Further we know that for all $x \in \Delta_k^{p+1}$ the first $p+1$ binaries in the dyadic expansion will be fixed and thus w_n will be constant for all $x \in \Delta_k^{p+1}$. Further we know that

$$\begin{aligned} x \in \Delta_{2k}^{p+1} &\implies \{x_1, \ldots, x_p, 0, \ldots\} \\ x \in \Delta_{2k+1}^{p+1} &\implies \{x_1, \ldots, x_p, 1, \ldots\} \end{aligned}$$

Since $n_{p+1} = 1$, we know that if $w_n(x) = 1$ for $x \in \Delta_{2k}^{p+1}$ then $w_n(x) = -1$ for Δ_{2k+1}^{p+1} and vice versa. □

Proposition 4.9
For $R, n, m \in \mathbb{Z}_+$ and $x, y \in [0, 1)$, the Walsh function satisfies the following relations

(a)

$$w_n(x \oplus y) = w_n(x) w_n(y)$$

(b)

$$w_n(x) w_m(x) = w_{n \oplus m}(x)$$

(c)

$$\int_0^1 w_n(x) w_m(x) \, \mathrm{d}x = \begin{cases} 1 & \text{if } m = n \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

(d)

$$w_n(2^{-R} x) = w_{\lfloor n/2^R \rfloor}(x)$$

Proof.

(a)

$$\begin{aligned} w_n(x \oplus y) &= (-1)^{\sum_{j=0}^{\infty} n_j |x_j - y_j|} = (-1)^{\sum_{j=1}^{\infty} n_j (x_j + y_j)} \\ &= (-1)^{\sum_{j=1}^{\infty} n_j x_j} (-1)^{\sum_{j=1}^{\infty} n_j y_j} = w_n(x) w_n(y) \end{aligned}$$

(b) The result follows by a similar computation as in (a).

(c) The integral in equation (4.4) is a Lebesgue integral. Since we have removed countably many singletons in $[0, 1)$, the removed set have Lebesgue measure zero. The integral over the set where the singletons are removed, and the integral where they are not will therefore coincide.

Assume $m \neq n$ then $w_n(x) w_m(x) = w_{n \oplus m}(x) = w_\ell(x)$ for $\ell = m \oplus n > 0$. Let $p \in \mathbb{Z}_+$ be such that $2^p \leqslant \ell < 2^{p+1}$. From lemma 4.8 we know that w_ℓ will be equal to 1 on exactly one of the intervals Δ_{2k}^{p+1} and Δ_{2k+1}^{p+1} while it will be -1 on the other. This implies that

$$\int_{\Delta_k^p} w_\ell(x) \, \mathrm{d}x = 0$$

Thus we can derive

$$\int_0^1 w_\ell(x)\,\mathrm{d}x = \sum_{k=0}^{2^p-1} \int_{\Delta_k^p} w_\ell(x)\,\mathrm{d}x = 0$$

For $m = n$ we obtain $m \oplus n = 0$, which implies that $w_0(x) = 1$ and $\int_0^1 1\,\mathrm{d}x = 1$.

(d)

$$w_n(2^{-R}x) = (-1)^{\sum_{j=1}^{R} 0\cdot n_j + \sum_{j=R+1}^{\infty} n_j x_{j-R}} = (-1)^{\sum_{j=1}^{R} n_{j+R} x_j} = w_{\lfloor n/2^R \rfloor}(x)$$

□

Definition 4.10 (Walsh transform)**.** Let f be a Lebesgue integrable function on $[0, 1)$. The Walsh transform of f is

$$\check{f}(n) := \int_0^1 f(x) w_n(x)\,\mathrm{d}x.$$

Further, one can show that the set of Walsh functions $\{w_n\}_{n\in\mathbb{Z}_+}$ is closed and complete in the spaces $\mathcal{L}^p([0,1])$, $p \geqslant 1$ [23, Sec. 2.6]. Thus, if $f \in \mathcal{L}^1([0,1])$ the series

$$\sum_{n=0}^{\infty} \check{f}(n) w_n$$

converges to f almost everywhere.

CHAPTER 5

Asymptotic compressive sensing

Traditionally the theory of compressive sensing have been based on the three principles of *sparsity*, *incoherence* and *uniform random subsampling*. These principles rely on the assumption that we have no priori knowledge of the position of the signal's non-zero entries. Practical experiments, using wavelets as the sparsifying transform do, however, impose an asymptotic structure on the non-zero coefficients of the signal. For such setups these old principles have proven to yield insufficient results [31]. Due to these insufficiencies we will therefore review the principles of *asymptotic sparsity*, *asymptotic incoherence* and *multilevel random subsampling*.

These principles and the related theorems involve a lot of notation and technicalities. To redefine all of the needed structure for a complete presentation of this theory is beyond the scope of this text. Instead interested readers are referred to the paper [2] by Adcock, Hansen, Poon & Roman. We will only review the most important theory, to motivate the need for a new coherence result between Hadamard matrices and Daubechies wavelets.

The principles may be reviewed either in a finite or infinite setting. We will state the following definitions for both settings, while we will comment on the finite dimensional theory. The infinite-dimensional theory will be considered on its own in section 5.4.

5.1 Asymptotic principles

We recall from chapter 1 that the problem at hand is to construct a sampling scheme $\Omega \subseteq \{1, \dots, N\}$ such that the recovered solution $\boldsymbol{c}^\sharp$ of the optimization problem

$$\text{minimize } ||\boldsymbol{z}||_1 \quad \text{subject to} \quad ||\mathbf{P}_\Omega \mathbf{U} \boldsymbol{z} - \boldsymbol{b}||_2 \leqslant \eta \qquad (P_{1,\eta,\Omega})$$

is close to the wavelet coefficients $\boldsymbol{c} = \boldsymbol{\Psi}\boldsymbol{x}$. In this setup $\mathbf{P}_\Omega$ was the projection onto the set Ω and $\mathbf{U}$ was an isometry consisting either of $\mathbf{V}_{\text{dft}}\boldsymbol{\Psi}^{-1}$ or $\mathbf{V}_{\text{had}}\boldsymbol{\Psi}^{-1}$, where $\mathbf{V}_{\text{dft}}$ was the discrete Fourier transform, $\mathbf{V}_{\text{had}}$ was the Hadamard matrix and $\boldsymbol{\Psi}$ was the discrete wavelet transform. The solution $\boldsymbol{c}^\sharp$ of $(P_{1,\eta,\Omega})$ would correspond to the wavelet coefficients of $\boldsymbol{x}$ i.e., $\boldsymbol{c}^\sharp \approx \boldsymbol{\Psi}\boldsymbol{x}$.

Due to the properties of the sparsifying wavelet transform we argued that most of the non-zero coefficients would be at the beginning of the wavelet coefficient vector $\boldsymbol{c}$. Thus to recover such a vector this structure should be reflected

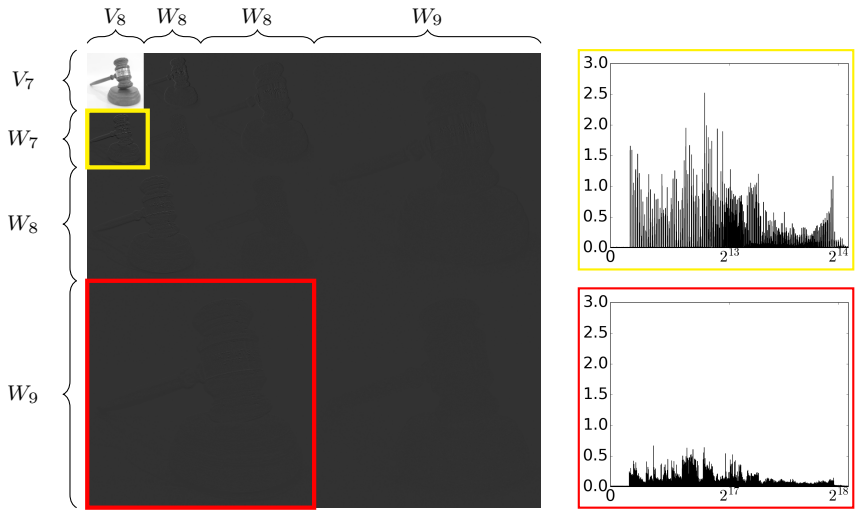


Figure 5.1: Two dimensional discrete wavelet transform using the Haar wavelet. The original image have values in the range $[0, 1]$. Red box: 2.1% of the coefficients have an absolute value above 0.06. Yellow box: 21.3 % of the coefficients have an absolute value above 0.06.

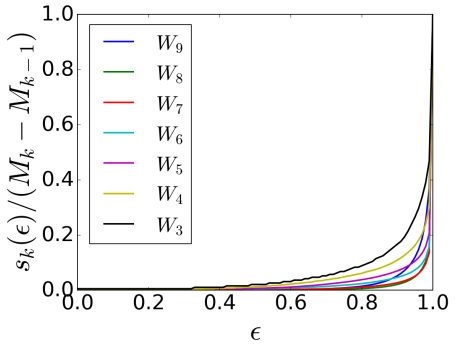


Figure 5.2: The local sparsity in ℓ_2-norm of the different wavelet detail spaces of the image 5.6a. Some of these detail spaces can also be seen visually in figure 5.1.

in the sampling scheme. We therefore start by formalizing this *asymptotic sparsity* structure.

Definition 5.1 (Asymptotic sparsity [2])**.** Let $\boldsymbol{x}$ be an element of either $\mathbb{C}^N$ or $\ell^2(\mathbb{N})$. Let $\mathbf{M} = [M_1, \dots, M_r] \in \mathbb{N}^r$ be the *sparsity levels* of $\boldsymbol{x}$ with $1 \leqslant M_1 < \cdots < M_r$. Further let $\boldsymbol{s} = [s_1, \dots, s_r] \in \mathbb{N}^r$ with $s_k \leqslant M_k - M_{k-1}$, $k = 1, \dots r$ and $M_0 = 0$, be the sparsity within each level. We say that $\boldsymbol{x}$ is $(\boldsymbol{s}, \mathbf{M})$-sparse if, for each $k = 1, \dots, r$, the set $\Gamma_k := \operatorname{supp}(\boldsymbol{x}) \cap \{M_{k-1} + 1, \dots, M_k\}$ satisfies $|\Gamma_k| \leqslant s_k$. We denote the set of $(\boldsymbol{s}, \mathbf{M})$-sparse vectors by $\Sigma_{\boldsymbol{s},\mathbf{M}}$. We say that a vector $\boldsymbol{y} \in \Sigma_{\boldsymbol{s},\mathbf{M}}$ is *asymptotically sparse in levels* if $s_k/(M_k - M_{k-1}) \to 0$ as $k \to \infty$.

For our purposes we shall let the different M_k's correspond to the boundaries between the wavelet resolution spaces. Applying a DWT to a vector $\boldsymbol{x} \in V_R$, corresponds to decomposing it into the space $V_0 \oplus_{k=0}^{R-1} W_k$, where the size of W_k is 2^k. A natural choice is therefore $\mathbf{M} = [2^0, 2^1, \dots 2^R]$.

The principle of asymptotic sparsity can be seen in figure 5.1, where a three level, two-dimensional DWT have been applied to the image. This has resulted in one low resolution approximation of the image, and three detail spaces at different resolutions. Each of these detail spaces can again be divided into three different detail spaces, based on the type of wavelet used in the decomposition.

By studying the image closely, one would find that the fraction of large coefficients in the first detail space is much lower than in the other spaces. As one decompose the image into the next detail space this fraction increases. This is illustrated in the two colored boxes in the image. In section 5.3 we will also see a practical setup exploiting this fact.

In practice, a DWT will only produce coefficients which are compressible rather than sparse. To measure the degree of sparsity in practice, we therefore need a measure taking this consideration into account. One such measure is the *local sparsity*.

Definition 5.2 (Local sparsity [31])**.** Let be $\mathbf{M} = [M_1, \dots, M_r] \in \mathbb{N}^r$ be the sparsity levels of a vector $\boldsymbol{x} \in \mathbb{C}^N$. Further, let $\mathcal{M}_k = \{M_{k-1} + 1, \dots, M_k\}$ and let $\mathcal{M}_{k,L} \subseteq \mathcal{M}_k$ be such that $|x_l| \geqslant |x_j|$ for all $l \in \mathcal{M}_{k,L}$ and all $j \in \mathcal{M}_k \backslash \mathcal{M}_{k,L}$. For $\epsilon \in (0, 1]$ the *local sparsity* of $\boldsymbol{x}$ is

$$s_k(\epsilon) := \min \left\{ L : ||\boldsymbol{x}_{\mathcal{M}_{k,L}}|| \geqslant \epsilon ||\boldsymbol{x}_{\mathcal{M}_k}|| \right\}$$

For reasonable choices of ϵ this measure will necessarily yield high values for uniform signals, while lower values for sparse signals. In figure 5.2 we have used this measure to find the sparsities within each level for $\epsilon \in [0, 1]$. In this image the fraction of significant coefficients within each level decreases with increasing resolution spaces for almost all spaces. The only exception is the last space W_9, which have a more uniform distribution of low value coefficients than the other detail spaces. In practice we will, however, characterize this signal as asymptotically sparse in wavelets.

As the sparsity within each level changes, it is natural to create a sampling pattern which takes these considerations into account. A sampling strategy with provable good results are multilevel random subsampling.

Definition 5.3 (Multilevel random subsampling [2])**.** Let $\mathbf{N} = [N_1, \dots, N_r] \in \mathbb{N}^r$ with $1 \leqslant N_1 < \cdots < N_r$, and $\boldsymbol{m} = [m_1, \dots, m_r] \in \mathbb{N}^r$ with $m_k \leqslant N_k - N_{k-1}$, $k = 1, \dots, r$ and $N_0 = 0$. Suppose the local sampling sets $\Omega_k \subseteq \{N_{k-1} + 1, \dots, N_k\}$, with $|\Omega_k| = m_k$ for $k = 1, \dots, r$ are chosen uniformly at random. We then refer to the set $\Omega_{\mathbf{N},\boldsymbol{m}} = \Omega_1 \cup \cdots \cup \Omega_r$ as an $(\mathbf{N}, \boldsymbol{m})$-multilevel sampling scheme.

Using a Hadamard sampling matrix, it is natural to choose the sampling levels equal to the boundaries between the different Walsh frequency resolutions. That is $\mathbf{N} = \mathbf{M} = [2^0, 2^1, \dots, 2^R]$. In order to obtain a complete sampling scheme, we must also decide on the sampling distribution $\boldsymbol{m} = [m_1, \dots, m_r] \in \mathbb{N}^r$. To find a good such distribution, we first need to know the coherence properties of our system.

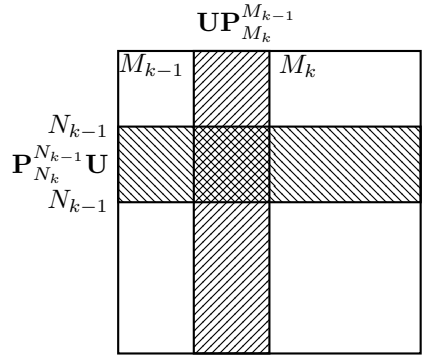


Figure 5.3: The effect of the projection $\mathbf{P}_{N_k}^{N_{k-1}}\mathbf{U}$ and $\mathbf{U}\mathbf{P}_{M_k}^{M_{k-1}}$ used in the definition of coherence.

Before we review the definition of coherence, we start by recalling the following notation from definition 1.3. $\mathbf{P}_b^a$ denotes the projection onto $\text{span}\{\boldsymbol{e}_j : j = a+1, \dots, b\}$ where $\boldsymbol{e}_j$ is the canonical basis. Similarly, we let $(\mathbf{P}_K^0)^\perp = \mathbf{P}_K^\perp$ denote $\mathbf{P}_K^0$'s orthogonal complement. Further, we will also let $\mathcal{B}(X)$ denote the set of bounded linear operators on the normed linear space X.

Definition 5.4 (Asymptotic incoherence [2])**.** Let $\{\mathbf{U}_N\}$ be a sequence of isometries with $\mathbf{U}_N \in \mathbb{C}^{N\times N}$ or let $\mathbf{U} \in \mathcal{B}(\ell^2(\mathbb{N}))$ be an isometry. Then

(a) $\{\mathbf{U}_N\}$ is asymptotically incoherent of $\mu(\mathbf{P}_K^\perp \mathbf{U}_N)$, $\mu(\mathbf{U}_N \mathbf{P}_K^\perp) \to 0$ as $K \to \infty$ with $N/K = c$ for all $c \geqslant 1$.

(b) $\mathbf{U}$ is asymptotically incoherent if $\mu(\mathbf{P}_K^\perp \mathbf{U})$, $\mu(\mathbf{U}\mathbf{P}_K^\perp) \to 0$, when $K \to \infty$.

Here $\mathcal{B}(X)$ denotes the set of bounded linear operators on the normed linear space X.

Definition 5.5 (Local coherence [2])**.** Let $\mathbf{U}$ be an isometry of either $\mathbb{C}^N$ or $\ell^2(\mathbb{N})$. Further, let $\mathbf{M}, \mathbf{N} \in \mathbb{N}^r$ with $1 \leqslant N_1 < \cdots < N_r$, $1 \leqslant M_1 < \cdots < M_r$ and let $M_0 = N_0 = 0$. The $(k,l)^{\text{th}}$ *local coherence* of $\mathbf{U}$ with respect to $\mathbf{M}$ and $\mathbf{N}$ is given by

$$\mu_{\mathbf{N},\mathbf{M}}(k,l) \coloneqq \sqrt{\mu\left(\mathbf{P}_{N_k}^{N_{k-1}} \mathbf{U} \mathbf{P}_{M_l}^{M_{l-1}}\right) \mu\left(\mathbf{P}_{N_k}^{N_{k-1}} \mathbf{U}\right)} \qquad k,l = 1,\dots,r.$$

From chapter 2 we learned that the best s-term approximation to a solution vector $\boldsymbol{x}$ was given by $\sigma_s(\boldsymbol{x})_p \coloneqq \inf\{||\boldsymbol{x} - \boldsymbol{z}||_p, ||\boldsymbol{z}||_0 \leqslant s\}$. This constituted a lower bound on the error in the search for a s-sparse solution. As we are searching for a $(\boldsymbol{s}, \mathbf{M})$-sparse solution of $(P_{1,\eta,\Omega})$, we need to introduce a similar lower bound of *best* $(\boldsymbol{s}, \mathbf{M})$*-term approximation*

$$\sigma_{\boldsymbol{s},\mathbf{M}}(\boldsymbol{x})_p \coloneqq \inf\{||\boldsymbol{x} - \boldsymbol{z}||_p : \boldsymbol{z} \in \Sigma_{\boldsymbol{s},\mathbf{M}}\}$$

Using these definitions we are able to state the main theorem in this section. We state the theorem for the finite dimensional setting, an infinite version can be found in [2, Thm. 5.3]. To shorten notation, we use the abbreviation $A \gtrsim B$ if $A \geqslant CB$ for some constant C independent of all relevant parameters.

Theorem 5.6 ([2])
Let $\mathbf{U} \in \mathbb{C}^{N\times N}$ *be an isometry and* $\boldsymbol{x} \in \mathbb{C}^N$. *Suppose that* $\Omega = \Omega_{\mathbf{N},\mathbf{M}}$ *is a multilevel sampling scheme, where* $\mathbf{N} = [N_1,\dots,N_r] \in \mathbb{N}^r, N_r = N$ *and* $\boldsymbol{m} = [m_1,\dots,m_r] \in \mathbb{N}^r$. *Further let* $\mathbf{M} = [M_1,\dots,M_r] \in \mathbb{N}^r$ *be the sparsity levels of* $\boldsymbol{x}$ *with* $M_r = N$, *and let* $\boldsymbol{s} = [s_1,\dots,s_r] \in \mathbb{N}^r$ *be the sparsities within each of these levels. Let* $(\boldsymbol{s},\mathbf{M})$ *be any pair such that the following holds: for* $\epsilon \in (0, e^{-1}]$ *and* $1 \leqslant k \leqslant r$,

$$1 \gtrsim \frac{N_k - N_{k-1}}{m_k} \log(\epsilon^{-1}) \left(\sum_{l=1}^{r} \mu_{\mathbf{N},\mathbf{M}}(k,l) \cdot s_l \right) \log N, \tag{5.1}$$

where $m_k \gtrsim \hat{m}_k \log(\epsilon^{-1}) \log(N)$, *and* $\hat{m}_k$ *is such that*

$$1 \gtrsim \sum_{k=1}^{r} \left(\frac{N_k - N_{k-1}}{\hat{m}_k} - 1 \right) \cdot \mu_{\mathbf{N},\mathbf{M}}(k,l) \tilde{s}_k \tag{5.2}$$

for all $l = 1,\dots,r$ *and all* $\tilde{s}_1,\dots,\tilde{s}_r \in (0,\infty)$ *satisfying*

$$\tilde{s}_1 + \cdots + \tilde{s}_r \leqslant s_1 + \cdots + s_r, \qquad \tilde{s}_k \leqslant \max_{\boldsymbol{z}\in\Theta} ||\mathbf{P}_{N_k}^{N_{k-1}} \mathbf{U}\boldsymbol{z}||^2,$$

where $\Theta = \{\boldsymbol{z} \in \mathbb{C}^N : ||\boldsymbol{z}||_\infty \leqslant 1, ||\mathbf{P}_{M_l}^{M_{l-1}} \boldsymbol{z}||_0 \leqslant s_l, l = 1,\dots,r\}$. *Suppose that* $\boldsymbol{c}^\sharp \in \mathbb{C}^N$ *is a minimizer of* $(P_{1,\eta,\Omega})$ *with* $\eta = \tilde{\eta}\sqrt{K^{-1}}$ *and* $K = \max_{1\leqslant k\leqslant r}\{(N_k - N_{k-1})/m_k\}$. *Then with probability exceeding* $1 - s\epsilon$, *where* $s = s_1 + \cdots + s_r$, *we have that*

$$||\boldsymbol{x}^\sharp - \boldsymbol{x}|| \lesssim \left(\tilde{\eta}(1 + L\sqrt{s}) + \sigma_{\boldsymbol{s},\mathbf{M}}(\boldsymbol{x})_1 \right),$$

for $L = 1 + \frac{\sqrt{\log_2(6\epsilon^{-1})}}{\log_2(4KM\sqrt{s})}$. *If* $m_k = N_k - N_{k-1}$, $1 \leqslant k \leqslant r$, *then this holds with probability 1.*

The key property of this result is the bounds of equation (5.1) and (5.2). Form these bounds we see that the success of any multilevel sampling scheme, depends on the local coherences of the isometry $\mathbf{U}$, the number of samples within each levels m_k, and the sparsity s_k and $\tilde{s}_k$. In particular we see that it would be advantageous if the isometry $\mathbf{U}$ was asymptotically incoherent, as it would imply that we could reduce the number of samples m_k for large k. As the size of each detail space W_k is 2^k, this is an important property, even for mildly large k.

To clarify the idea, we will review the coherence structures for two concrete matrices, Hadamard matrix multiplied with the Haar wavelet matrix and Fourier matrix multiplied with a Daubechies wavelet matrix.

5.2 Two asymptotic coherence estimates

Coherence between Hadamard and Haar wavelets

The simplest of all isometries possessing the asymptotic incoherence property is a sequency or Paley enumerated Hadamard matrix multiplied with the inverse Haar wavelet matrix. The Haar wavelet is a Daubechies-1 wavelet, having

one vanishing moment i.e., $\int_{\mathbb{R}} \psi(t)\,\mathrm{d}t = 0$. It is defined through the piecewise constant functions

$$\phi(x) = \begin{cases} 1 & \text{if } x \in [0,1) \\ 0 & \text{otherwise} \end{cases}, \qquad \psi(x) = \begin{cases} 1 & \text{if } x \in [0,1/2) \\ -1 & \text{if } x \in [1/2,1) \\ 0 & \text{otherwise} \end{cases}.$$

As each wavelet $\psi_{j,k} = 2^{j/2}\psi(2^j \cdot - k)$ has support on the interval $[2^{-j}k, 2^{-j}(k+1)]$, this implies that it will be constant on two dyadic intervals of size $2^{-(j+1)}$. Thus, due to lemma 4.8 it is not hard to see that $\psi_{j,k}$ will be orthogonal to all Walsh-functions w_n outside the range $2^j \leqslant n < 2^{j+1}$. That is

$$\begin{aligned} \left| \int_0^1 \psi_{j,k}(x) w_n(x)\,\mathrm{d}x \right| &= \left| \int_0^1 2^{j/2}\psi(2^j x - k) w_n(x)\,\mathrm{d}x \right| \\ &= 2^{j/2} \left| \int_{2^{-j}k}^{2^{-j}(k+1)} \psi(2^j x - k) w_n(x)\,\mathrm{d}x \right| \\ &= \begin{cases} 2^{-j/2} & \text{if } n \in [2^j, 2^{j+1}) \\ 0 & \text{otherwise} \end{cases}. \end{aligned}$$

Next we fix the sparsity and sampling levels to $\mathbf{M} = \mathbf{N} = [2^0, 2^1, \dots, 2^R]$ and let $\mathbf{\Psi}_{\text{haar}} \in \mathbb{R}^{N \times N}$ denote the Haar wavelet matrix and let $\mathbf{V}_{\text{had}} \in \mathbb{R}^{N \times N}$ denote the sequency or Paley enumerated Hadamard matrix for $N = 2^R$. It then follows that the local coherence of the matrix $\mathbf{U} = \mathbf{V}_{\text{had}}\mathbf{\Psi}_{\text{haar}}^{-1}$ is given by

$$\mu_{\mathbf{N},\mathbf{M}}(k,l) = \begin{cases} 2^{-(k-2)/2} & \text{if } k = l \\ 0 & \text{otherwise} \end{cases} \qquad k,l = 2,\dots R+1.$$

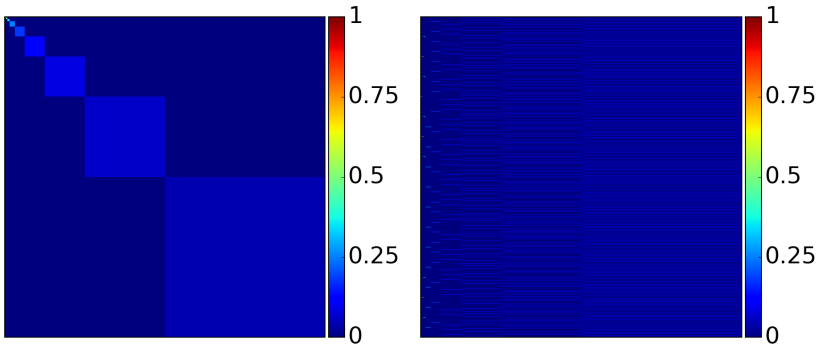
Here we have omitted the index 1, since it corresponds to the scaling function ϕ. It is an easy exercise to show that ϕ is orthonormal to w_n for all $n \neq 0$. The $k-2$ term is introduced to account for cumbersome indexing. To see this, note that $\mu_{\mathbf{N},\mathbf{M}}(R+1, R+1)$ corresponds to the lower right corner i.e., $\mathbf{P}_{2^R}^{2^{R-1}}\mathbf{U}\mathbf{P}_{2^R}^{2^{R-1}}$, which corresponds to the resolution space W_{R-1}. Hence the magnitude of the coefficients in this lower right corner is $\sqrt{2/N}$. This is close to being perfectly incoherent.

If we insert this estimate into the lower bound of (5.1) it becomes

$$\begin{aligned} m_R &\gtrsim (2^R - 2^{R-1}) \log(\epsilon^{-1}) \frac{\sqrt{2}}{2^{R/2}} s_R \log(2^R) \\ &= 2^{R/2-1/2} \log(\epsilon^{-1}) s_R \log(2^R). \end{aligned}$$

Where the $2^{R/2}$ term represents a tremendous reduction in the required number of samples. The greatest disadvantage of this wavelet is of course that it's discontinuity is not optimal for approximating smooth structures. To see this, assume we have solved $(P_{1,\eta,\Omega})$ and recovered a vector

$$\boldsymbol{c}^\sharp = \left\langle \boldsymbol{x}^\sharp, \phi_{0,0} \right\rangle \phi_{0,0} + \sum_{j=0}^{R-1} \sum_{k=0}^{2^j-1} \left\langle \boldsymbol{x}^\sharp, \psi_{k,j} \right\rangle \psi_{k,j}.$$



(a) (b)

Figure 5.4: The element-wise absolute values of (a) sequency ordered Hadamard and (b) ordinary ordered Hadamard matrix; multiplied by the Haar wavelet matrix. One can clearly see the block diagonal structure of the sequency ordering. The same block diagonal structure would bee seen, using Paley enumeration.

In each of these inner products $\left\langle \boldsymbol{x}^\sharp, \psi_{j,k} \right\rangle$ there will be an error ϵ, either due to round off error, the wavelet crime, or inaccurate recovery. If $\psi_{j,k}$ is smooth, the error $\epsilon\psi_{j,k}$ will also be smooth, while a discontinuous ψ will lead to more irregular error, even if the two errors have the same magnitude. In images a smooth error is often less visible than an irregular error. [29, p. 287]

Coherence between Fourier and orthonormal wavelets

The previous setup was particularly simple as $\operatorname{supp}(\phi) = \operatorname{supp}(\psi) = [0, 1)$ for the Haar scaling function and wavelet. This made it possible to construct an orthonormal wavelet basis on $\mathcal{L}^2([0, 1])$ without any boundary wavelets. For Daubechies wavelets with $\nu \geqslant 2$ vanishing moments, we know from theorem 3.6 that $\operatorname{supp}(\phi) = [0, 2\nu - 1]$ and $\operatorname{supp}(\psi) = [-\nu + 1, \nu]$. Thus, to construct a wavelet basis on $\mathcal{L}^2([0, 1])$ for compactly supported wavelets with $\nu \geqslant 2$ vanishing moments we have to handle the boundary problem.

In the coherence estimates provided by Adcock et al. in [2] one handles this problem in a simplified setup. In their work one assumes that ϕ and ψ are orthonormal with $\operatorname{supp}(\phi) = \operatorname{supp}(\psi) = [0, a]$ for some $a \geqslant 1$. Further one assumes that ψ has $\nu \geqslant 2$ vanishing moments. Using these wavelets one constructs an orthonormal basis on the interval $[0, a]$, by considering all wavelets intersecting this interval i.e.,

$$\Lambda_a = \{\phi_k, \psi_{j,k} : \operatorname{supp}(\phi_k)^\circ \cap [0, a] \neq \varnothing, \operatorname{supp}(\psi_{j,k})^\circ \cap [0, a] \neq \varnothing, j \in \mathbb{Z}_+, k \in \mathbb{Z}\}$$

where we have used the notation K° to denote the interior of a set $K \subseteq \mathbb{R}$. This construction essentially zero-expand all functions whose domain intersects $[0, a]$ and apply the general wavelet theory for $\mathbb{R}$ on these zero-expanded functions. This setup has two disadvantages [12]. Any function f whose domain is extended from $[0, a]$, to $\mathbb{R}$ by setting $f(x) = 0$ for $x \notin [0, a]$, is likely to be discontinuous at 0 and a, even if the function itself is continuous on the original domain. This discontinuity will necessarily introduce some large wavelet

coefficients near the discontinuity. To overcome this difficulty Adcock et al. samples functions with a wider support. That is

$$\left\{ f \in \mathcal{L}^2(\mathbb{R}) : \operatorname{supp}(f) \subseteq [-T_1, T_2] \right\} \supseteq \operatorname{Closure}\left(\operatorname{span}\{\rho_j \in \Lambda_a\}\right)$$

In practice this means that not all the sampled functions f can be represented by this wavelet basis.

The other disadvantage with this construction is that it uses $2^j a + a - 1$ wavelets in each of the detail spaces W_j, $j \geqslant 0$, for some $a \in \mathbb{N}$ which depends on ν. Ideally we would like the size of each of these spaces to be independent of ν and contain only 2^j wavelets.

To eliminate these drawbacks we shall apply the boundary wavelets introduced in section 3.3. We recall that the orthonormal boundary wavelet basis on $[0,1]$ was defined through the direct sum of the spaces $V_{j_0}^{\text{int}} \oplus_{j=j_0}^{\infty} W_j^{\text{int}}$ where each subspace V_j^{int} and W_j^{int} consisted of 2ν boundary basis functions and $2^j - 2\nu$ interior basis functions. That is

$$W_j^{\text{int}} = \{\psi_{j,0}^{\text{left}}, \ldots, \psi_{j,\nu-1}^{\text{left}}, \psi_{j,\nu}, \ldots, \psi_{j,2^j-\nu-1}, \psi_{j,2^j-\nu}^{\text{right}}, \ldots, \psi_{j,2^j-1}^{\text{right}}\}$$

and similarly for $V_{j_0}^{\text{int}}$. From equation (3.8) we learned that these boundary wavelets and boundary scaling functions where created as finite linear combinations of $\phi_{j,k}$ and $\psi_{j,k}$. All of these boundary basis functions will therefore possess the same Lipschitz regularity α as ϕ and ψ [12].

The Fourier sampling basis on $[0,1]$ will consist of the basis functions

$$\varphi_k(x) = e^{-2\pi i k x} \mathbb{1}_{[0,1]}(x),$$

where we have used the notation

$$\mathbb{1}_I(x) = \begin{cases} 1 & \text{if } x \in I \subseteq \mathbb{R} \\ 0 & \text{otherwise} \end{cases}$$

to denote the step function on some interval $I \subseteq \mathbb{R}$.

The final step in the creation of an asymptotically incoherent matrix $\mathbf{U}$, is to order the Fourier sampling vectors according to increasing frequencies. Thus, we let $\tilde{\varphi}_1 = \varphi_0$, $\tilde{\varphi}_{2n} = \varphi_n$ and $\tilde{\varphi}_{2n+1} = \varphi_{-n}$. Further we let ρ_n be element n in the following ordering of the boundary wavelet functions

$$\{\phi_{j_0,0}^{\text{int}}, \ldots, \phi_{j_0,2^{j_0}-1}^{\text{int}}, \psi_{j_0,0}^{\text{int}}, \ldots, \psi_{j_0,2^{j_0}-1}^{\text{int}}, \psi_{j_0+1,0}^{\text{int}}, \ldots\} \tag{5.3}$$

found in the spaces $V_{j_0} \oplus_{j=j_0}^{\infty} W_j^{\text{int}}$. Applying this setup we are able to state to following incoherence theorem.

Theorem 5.7 ([2, Thm. 7.15])
Let $\mathbf{U}$ *be the isometry corresponding to the product between the Fourier sampling basis* $\{\tilde{\varphi}_n\}_{n\in\mathbb{N}}$ *and the orthonormal boundary wavelet basis* $\{\rho_n\}_{n\in\mathbb{N}}$. *That is*

$$\mathbf{U} = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \ldots \\ u_{21} & u_{22} & u_{23} & \ldots \\ u_{31} & u_{32} & u_{33} & \ldots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \qquad u_{ij} = \langle \rho_i, \tilde{\varphi}_i \rangle .$$

We then have

$$\mu(\mathbf{P}_K^{\perp}\mathbf{U}) \leqslant \frac{C^2_{\phi^{int},\psi^{int}}}{K\pi(2\alpha+1)\left(1+1/(2\alpha+1)\right)^{2(1+\alpha)}} \tag{5.4}$$
$$\mu\left(\mathbf{U}\mathbf{P}_K^{\perp}\right) \leqslant 2K^{-1}||\widehat{\psi}^{int}||^2_{\infty}.$$

Where $C_{\phi^{int},\psi^{int}}$ is some constant independent of K.

Proof. From [14, p. 216] we know that the Fourier transform of a function f which is uniformly Lipschitz $\alpha \geqslant 0$ is bounded by

$$|\widehat{f}(\omega)| = \frac{C_1}{(1+|\omega|)^{1+\alpha}}$$

Thus, as ϕ and ψ are Lipschitz α we have that

$$|\widehat{\phi}(\omega)| \leqslant \frac{C_\phi}{(1+|\omega|)^{1+\alpha}} \quad \text{and} \quad |\widehat{\psi}(\omega)| \leqslant \frac{C_\psi}{(1+|\omega|)^{1+\alpha}}. \tag{5.5}$$

We recall that the ν first and ν last boundary scaling functions and wavelets are not translated by any integer k. That is $\phi^{\text{int}}_{j_0,k}(x) = 2^{j_0/2}\phi^{\text{left}}_k(2^{j_0}x)$ for $k = 0,\dots,\nu-1$, and similarly for the ν last functions. For $k = \nu,\dots,2^{j_0}-\nu-1$ we have the usual translation $\phi^{\text{int}}_{j_0,k}(x) = 2^{j_0/2}\phi(2^{j_0}x-k)$. Due to this inconsistency we need to treat the two cases separately. For $k = \nu,\dots,2^{j_0}-\nu-1$ we have the following relation

$$\begin{aligned}
\left|\left\langle \phi^{\text{int}}_{j_0,k}, \varphi_n \right\rangle\right| &= \left|\int_0^1 \phi^{\text{int}}_{j_0,k}(x) e^{2\pi i n x}\,\mathrm{d}x\right| \\
&= \left|\int_0^1 2^{j_0/2}\phi^{\text{int}}\left(2^{j_0}x-k\right)e^{2\pi i n x}\,\mathrm{d}x\right| \\
&= 2^{-j_0/2}\left|\widehat{\phi}^{\text{int}}\left(\frac{-2\pi n}{2^{j_0}}\right)e^{2\pi i n k}\right| = 2^{-j_0/2}\left|\widehat{\phi}^{\text{int}}\left(\frac{-2\pi n}{2^{j_0}}\right)\right|
\end{aligned}$$

Similarly for $k = 0,\dots,\nu-1, 2^{j_0}-\nu,\dots,2^{j_0}-1$ we have

$$\begin{aligned}
\left|\left\langle \phi^{\text{int}}_{j_0,k}, \varphi_n \right\rangle\right| &= \left|\int_0^1 \phi^{\text{int}}_{j_0,k}(x) e^{2\pi i n x}\,\mathrm{d}x\right| \\
&= \left|\int_0^1 2^{j_0/2}\phi^{\text{int}}_k\left(2^{j_0}x\right)e^{2\pi i n x}\,\mathrm{d}x\right| \\
&= 2^{-j_0/2}\left|\widehat{\phi}^{\text{int}}_k\left(\frac{-2\pi n}{2^{j_0}}\right)\right|.
\end{aligned}$$

The exact same computations can be made for $\psi^{\text{int}}_{j,k}$, $j \geqslant j_0$. As all of these

have the same regularity α it follows that

$$\begin{aligned}
\mu\left(\mathbf{P}_K^{\perp}\mathbf{U}\right) &\leqslant \sup_{|n|\geqslant K/2} \max_{\substack{\rho\in \\ V_{j_0}^{\text{int}} \oplus_{j=j_0}^{\infty} W_j^{\text{int}}}} |\langle \rho, \varphi_n \rangle|^2 \\
&= \max\left\{ \sup_{|n|\geqslant K/2} \max_{j\geqslant j_0} 2^{-j} \left|\widehat{\psi}^{\text{int}}\left(\frac{-2\pi n}{2^j}\right)\right|^2, \sup_{|n|\geqslant K/2} 2^{j_0} \left|\widehat{\phi}^{\text{int}}\left(\frac{-2\pi n}{2^{j_0}}\right)\right|^2 \right\} \\
&= \max_{|n|\geqslant K/2} \max_{j\geqslant j_0} 2^{-j} \frac{C_{\phi^{\text{int}},\psi^{\text{int}}}^2}{\left(1+|2\pi n 2^{-j}|\right)^{2(1+\alpha)}} \\
&= \max_{j\geqslant j_0} 2^{-j} \frac{C_{\phi^{\text{int}},\psi^{\text{int}}}^2}{\left(1+|\pi K 2^{-j}|\right)^{2(1+\alpha)}}. \qquad (5.6)
\end{aligned}$$

Here $C_{\phi^{\text{int}},\psi^{\text{int}}} = \max\{C_\phi, C_{\phi_k^{\text{left}}}, C_{\phi_k^{\text{right}}}, C_\psi, C_{\psi_k^{\text{left}}}, C_{\psi_k^{\text{right}}}\}$ is the constants know from equation (5.5). The last equality (5.6) can be described as the function $f(x) = x^{-1}\left(1+\pi K x^{-1}\right)^{-2(1+\alpha)}$ whose derivative satisfies $f'(K\pi(2\alpha+1)) = 0$ for $x \in [1,\infty]$. Thus

$$\mu\left(\mathbf{P}_K^{\perp}\mathbf{U}\right) \leqslant \frac{C_{\phi^{\text{int}},\psi^{\text{int}}}^2}{K\pi(2\alpha+1)\left(1+1/(2\alpha+1)\right)^{2(1+\alpha)}}.$$

Similarly, we get for $2^p \leqslant K < 2^{p+1}$, $p \in \mathbb{N}$ with $p \geqslant j_0$.

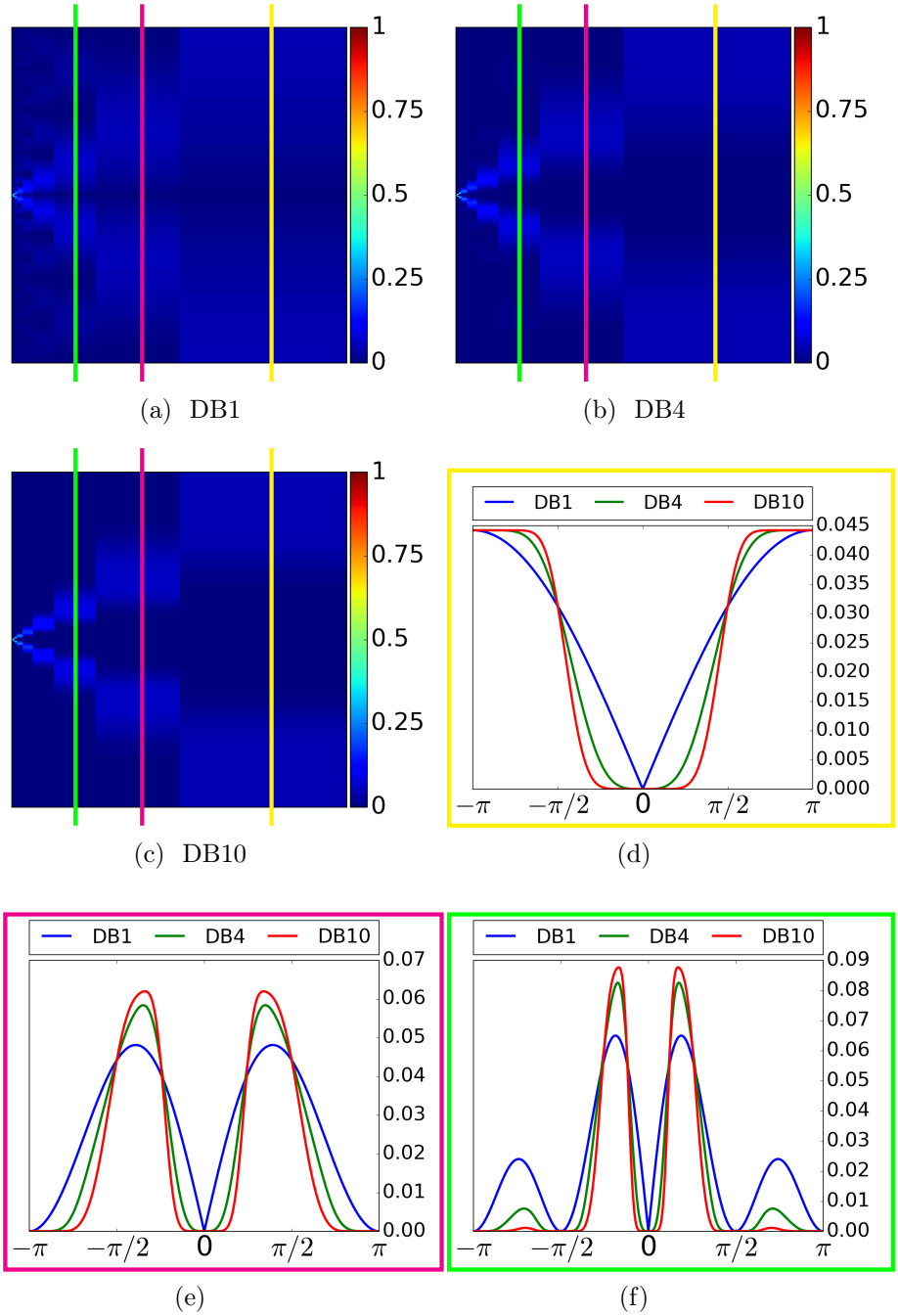
$$\begin{aligned}
\mu\left(\mathbf{U}\mathbf{P}_K^{\perp}\right) &= \max_{k\geqslant K} \max_{n\in\mathbb{Z}} |\langle \rho_k, \varphi_n \rangle|^2 \\
&= \max_{j\geqslant p} \max_{n\in\mathbb{Z}} \frac{1}{2^j} \left|\widehat{\psi}^{\text{int}}\left(\frac{-2\pi n}{2^j}\right)\right|^2 \\
&= 2^{-p} ||\widehat{\psi}^{\text{int}}||_\infty^2 = \frac{2||\widehat{\psi}^{\text{int}}||_\infty^2}{K}
\end{aligned}$$

□

Due to this theorem, we see that Fourier measurements sparsified by a Daubechies wavelet basis yield an asymptotically incoherent sampling basis. We also see from equation (5.4) that the coherence will decrease in each row for increasing regularity. This has been verified numerically in figure 5.5. In this figure we have extracted three columns from the matrix $\mathbf{U} = \mathbf{V}_{\text{dft}}\mathbf{\Psi}$ for three different number of vanishing moments. As see from the plots in (d,e,f), the asymptotic coherence decreases for increasing regularity.

5.3 Numerical experiments

Theorem 5.6 suggests that any sampling pattern should be signal dependent. In particular, one should fully sample all parts of a signal which is non-sparse and/or have high local coherence. On the other hand, one could reduce the number of samples for sparse parts of the signal, and in areas which are locally incoherent. In this section we intend to show numerically that this theorem coincides with practical experiments.



(a) DB1 (b) DB4

(c) DB10 (d)

(e) (f)

Figure 5.5: (a,b,c) Plot of $|\mathbf{V}_{\text{dft}}\mathbf{\Psi}^{-1}|$, $N = 2^{10}$, for various number of vanishing moments. The three colored lines indicate the cross-section of the matrices which have been extracted and plotted in the corresponding colored boxed found in (d,e,f).

Flip test

We know from practical experience using Fourier and Hadamard sampling that most of the signal energy will be contained within the first low frequent samples. This is due to the high correlation between low frequencies and homogeneous areas found in most signals. An example of this can be seen in figure 5.6, where the absolute values of the different domains are plotted. Due to the large magnitude of these low frequency samples, any recovery scheme failing to recover these coefficients will find the recovered signal unrecognizable.

From theorem 5.7 we know that all of these low frequencies have a local coherence of $\mathcal{O}(1)$. Thus, according to theorem 5.6 it will require full sampling rate to recover them.

Suppose we were in possession of a signal where the structure described above for the Fourier and Hadamard domain where not present. That is a signal where the low frequencies did not possess the most significant coefficients. In that case, the low-pass filtering performed by the DWT would produce coefficients close to zero, while the high-pass filters would give large magnitude coefficients. This would cause the sparsity structure of the signal to change.

According to theorem 5.6, this would imply that we would have to change the sampling pattern $\Omega_{\boldsymbol{s},\mathbf{M}}$. Note, however, that this would contradict principles such as the restricted isometry property and the null space property. According to these principles we could recover any s-sparse signal with the same error, regardless of the sparsity pattern and the sampling pattern. Thus we will show numerically that for the setup we have used, none of these old principles apply.

To test whether the recovery process is structure-dependent, as suggested by theorem 5.6, we will create a signal where the significant coefficients are at the lower half of the signal. This is done by reversing the DWT coefficients of a signal, and transforming the coefficients back to its original domain. Next, one can try to recover this new image through the usual setup. This is known as the *flip test*. A detailed description is presented below.

Let $\boldsymbol{c} = \boldsymbol{\Psi}\boldsymbol{x}$ be the wavelet coefficients of the signal $\boldsymbol{x} \in \mathbb{C}^N$ and let $\mathbf{T} \in \mathbb{R}^{N\times N}$ be the permutation matrix which reverses all vectors of length N. That is $\boldsymbol{c}' = \mathbf{T}\boldsymbol{c}$ will be the reversed wavelet coefficients of $\boldsymbol{x}$, where $c'_1 = c_N$, $c'_2 = c_{N-1}$, $\ldots$, $c'_N = c_1$. Suppose we sample this reversed signal $\boldsymbol{x}' = \boldsymbol{\Psi}^{-1}\boldsymbol{c}'$, solve $(P_{1,\eta,\Omega})$ and find the recovered reversed wavelet coefficients $\boldsymbol{c}^{\sharp\prime}$. If the recovery process was successful we would expect that by reversing the recovered wavelet coefficients back again, and transforming them to its original domain would yield the same reconstruction as the unflipped signal. Thus if we let $\boldsymbol{\Psi}^{-1}\mathbf{T}\boldsymbol{c}^{\sharp\prime} = \boldsymbol{x}_1^\sharp$ be the recovered flipped signal and let $\boldsymbol{\Psi}\boldsymbol{c}^\sharp = \boldsymbol{x}_2^\sharp$ be the recovered unflipped signal, we would expect $\boldsymbol{x}_1^\sharp \approx \boldsymbol{x}_2^\sharp$. As we can see from figure 5.7 this is not the case for both Fourier and Hadamard sampling.

As is evident from the test seen in figure 5.7, is that the optimal subsampling strategy for these sampling bases is signal dependent. It is also obvious that none of these bases satisfies the RIP or the NSP.

Two-dimensional compressive sensing

The setup seen in figure 5.7 involve two-dimensional signals, namely images. For such setups there exists at least two possible extensions of the theory we

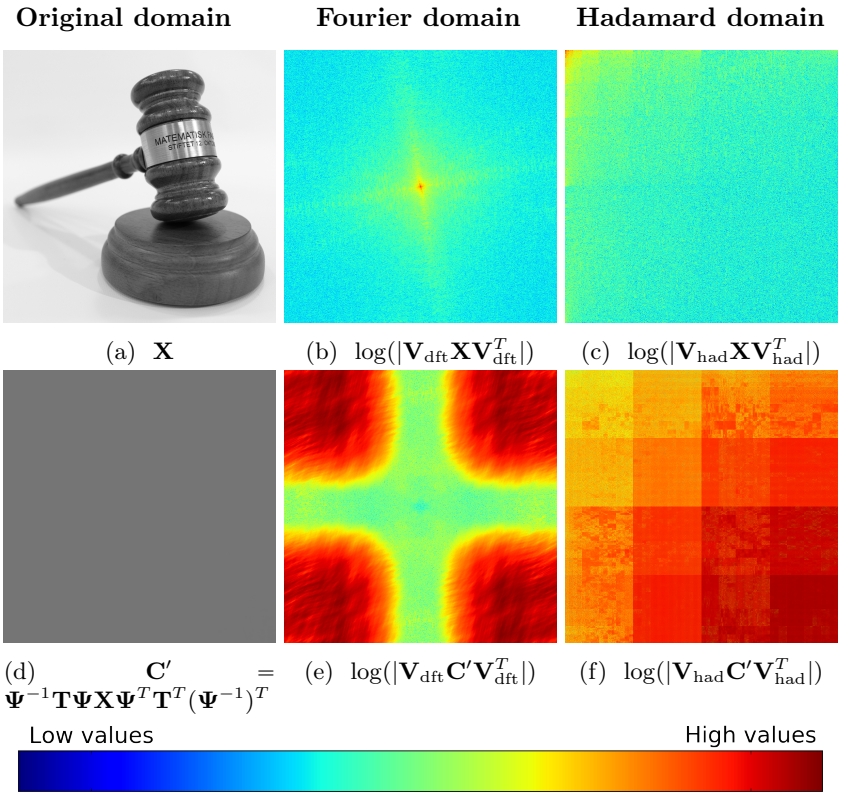


Figure 5.6: Top row: The unflipped image in the three domains. Bottom row: The image of the flipped wavelet coefficients in the three domains.

have discussed so far. The simplest of these extensions reshapes the signal into one-dimension, and apply the one-dimensional theory we have already discussed. This will destroy any structure in either the horizontal or vertical direction, depending on how we reshape the signal.

The other solution is to extend our sampling basis and sparsifying basis by the use of tensor products. This means that we will need to perform the sampling and sparsifying transforms which know from the one-dimensional theory, to both the columns and the rows of the images. Written in matrix notation, we end up with the following system

$$\mathbf{P}_\Omega \left(\mathbf{V}\mathbf{\Psi}^{-1}\right) \otimes \left(\mathbf{\Psi}^{-1}\mathbf{V}\right) \text{vec}(\mathbf{X}) = \boldsymbol{b} \tag{5.7}$$

where

$$\text{vec}(\mathbf{X}) = [x_{11}, \ldots, x_{1N}, x_{21}, \ldots, \ldots, x_{NN}]^T$$

denotes the vectorization of a matrix by placing the matrix columns into a single vector, and $\otimes$ is the Kronecker product.

To speed up numerical simulations, we note that the formulation in equation (5.7) is equivalent to the following equation

$$\mathbf{V}\mathbf{\Psi}^{-1}\mathbf{X}(\mathbf{\Psi}^{-1})^T\mathbf{V}^T = \mathbf{B} \tag{5.8}$$

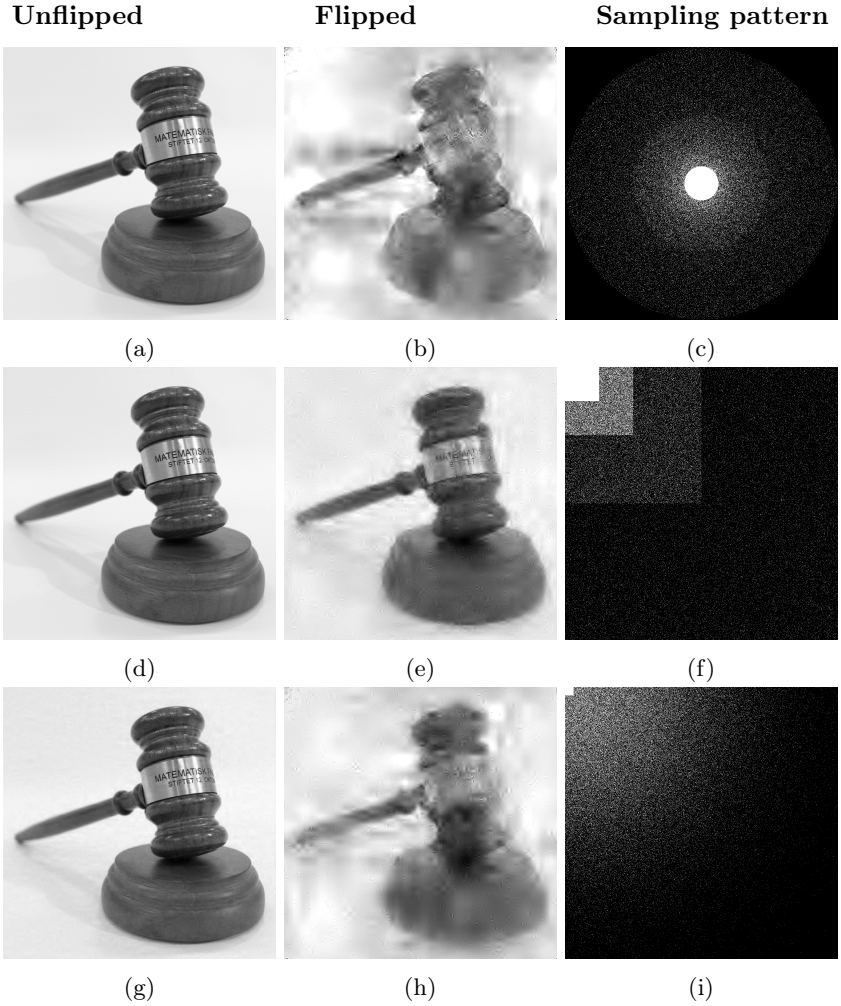


Figure 5.7: The flip test seen in practice. In the top row the same sampling pattern have been used to sample both the flipped and unflipped signal, using a Fourier sampling basis and the Daubechies 4 wavelet as sparsifying basis. In the two next rows a Hadamard sampling basis have been used in the same experiment, with two different sampling patterns.

where we have omitted the projection operator, as it is cumbersome to express in a two-dimensional setting. Using formulation (5.8) one can, however, apply the one-dimensional transforms in $\mathcal{O}(N \log_2 N)$ operations to each row and column, without explicitly creating a matrix. This saves memory and reduces the total number of floating point operations as the cost applying a $\mathcal{O}(N \log_2 N)$ transform N times is lower than applying a $\mathcal{O}(N^2 \log_2 N^2)$ one time.

In software packages such as SPGL1 [6], the setup described in (5.8) can be implemented as a matrix operator. This operator starts by reshaping the input vector into a square matrix, it then performs all the computations in the two-dimensional domain. The resulting output matrix is then vectorized before it is returned. This makes the algorithm observe the behaviour of (5.7) while the actual algorithm uses (5.8). This is how all of the operators used to create the CS-recovery images in this text, have been implemented.

Super-resolution

As we have already suggested, most real life signals are not sparse, but asymptotically sparse in wavelets. This means that the fraction of significant coefficients within each level decreases asymptotically for higher order resolution spaces. In addition, we know that the coherence in these high order resolution spaces are close to being incoherent. Using theorem 5.6, this implies that we can recover these higher order wavelet coefficients with very few samples.

On the other hand, if we sampled the signal at a lower resolution, then both the number of significant coefficients of the wavelet decomposition, and the coherence of the corresponding isometry would be higher. Hence, sampling a low resolution approximation would require a higher fraction of samples to obtain the same relative accuracy, as one obtains in a higher resolution system

This is verified in figure 5.8, where we have kept the fraction of samples fixed, and sampled the images at different resolutions. As one can see from the images, the error decreases for increasing resolution.

Error measurements To measure the error in an image reconstruction, can be a cumbersome task, as two images can differ largely in any norm without being different to the naked eye. Typical examples of this occurs if we translate all pixels in an image by a few pixels in the same direction, or if we add a small constant to all the pixel intensity values. Thus, in addition to viewing the actual mathematical error, one should also consider the visual error in the reconstruction.

For the error measurements used in figure 5.8, we have used the relative error between the sampled image and its reconstructed version. That is

$$\frac{||\mathbf{X} - \mathbf{X}^\sharp||_{\mathrm{F}}}{||\mathbf{X}||_{\mathrm{F}}},$$

where $||\cdot||_{\mathrm{F}}$ denotes the Frobenius norm. One could also argue that the error should be measured using the $\sigma_{\boldsymbol{s},\mathbf{M}}$ measure, since we are searching for a solution $\boldsymbol{c}^\sharp \in \Sigma_{\boldsymbol{s},\mathbf{M}}$ of $(P_{1,\eta,\Omega})$. This would be meaningful if the original signal was $(\boldsymbol{s},\mathbf{M})$-sparse or if the we had an estimate of the sparsity. In our setup, neither the original set of wavelet coefficients $\boldsymbol{c}$, nor the wavelet coefficients recovered by the SPGL1 algorithm $\boldsymbol{c}^\sharp$ will be $(\boldsymbol{s},\mathbf{M})$-sparse. Instead both of them will be compressible. This makes the lower bound of $\sigma_{\boldsymbol{s},\mathbf{M}}$ cumbersome to apply.

Experimental setup All images used in this text have been captured by a conventional digital camera. These images was captured as raw images, without any image compression. The images have then been cropped into a 2048×2048 image and stored in a lossless format. To obtain the lower resolution images of dimension 1024×1024, 512×512 and 256×256 the images was resampled using the GNU Image Manipulation Program (GIMP) [22]. This program implements a resampling strategy based on cubic interpolation, which the author did not bother to implement himself. All of the Fourier and Hadamard sampling have then been applied to these digital images.

This setup have been used so that the results would not be overly optimistic. If we instead had sampled an image stored in a lossy format, most of the low magnitude coefficients would already have been truncated to zero. This would have increased the signal sparsity, and simplified the reconstruction.

The author have not been able to test compressive sensing on 4096×4096 or larger images, due to the limitation in camera technology. The author could of course used the conventional approach of image interpolation to increase the size further [24, sec 2.4]. This would, however, also introduced sparsity as our wavelet functions are orthonormal to polynomials.

5.4 Infinite dimensional compressive sensing

In most papers considering infinite-dimensional compressive sensing one does only consider the Fourier sampling basis on $\mathcal{L}^p$-spaces or some general sampling basis on a Hilbert space [1, 11]. In this section, we will review this theory using a Walsh sampling basis. The presentation will be non-exhaustive and is only meant as a motivation for the coherence estimates for infinite-dimensional matrices. Before we study this theory, we will start out by a simplified description of how one obtains a continuous Hadamard sample. In this discussion we will omit any technical details concerning the point spread functions blurring effect on the signal [32], photonic noise [31], the required bit depth of the sensor [17], and the leakage of light between the sampling mirrors [34].

The Hadamard sampling matrix can be used in any application where one can obtain binary samples. Typical examples would be example 1.1 with a counterfeit coin, single-pixel imaging [17, 35] and fluorescence microscopy [34]. In the two latter approaches the measurements are obtained from a continuous two-dimensional signal. In these setups the light from the object being imaged is directed towards a digital micro-mirror device (DMD), whose mirrors can be turn either on or off. The light from the active mirrors are then directed towards the sensor which produce an output voltage, whose strength represents the number of photons the sensor perceived during exposure. A sketch of this setup can be seen in figure 5.9.

This creates measurements b_i, which are realizations of the inner product $\langle \boldsymbol{a}_i, \boldsymbol{x} \rangle = b_i$ between the signal $\boldsymbol{x} \in \mathbb{R}^N$ and a vector $\boldsymbol{a}_i \in \{0,1\}^N$, whose entries are either 0 or 1, depending on whether or not the corresponding mirror on the DMD was turned on or off. To capture a row in the Hadamard matrix, one start by obtaining a measurement $b_{\mathbf{1}}$ with all mirrors on. This can be modeled as a sample from the sampling vector $\mathbf{1} = [1, 1, \ldots, 1]^T$. By applying the linearity of the inner product, one can then obtain a row from the Hadamard matrix as

$$\langle \mathbf{1} - 2\boldsymbol{a}_i, \boldsymbol{x} \rangle = \langle \mathbf{1}, \boldsymbol{x} \rangle - \langle 2\boldsymbol{a}_i, \boldsymbol{x} \rangle = b_{\mathbf{1}} - 2b_i$$

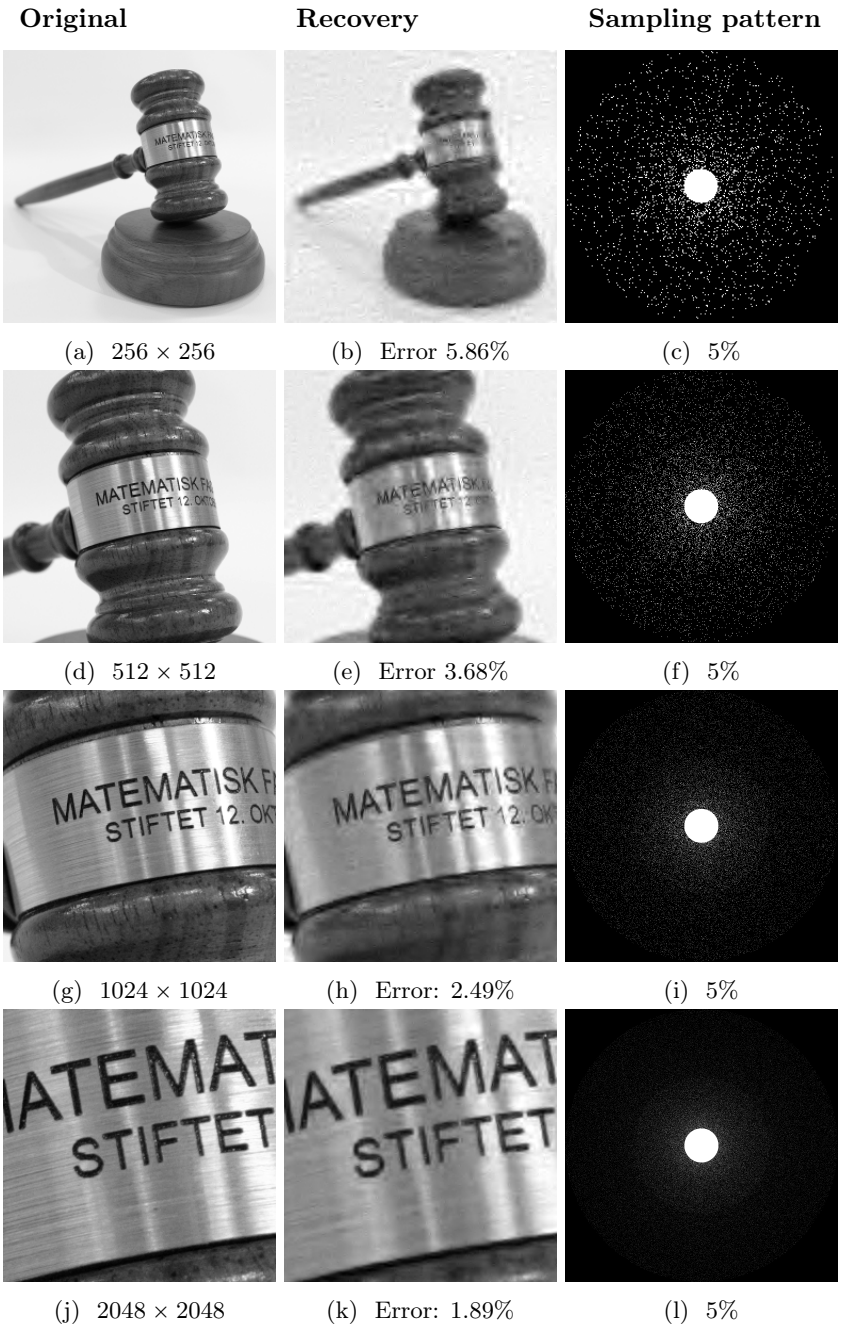


Figure 5.8: The relative error using Fourier sampling with 5% subsampling and the DB-4 wavelet. The error is measured using the Frobenius norm.

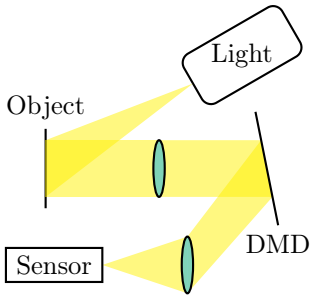


Figure 5.9: Conceptual sketch of single-pixel imaging.

by choosing an appropriate pattern of the 1's in $\boldsymbol{a}_i$. There are of course other ways to create this sampling pattern from 0, 1 patterns, all of which use the linearity of the inner product.

In the applications described above, the samples came from two-dimensional signals, but to simplify the discussion we will assume the sampled signal is one dimensional i.e. $f\colon [0,1] \to \mathbb{R}$. This means that the measurements we observe in any of these applications come from the continuous integral

$$\breve{f}(n) = \int_0^1 f(x) w_n(x)\,\mathrm{d}x$$

where w_n denotes the Paley enumerated Walsh function.

If we applied these samples directly in a finite-dimensional setup, this would lead to *measurement mismatch* [11] and the wavelet crime. To see this let $\breve{\boldsymbol{f}}_N = [\breve{f}(0), \ldots, \breve{f}(N-1)]^T \in \mathbb{R}^N$ be the N first continuous Walsh measurements. By multiplying $\mathbf{V}_{\text{had}}^T \breve{\boldsymbol{f}} = \boldsymbol{x}$ one will project these samples onto a N-point grid on $[0,1]$. That is, the values of $\boldsymbol{x}$ will be given by the function $f_N(x) = \sum_{k=0}^{N-1} \breve{f}(k) w_k(x)$ [31].

If f is a smooth function, it will be sparse in wavelets. In the setup we have presented above this can not happen, since we approximate f by the truncated Walsh-series f_N. These truncated series is spanned by the discontinuous Walsh functions, which cannot be sparse in wavelets. Note, that this is not an issue in the experiments we have presented above as these measurements emerged from the Hadamard matrix $\mathbf{V}_{\text{had}}$, rather than the continuous samples $\breve{f}(k)$ [31]. This is known as the *inverse crime* [25], and by committing it we will obtain artificially good results.

To solve these problems we will consider the recovery problem of f as infinite-dimensional. To do this, we recall from equation (5.3) that we let $\{\rho_n\}_{n\in\mathbb{N}}$ denote the boundary wavelet basis belonging to the space $V_{j_0}^{\text{int}} \oplus_{j=j_0}^{\infty} W_j^{\text{int}}$. Next let

$$\mathbf{U} = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \ldots \\ u_{21} & u_{22} & u_{23} & \ldots \\ u_{31} & u_{32} & u_{33} & \ldots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \qquad u_{ij} = \langle w_i, \rho_j \rangle,$$

be the isometry of $\ell^2(\mathbb{N})$ consisting of the sampling and sparsifying bases. Further let $\breve{\boldsymbol{f}} = [\breve{f}(0), \breve{f}(1), \ldots]^T$ be the infinite vector of Walsh samples of f. Then

the problem we are trying to solve can be described in infinite-dimensions as

$$\inf_{\boldsymbol{z}\in\ell^1(\mathbb{N})} ||\boldsymbol{z}||_1 \quad \text{subject to} \quad \mathbf{P}_\Omega \mathbf{U}\boldsymbol{z} = \mathbf{P}_\Omega \check{\boldsymbol{f}}$$

To solve this on a computer, we must project the problem into finite dimensions. The simplest approach would be to approximate $\mathbf{U}$ by the matrix $\mathbf{V}_{\text{had}}\boldsymbol{\Psi}^{-1} \in \mathbb{R}^{N\times N}$, but this would cause all of the problems presented above. Another approach would be to project $\mathbf{U}$ onto $\mathbb{R}^{N\times N}$ by $\mathbf{P}_N^0\mathbf{U}\mathbf{P}_N^0$. Using this approach would however ruin the isometry property of $\mathbf{U}$. As an example one could find that for a Fourier sampling basis this would create a condition number of 10^6 [1].

The solution to the discretization problem above is to choose a $\tilde{N}$ and $\tilde{M} \leqslant \tilde{N}$ according to the *balancing property* [1]. This will ensure that the finite matrix $\mathbf{P}_{\tilde{N}}\mathbf{U}\mathbf{P}_{\tilde{M}}$ preserves the required amount of isometric structure. Hence, if we let $\Omega \subset \{1,\dots,\tilde{N}\}$, $|\Omega| \leqslant m$ and solve the finite dimensional problem

$$\inf_{\boldsymbol{z}\in\mathbf{P}_{\tilde{M}}^0(\ell^1(\mathbb{N}))} ||\boldsymbol{z}||_1 \quad \text{subject to} \quad \mathbf{P}_\Omega\mathbf{P}_{\tilde{N}}^0\mathbf{U}\mathbf{P}_{\tilde{M}}^0\boldsymbol{z} = \mathbf{P}_\Omega\check{\boldsymbol{f}} \tag{5.9}$$

we will see a substantial gain in performance. This is because we formulate the problem as infinite-dimensional and then discretize, rather than applying the discrete formulation directly. Examples illustrating the success of this theory for a Fourier sampling basis can be found in [1, 31].

CHAPTER 6

Coherence between Hadamard and orthonormal wavelets

In chapter 5, we saw that Hadamard matrices were asymptotically incoherent to the Haar wavelet matrix. In this chapter, we will generalize this result to all compactly supported orthonormal Daubechies wavelets on an interval. As we have already seen in figure 5.7, these wavelets combined with a Hadamard sampling passed the flip test. This indicated that there is at least some incoherence between the bases.

We will start this chapter with the actual coherence result, before we show some numerical simulations indicating their accuracy. Finally, we end this chapter with a short summary of this thesis, indicating how this result fits into the rest of the theory we have presented.

6.1 Asymptotic coherence estimate

Lemma 6.1
Let ψ be a wavelet with $supp(\psi) \subseteq [0,1]$ and let $\psi_{j,k} = 2^{j/2}\psi(2^j x - k)$ for $j,k \in \mathbb{Z}_+$ and $0 \leqslant k < 2^j$. Then

$$\langle \psi_{j,k}, w_n(x) \rangle = 2^{-j/2} w_n\left(\frac{k}{2^j}\right) \check{\psi}\left(\left\lfloor \frac{n}{2^j} \right\rfloor\right).$$

Proof.

$$\begin{aligned}
\langle \psi_{j,k}, w_n(x) \rangle &= \int_0^1 \psi_{j,k}(x) w_n(x)\,\mathrm{d}x \\
&= \int_0^1 2^{j/2}\psi\left(2^j x - k\right) w_n(x)\,\mathrm{d}x \\
&= 2^{j/2}\int_{2^{-j}k}^{2^{-j}(k+1)} \psi\left(2^j x - k\right) w_n(x)\,\mathrm{d}x \\
&= 2^{-j/2}\int_0^1 \psi(x) w_n\left(\frac{x}{2^j} + \frac{k}{2^j}\right)\,\mathrm{d}x \\
&= 2^{-j/2}\int_0^1 \psi(x) w_n\left(\frac{x}{2^j} \oplus \frac{k}{2^j}\right)\,\mathrm{d}x \qquad (6.1)
\end{aligned}$$

$$= 2^{-j/2} w_n\left(\frac{k}{2^j}\right) \int_0^1 \psi(x) w_n\left(\frac{x}{2^j}\right) \mathrm{d}x \tag{6.2}$$
$$= 2^{-j/2} w_n\left(\frac{k}{2^j}\right) \breve{\psi}\left(\left\lfloor \frac{n}{2^j} \right\rfloor\right)$$

where we in equation (6.1) and (6.2) used lemma 4.7 and proposition 4.9a, respectively. □

In short, this lemma limits the size of the coefficients on the right hand side of the isometry $\mathbf{U} = \mathbf{V}_{\mathrm{had}} \boldsymbol{\Psi}^{-1}$. The factor $w_n(\cdot)$ is negligible as it is either -1 or 1, while we would still need to limit the factor $\breve{\psi}(\lfloor 2^{-j} n \rfloor)$ in order to get an estimate of the coefficients in the lower half of $\mathbf{U}$. The next lemma will do just this.

Lemma 6.2
Let $\psi\colon [0,1] \to \mathbb{R}$ be uniformly Lipschitz $\alpha > 0$. Next let $n, p \in \mathbb{Z}_+$ be such that $2^p \leqslant n < 2^{p+1}$. Then for $0 < \alpha < 1$

$$\breve{\psi}(n) = \int_0^1 \psi(x) w_n(x)\, \mathrm{d}x \leqslant C 2^{-p\alpha}$$

and for $\alpha \geqslant 1$

$$\breve{\psi}(n) = \int_0^1 \psi(x) w_n(x)\, \mathrm{d}x \leqslant C 2^{-p}$$

where C is some constant independent of n.

Proof. To simplify notation let $\Delta_k^p = [2^{-p}k, 2^{-p}(k+1))$. First we consider $0 < \alpha < 1$. Due to the Lipschitz regularity we know that there exists a constant C such that $\psi(x) \leqslant \psi(s) + C|s - x|^\alpha$ for any $s \in [0,1]$. Hence,

$$\sup_{x \in \Delta_k^p} \psi(x) \leqslant \psi\left(2^{-p}k + 2^{-(p+1)}\right) + C 2^{-(p+1)\alpha}$$
$$\sup_{x \in \Delta_k^p} -\psi(x) \leqslant -\psi\left(2^{-p}k + 2^{-(p+1)}\right) + C 2^{-(p+1)\alpha}.$$

From lemma 4.8 we know that on each interval Δ_k^p, w_n is constant equal to 1 on one of the subintervals Δ_{2k}^{p+1}, Δ_{2k+1}^{p+1}, and equal to -1 on the other. Hence,

$$\begin{aligned}\left|\int_{\Delta_k^p} \psi(x) w_n(x)\, \mathrm{d}x\right| \leqslant\ & 2^{-p} \Big| \left(\psi\left(2^{-p}k + 2^{-p-1}\right) + C 2^{-(p+1)\alpha}\right) \\ & + \left(-\psi\left(2^{-p}k + 2^{-p-1}\right) + C 2^{-(p+1)\alpha}\right) \Big| \\ \leqslant\ & 2^{-p} C 2^{-p\alpha}.\end{aligned}$$

Thus,

$$\begin{aligned}\int_0^1 \psi(x) w_n(x)\, \mathrm{d}x &= \sum_{k=0}^{2^p-1} \int_{\Delta_k^p} \psi(x) w_n(x)\, \mathrm{d}x \\ &\leqslant \sum_{k=0}^{2^p-1} 2^{-p} C 2^{-p\alpha} = C 2^{-p\alpha}.\end{aligned}$$

Next, consider the case where ψ is uniformly Lipschitz $\alpha \geqslant 1$ on $[0,1]$. Then the derivative $\psi'(x)$ exists for all $x \in [0,1]$. Thus by Taylors formula we know that for any point $s \in [0,1]$ we have

$$\psi(x) = \psi(s) + \psi'(t)(x-s)$$

for some t in the interval containing both s and x. Using this formula we know that

$$\begin{aligned}
\sup_{x \in \Delta_k^p} \psi(x) &\leqslant \psi\left(2^{-p}k + 2^{-(p+1)}\right) + \sup_{t \in [0,1]} |\psi'(t)| 2^{-(p+1)} \\
\sup_{x \in \Delta_k^p} -\psi(x) &\leqslant -\psi\left(2^{-p}k + 2^{-(p+1)}\right) + \sup_{t \in [0,1]} |\psi'(t)| 2^{-(p+1)}.
\end{aligned}$$

Hence, by applying the same technique as before, the result it self-evident. □

A valid question is whether this result can be sharpened for functions which are uniformly Lipschitz $\alpha \geqslant 2$. For such a function f, the corresponding Taylor polynomials would be

$$\begin{aligned}
\sup_{x \in \Delta_k^p} f(x) \leqslant & f\left(2^{-p}k + 2^{-(p+1)}\right) + f'\left(2^{-p}k + 2^{-(p+1)}\right) 2^{-(p+1)} \\
& + \sup_{x \in \Delta_k^p} \frac{|f''(t)|}{2} (2^{-(p+1)})^2 \\
\sup_{x \in \Delta_k^p} -f(x) \leqslant & - f\left(2^{-p}k + 2^{-(p+1)}\right) + f'\left(2^{-p}k + 2^{-(p+1)}\right) 2^{-(p+1)} \\
& + \sup_{x \in \Delta_k^p} \frac{|f''(t)|}{2} (2^{-(p+1)})^2
\end{aligned}$$

Here the two first derivatives have the same sign. This ruins the cancellation effects seen in the proof. This means that we can not expect to gain a lower coherence by increasing the regularity. Thus, the effect seen in figure 5.5 can not be expected. This has been verified numerically in figure 6.1, where we see that increasing regularity does not affect the coherence.

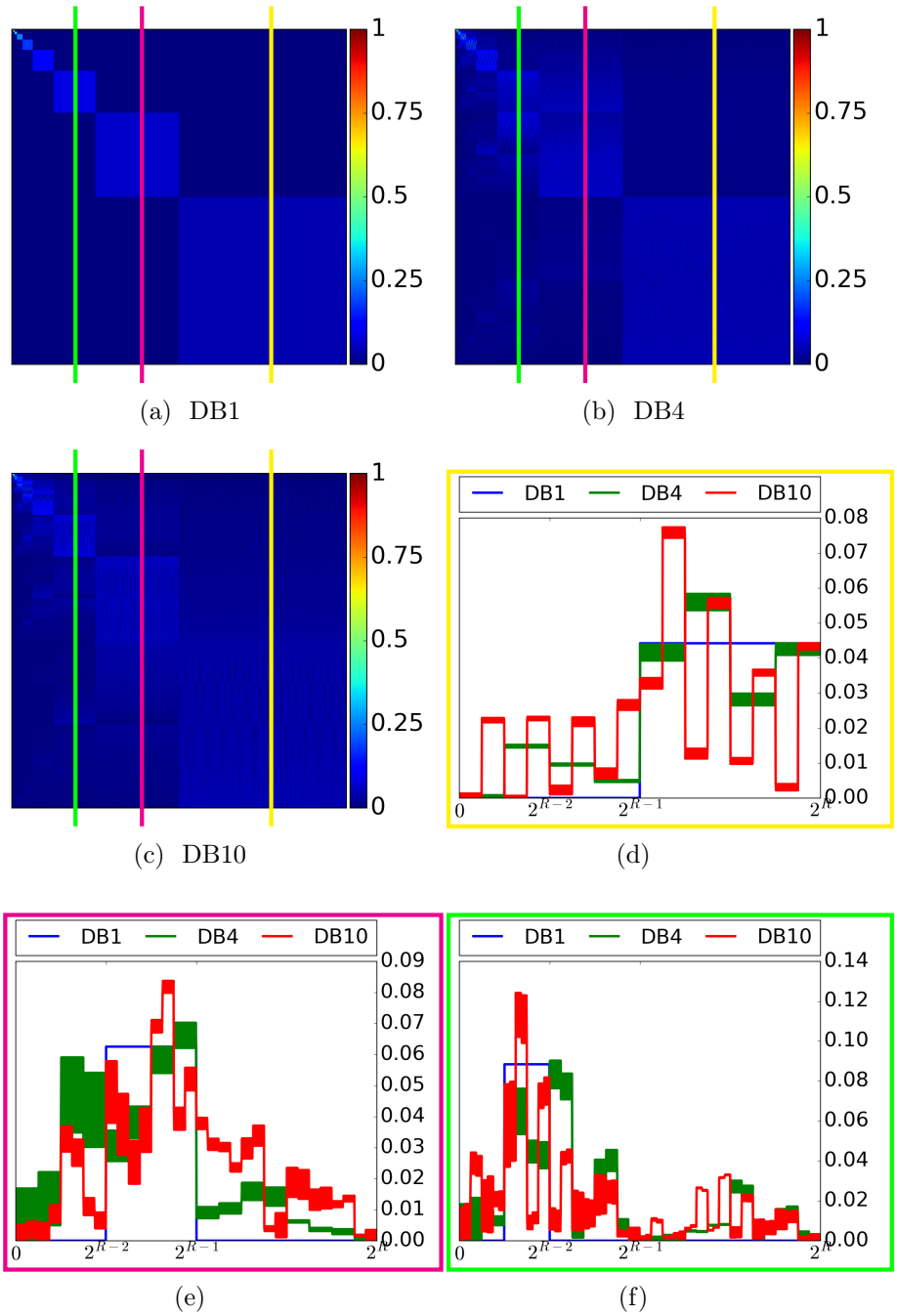
Next, we shall see that the regularity of a function is unaffected by scaling.

We have now proven all the necessary lemmas, in order to obtain the final coherence result between a Walsh sampling basis and Daubechies wavelets. Before we state the final theorem, we recall the ordering of the boundary wavelet basis introduced in equation (5.3). That is ρ_n denotes element n in the following ordering of the basis

$$\{\phi^{\text{int}}_{j_0,0}, \ldots, \phi^{\text{int}}_{j_0,2^{j_0}-1}, \psi^{\text{int}}_{j_0,0}, \ldots, \psi^{\text{int}}_{j_0,2^{j_0}-1}, \psi^{\text{int}}_{j_0+1,0}, \ldots\}.$$

Theorem 6.3
Let $\mathbf{U}$ *correspond to the isometry generated by the Paley enumerated sampling basis* $\{w_{n-1}\}_{n \in \mathbb{N}}$ *and the orthonormal boundary wavelet basis* $\{\rho_n\}_{n \in \mathbb{N}}$ *on the interval* $[0,1]$*, for a wavelet with* ν *vanishing moments and a uniform Lipschitz regularity* $\alpha > 0$*. That is*

$$\mathbf{U} = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \ldots \\ u_{21} & u_{22} & u_{23} & \ldots \\ u_{31} & u_{32} & u_{33} & \ldots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \qquad u_{ij} = \langle \rho_i, w_j \rangle$$



(a) DB1 (b) DB4

(c) DB10 (d)

(e) (f)

Figure 6.1: (a,b,c) Plot of $|\mathbf{V}_{\text{had}}\mathbf{\Psi}^{-1}|$, $N = 2^{10}$, for a sequency ordered Hadamard matrix and various number of vanishing moments. The three colored lines indicate the cross section of the matrices which have been extracted and plotted in each of the three colored boxed found in (d,e,f).

For $\beta = \min\{1, \alpha\}$ and $p \geqslant j_0$ we then have

$$\mu\left(\mathbf{P}_{2^p}^{\perp}\mathbf{U}\right) \leqslant K 2^{-p\beta}$$
$$\mu\left(\mathbf{U}\mathbf{P}_{2^p}^{\perp}\right) \leqslant 2^{-(p-j_0)} D^2,$$

where K is some constant independent of p and

$$D = \max\left\{\left|\left|\psi_{j_0,1}^{left}\right|\right|_\infty, \ldots, \left|\left|\psi_{j_0,\nu}^{left}\right|\right|_\infty, \left|\left|\psi_{j_0,1}^{right}\right|\right|_\infty, \ldots, \left|\left|\psi_{j_0,\nu}^{right}\right|\right|_\infty, ||\psi_{j_0}||_\infty\right\}.$$

Proof. The space V_{j_0} is spanned by 2ν boundary scaling functions and $2^{j_0} - 2\nu$ of the usual Daubechies scaling functions with a support strictly inside $[0, 1]$. Similarly the W_j, $j \geqslant j_0$, is spanned by 2ν boundary wavelets and $2^j - 2\nu$ of the usual Daubechies wavelets. Further, we know from equation (3.8) that all of these boundary functions can be written as linear combinations of the scaling and wavelet functions. Thus, they have the same regularity as ϕ and ψ.

From lemma 6.2 we know that

$$\breve{\phi}^{\text{int}}(n) \leqslant C 2^{-p\beta}, \qquad \breve{\psi}^{\text{int}}(n) \leqslant C 2^{-p\beta}$$

for n so that $2^p \leqslant n \leqslant 2^{p+1}$. For all $j \geqslant j_0$, with $j = j_1 + j_0$, we know that all of the functions $\phi_{j,k}^{\text{int}}$ and $\psi_{j,k}^{\text{int}}$ are supported on $[0, 1]$. The ν first and ν last of these functions, are not translated as the other $2^j - 2\nu$ wavelets and scaling functions. In the following, we are only interested in the absolute value of the Walsh transform of these functions. From lemma 6.1 this absolute value will be unaffected by the translation by k. To shorten the argument we will therefore assume all of them are translated by k. As the regularity of all these functions are the same, our use of lemma 6.2 will also be unaffected by this abuse of notation.

From lemma 6.1 we know that

$$\left|\left\langle \phi_{j_0,k}^{\text{int}}, w_n \right\rangle\right| \leqslant \left|w_n(k)\, \breve{\phi}_{j_0}^{\text{int}}(n)\right| = \left|\breve{\phi}_{j_0}^{\text{int}}(n)\right|$$
$$\left|\left\langle \psi_{j,k}^{\text{int}}, w_n \right\rangle\right| \leqslant 2^{-j_1/2} \left|w_n\left(\frac{k}{2^{j_1}}\right) \breve{\psi}_{j_0}^{\text{int}}\left(\left\lfloor \frac{n}{2^{j_1}} \right\rfloor\right)\right| = 2^{-j_1/2} \left|\breve{\psi}_{j_0}^{\text{int}}\left(\left\lfloor \frac{n}{2^{j_1}} \right\rfloor\right)\right|$$

Next note that due to lemma 6.2 we obtain the following relation for $2^p \leqslant n < 2^{p+1}$,

$$\begin{aligned}\left|\left\langle \psi_{j,k}^{\text{int}}, w_n \right\rangle\right|^2 &\leqslant 2^{-j_1} \left|\breve{\psi}_{j_0}^{\text{int}}\left(\left\lfloor \frac{n}{2^{j_1}} \right\rfloor\right)\right|^2 \\ &\leqslant \begin{cases} 2^{-j_1} D^2 & j_1 > p \\ 2^{-j_1} C 2^{-2(p-j_1)\beta} & j_1 \leqslant p \end{cases}.\end{aligned}$$

Here the last equality for $j \in [0, p]$ can be written as $2^{-2p\beta} C 2^{(2\beta-1)j_1}$. From table 3.1 we know that $\alpha \approx 0.55$ for $\nu = 2$, and that the regularity increases with the number of vanishing moments. This implies that $\beta > \frac{1}{2}$, which again implies that $2^{-2p\beta} C 2^{(2\beta-1)j_1}$ attains it maximum for $j_1 = 0$. This fact will be

used in the following calculations

$$\begin{aligned}
\mu\left(\mathbf{P}_{2^p}^{\perp}\mathbf{U}\right) &\leqslant \sup_{n>2^p} \max_{\substack{\rho\in \\ V_{j_0}^{\mathrm{int}}\oplus_{j=j_0}^{\infty} W_j^{\mathrm{int}}}} |\langle \rho, w_n\rangle|^2 \\
&= \max\left\{ \sup_{n>2^p} \max_{j_1\geqslant 0} 2^{-j_1} \left|\breve{\psi}_{j_0}^{\mathrm{int}}\left(\left\lfloor \frac{n}{2^{j_1}}\right\rfloor\right)\right|^2, \sup_{n>2^p}\left|\breve{\phi}_{j_0}^{\mathrm{int}}(n)\right|^2 \right\} \\
&= \max\left\{ \max_{j_1>p}\left(2^{-j_1/2}D\right)^2, \max_{0\leqslant j_1\leqslant p}\left(2^{-j_1/2}C2^{-(p-j_1)\beta}\right)^2, \max_{k\geqslant p} C2^{-2k\beta}\right\} \\
&\leqslant \max\left\{ D^2 2^{-(p+1)}, C2^{-2p\beta}\right\} \\
&\leqslant K2^{-p}
\end{aligned}$$

with $K = \max\{2^{-1}D^2, C\}$. Here we used the fact that $\beta > \frac{1}{2}$ to obtain the inequality $C2^{-2p\beta} < C2^{-p}$. Similarly we obtain

$$\begin{aligned}
\mu(\mathbf{U}\mathbf{P}_{2^p}^{\perp}) &= \sup_{i>2^p}\max_{n\in\mathbb{Z}_+} |\langle \rho_i, w_n\rangle|^2 \\
&\leqslant \sup_{\substack{j\geqslant p \\ 0\leqslant k<2^j}} \max_{n\in\mathbb{Z}_+} \left|\left\langle \psi_{j,k}^{\mathrm{int}}, w_n\right\rangle\right|^2 \\
&\leqslant \sup_{j_1\geqslant p-j_0}\max_{n\in\mathbb{Z}_+} 2^{-j_1}\left|\breve{\psi}_{j_0}^{\mathrm{int}}\left(\left\lfloor\frac{n}{2^{j_1}}\right\rfloor\right)\right|^2 \\
&\leqslant \sup_{j_1\geqslant p-j_0} 2^{-j_1}\left|\breve{\psi}_{j_0}^{\mathrm{int}}(0)\right|^2 \\
&\leqslant 2^{-(p-j_0)}\left(\int_0^1 |\psi_{j_0}^{\mathrm{int}}|\,\mathrm{d}x\right)^2 \\
&\leqslant 2^{-(p-j_0)}\left(\int_0^1 D\,\mathrm{d}x\right)^2 \\
&\leqslant 2^{j_0}D^2 2^{-p}
\end{aligned}$$

□

In the theorem above we have only considered the Paley enumerated Hadamard matrix. Note, however, that due to lemma 4.6 we have the following relation

$$\operatorname{span}\{\mathrm{WAL}(n,\cdot) : n\in\{2^p,\dots,2^{p+1}-1\}\} = \operatorname{span}\{\mathrm{PAL}(n,\cdot) : n\in\{2^p,\dots,2^{p+1}-1\}\},$$

for all $p\in\mathbb{N}$. Thus, the result should hold equally well for the sequency ordered Hadamard matrix.

6.2 Accuracy of the results

In the proof above we have only considered boundary wavelets, as these wavelets preserve all vanishing moments. Another approach would have been to create a periodic wavelet basis on $[0,1]$ or some other interval $[0,a]$, $a\geqslant 1$, as was done in [2]. The later approach would of course require us to extend the Walsh functions to R_+. This is done in e.g., [20] and [23, sec 1.5]. An advantage with this approach would of course be that we could decompose the space V_{j_0} further into the spaces $V_0\oplus_{j=0}^{j_0-1} W_j$

To verify the above results in a finite-dimensional setup, the two matrices $\left|\mathbf{V}_{\text{had}}\mathbf{\Psi}^{-1}\right| = |\mathbf{U}| \in \mathbb{C}^{N\times N}$, $N = 2^R$ can be multiplied together in order to find the maximum element in each of the rows and columns. This is done in figure 6.2 for the Daubechies wavelet with 2 and 3 vanishing moments. For both of these wavelets, a periodic wavelet basis and a boundary wavelet basis have been applied. As seen in the figure, some of these boundary wavelets have a higher constant than the interior wavelets. If we consider the plots of the wavelet functions for $\nu = 2$ vanishing moments, found in figure 3.2 and 3.4, we would find that $||\psi_1^{\text{left}}||_\infty \approx 2.4$, while $||\psi||_\infty \approx 1.8$. Approximately the same difference would have been found for boundary wavelets with $\nu = 3$ vanishing moments. It is therefore not surprising that both of these boundary wavelet bases have a higher constant than their periodic equivalence.

One thing which is surprising is the fact that it is the right boundary wavelet which creates these large constants. This can be seen by studying the $\mathbf{\Psi}^{-1}$ matrix, whose $N/2$ rightmost columns consists of the filter coefficients of g, shifted by two entries between each column. The ν last of these columns will consist of all the different g^{right} filters. Thus, by considering the absolute sum of these high-pass filters for $\nu = 2$ vanishing moments we see that

$$\begin{aligned}
\sum_n \left|g_1^{\text{left}}[n]\right| &= 1.6017, & \sum_n \left|g_2^{\text{right}}[n]\right| &= 2.0858,\\
\sum_n \left|g_2^{\text{left}}[n]\right| &= 1.6794, & \sum_n \left|g_2^{\text{right}}[n]\right| &= 1.5988,\\
\sum_n |g[n]| &= 1.6017
\end{aligned}$$

the filter g_2^{right} creates the high coherence. The sum of g_2^{right} is in fact so large, that the coherence of $\mu\left(\mathbf{U}\mathbf{P}_{2^p}^{\perp}\right)$ between level 2^{14} and 2^{15} is the same for a system of size $N = 2^{16}$. This can be seen in figure 6.3(b).

The same effect with constant coherence occur in figure 6.3(d) as well. This can be seen between the levels 2^{14} and 2^{15} for $\mu\left(\mathbf{P}_{2^p}^{\perp}\mathbf{U}\right)$. In this case the same is true for the periodic wavelet basis.

To plot the upper coherence boundaries found in theorem 6.3 is a cumbersome task, as an estimate of the constants C and D is needed. In the plots found in figure 6.3 we chose $D = 2.5$ for the boundary wavelet, while $D = 1.9$ for the periodic wavelet bases. These choices are based on the maximum absolute value of the wavelets found in figure 3.2 and 3.4. In these plots we simply omitted the constant 2^{j_0} to keep the upper bound closer to the actual coherence.

The final coherence estimate created in theorem 6.3 is infinite-dimensional. For future work, an interesting experiment would therefore be to apply this estimate in an infinite-dimensional setup. This could be done by choosing $\tilde{N}$ and $\tilde{M}$ in the balancing property according to this estimate, and obtain continuous Walsh-samples $\widetilde{f(k)}$ from some function f. Further one would have to apply the setup introduced section 5.4 and solve equation (5.9), to obtain a finite dimensional solution.

6.3 Summary

In this thesis we started by reviewing some of the limitations of the traditional theory within the field compressive sensing. We also verified numerically that the new principal of a multilevel random subsampling scheme outperformed the traditional principle of uniform random subsampling. In chapter 2, we studied how the ℓ_1-minimizer could be used to find a unique sparse solution in an underdetermined linear system of equations. Furthermore, we introduced the SPGL1 algorithm, which could solve the QCBP problem with matrices acting as linear operators by computing all matrix multiplications in-place.

Next, the necessary theory of wavelets and Hadamard matrices were reviewed in chapter 3 and 4. In chapter 5 we described the mathematical theory which explained why a multilevel subsampling strategy could guarantee recovery of a sparse signal. To provide this recovery guarantee, a critical requirement was to use a sampling basis which was asymptotically incoherent to the sparsifying basis. We then showed that the Hadamard sampling basis and the Haar sparsifying basis possessed this property, before reviewing the proof of the same result for a Fourier sampling basis and a Daubechies wavelet basis. Afterwards, we conducted some practical experiments verifying that the multilevel random subsampling scheme worked for these bases. We then concluded chapter 5 with a brief introduction introduction to infinite-dimensional compressive sensing.

In this final chapter we have applied our knowledge from the three previous chapters to provide a new asymptotic coherence estimate between the Hadamard sampling basis and the Daubechies wavelet basis. Additionally we have studied the coherence of some finite-dimensional matrices, and found that the new coherence estimate found in theorem 6.3 provided an upper bound on the coherence in these matrices.

As a part of this work there has also been created an open-source implementation of the Hadamard transform in C++ with bindings to MATLAB and Python. This code have proven to outperform MATLAB's own implementation, and extend Python with lacking functionality. Additionally, there have been developed an open-source implementation of the SPGL1 algorithm, making all of the result presented here reproducible without any affiliation with commercial software, such as MATLAB. Such a setup would of course require some programming experience with C++. All code used to create any figure in this text have also been made publicly available, so that any result can be understood and reproduced. This is done in the spirit of reproducible research.

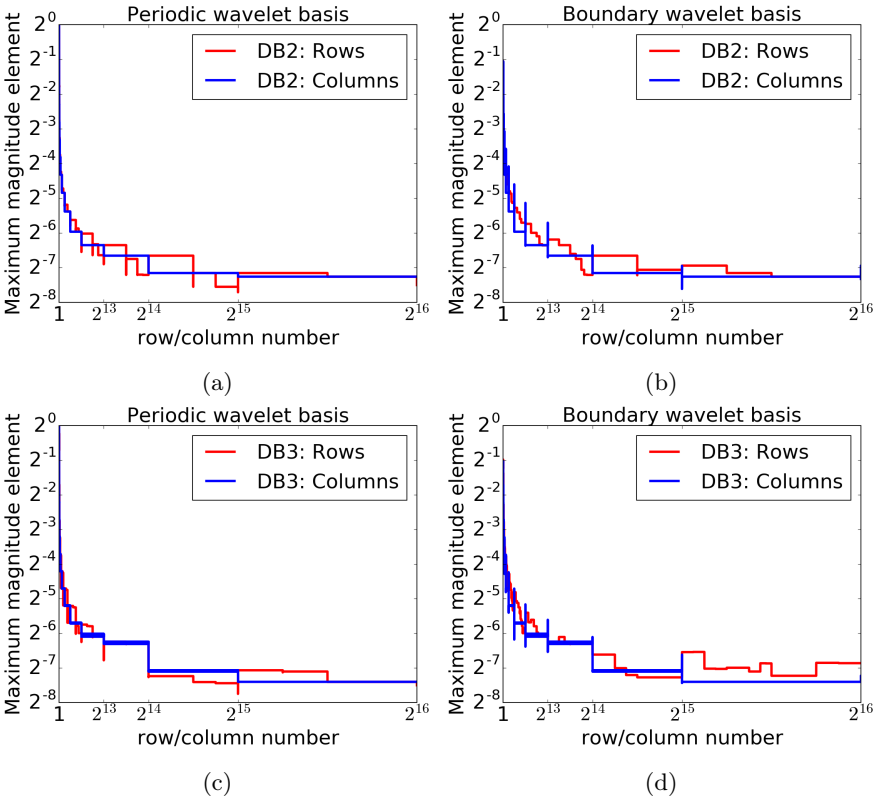


Figure 6.2: The maximum entry in each row and column found in the matrix $\mathbf{U} = \mathbf{V}_{\text{had}}\boldsymbol{\Psi}^{-1}$ for $\boldsymbol{\Psi}^{-1}$ with a periodic wavelet basis and a boundary wavelet basis.

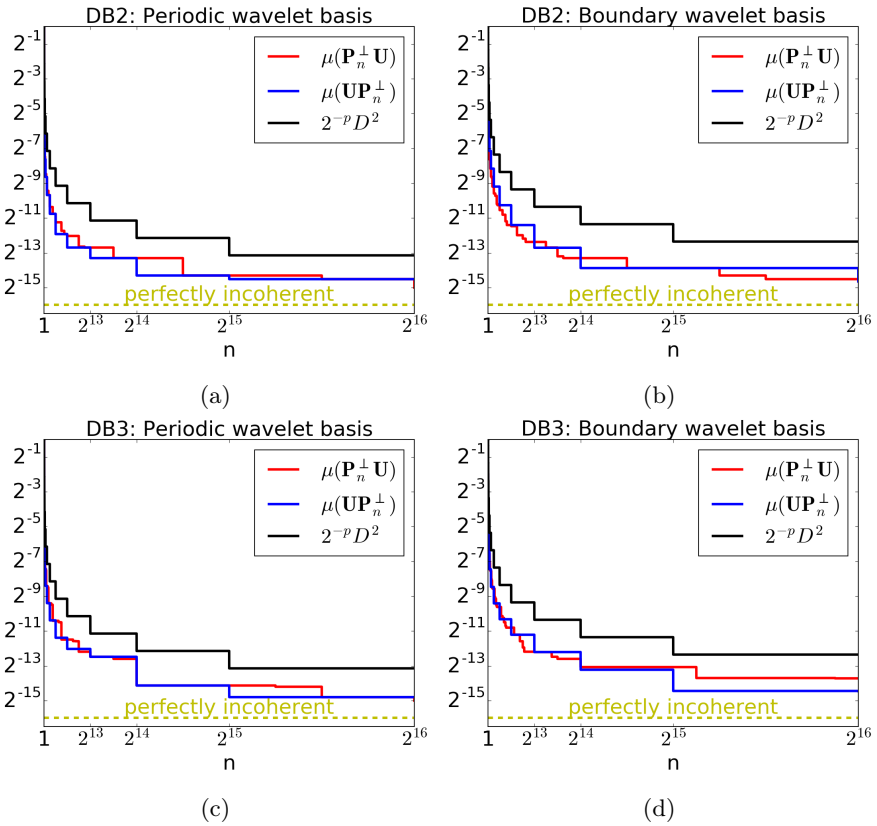


Figure 6.3: The coherence for various projections of the matrix $\mathbf{U} = \mathbf{V}_{\text{had}}\mathbf{\Psi}^{-1}$ for $\mathbf{\Psi}^{-1}$ with a periodic wavelet basis and a boundary wavelet basis. For the boundary wavelet basis $D = 2.5$ while for the periodic $D = 1.9$, the value of p is $p = \lfloor \log_2(n) \rfloor$.

Bibliography

[1] Ben Adcock and Anders C Hansen. "Generalized sampling and infinite-dimensional compressed sensing". In: *Foundations of Computational Mathematics* (2015), pp. 1–61.

[2] Ben Adcock, Anders C Hansen, Clarice Poon, and Bogdan Roman. "Breaking the coherence barrier: A new theory for compressed sensing". In: *arXiv preprint arXiv:1302.0561* (2013).

[3] Jörg Arndt. *Matters Computational: ideas, algorithms, source code*. Springer Science & Business Media, 2010.

[4] Kenneth George Beauchamp. *Walsh functions and their applications*. Vol. 3. Academic press, 1975.

[5] E. van den Berg and M. P. Friedlander. "Probing the Pareto frontier for basis pursuit solutions". In: *SIAM Journal on Scientific Computing* 31.2 (2008), pp. 890–912.

[6] E. van den Berg and M. P. Friedlander. *SPGL1: A solver for large-scale sparse reconstruction*. http://www.cs.ubc.ca/labs/scl/spgl1. 2007.

[7] Ewout Van den Berg and Michael P Friedlander. "Sparse optimization with least-squares constraints". In: *SIAM Journal on Optimization* 21.4 (2011), pp. 1201–1229.

[8] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[9] Kurt Bryan and Tanya Leise. "Making do with less: an introduction to compressed sensing". In: *SIAM Review* 55.3 (2013), pp. 547–566.

[10] Emmanuel J Candes and Yaniv Plan. "A probabilistic and RIPless theory of compressed sensing". In: *Information Theory, IEEE Transactions on* 57.11 (2011), pp. 7235–7254.

[11] Yuejie Chi, Louis L Scharf, Ali Pezeshki, and A Robert Calderbank. "Sensitivity to basis mismatch in compressed sensing". In: *Signal Processing, IEEE Transactions on* 59.5 (2011), pp. 2182–2195.

[12] Albert Cohen, Ingrid Daubechies, and Pierre Vial. "Wavelets on the interval and fast wavelet transforms". In: *Applied and computational harmonic analysis* 1.1 (1993), pp. 54–81.

[13] Germund Dahlquist and Åke Björck. *Numerical Methods in Scientific Computing*. First edition. Vol. 1. SIAM, 2008.

[14] Ingrid Daubechies. *Ten lectures on wavelets*. Vol. 61. SIAM, 1992.

[15] David Donoho and Michael Elad. "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization". In: *Proceedings of the National Academy of Sciences* 100.5 (2003), pp. 2197–2202.

[16] David Donoho, Arian Maleki, and Morteza Shahram et al. *Wavelab850*. http://statweb.stanford.edu/~wavelab/.

[17] Marco F. Duarte et al. "Single-Pixel Imaging via Compressive Sampling". In: *IEEE Signal Processing Magazine* 25.2 (2008).

[18] Yonina C Eldar and Gitta Kutyniok. *Compressed sensing: theory and applications*. Cambridge University Press, 2012.

[19] Nathan Jacob Fine. "On the Walsh functions". In: *Transactions of the American Mathematical Society* 65.3 (1949), pp. 372–414.

[20] Nathan Jacob Fine. "The generalized Walsh functions". In: *Transactions of the American Mathematical Society* 69.1 (1950), pp. 66–77.

[21] Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. First edition. Springer - Birkäuser, 2013.

[22] *GNU Image Manipulation Program*. http://www.gimp.org.

[23] B Golubov, A Efimov, and V Skvortsov. *Walsh series and transforms: theory and applications*. Vol. 64. Springer Science & Business Media, 1991.

[24] Rafael C. Gonzalez and Richard E. Woods. *Digital image processing*. Third edition. Pearson - Prentice Hall, 2008.

[25] Matthieu Guerquin-Kern, L Lejeune, Klaas P Pruessmann, and Michael Unser. "Realistic analytical phantoms for parallel magnetic resonance imaging". In: *Medical Imaging, IEEE Transactions on* 31.3 (2012), pp. 626–636.

[26] Henning F Harmuth. *Transmission of information by orthogonal functions*. Springer-Verlag, 1970.

[27] Eugenio Hernández and Guido Weiss. *A first course on wavelets*. CRC press, 1996.

[28] Brian W Kernighan, Dennis M Ritchie, and Per Ejeklint. *The C programming language*. Vol. 2. prentice-Hall Englewood Cliffs, 1988.

[29] Stéphane Mallat. *A wavelet tour of signal processing: the sparse way*. Third edition. Academic Press, 2008.

[30] John N. Mcdonald and Neil A. Weiss. *A Course in Real Analysis*. Second edition. Academic Press, 2013.

[31] Bogdan Roman, Anders Hansen, and Ben Adcock. "On asymptotic structure in compressed sensing". In: *arXiv preprint arXiv:1406.4178* (2014).

[32] Rudi Rottenfusser, Erin E. Wilson, and Michael W. Davidson. *The Point Spread Function*. Zeizz. URL: http://zeiss-campus.magnet.fsu.edu/articles/basics/psf.html.

[33] Gilbert Strang and Truong Nguyen. *Wavelets and filter banks*. Wellesley-Cambridge Press, 1996.

[34] Vincent Studer et al. "Compressive fluorescence microscopy for biological and hyperspectral imaging". In: *Proceedings of the National Academy of Sciences* 109.26 (2012), E1679–E1687.

[35] D. Takhar et al. "A New Compressive Imaging Camera Architecture using Optical-Domain Compression". In: *Proc. Computational Imaging IV at SPIE Electronic Imaging*. 2006.

[36] David S Taubman and Michael W Marcellin. "JPEG2000: Standard for interactive imaging". In: *Proceedings of the IEEE* 90.8 (2002), pp. 1336–1357.

Index