

Mathematical Foundations of Data Sciences



Gabriel Peyré
CNRS & DMA
École Normale Supérieure
gabriel.peyre@ens.fr
<https://mathematical-tours.github.io>
www.numerical-tours.com

September 11, 2018

Presentation

This book draft presents an overview of important mathematical and numerical foundations for modern data sciences. It covers in particulars the basics of signal and image processing (Fourier, Wavelets, and their applications to denoising and compression), imaging sciences (inverse problems, sparsity, compressed sensing) and machine learning (linear regression, logistic classification, deep learning). The focus is on the mathematically-sounded exposition of the methodological tools (in particular linear operators, non-linear approximation, convex optimization, optimal transport) and how they can be mapped to efficient computational algorithms. These course notes are also intended to be the theoretical companion for the Numerical Tours¹ web site, which presents Matlab/Python/Julia/R detailed implementations of all the concepts covered here.

¹www.numerical-tours.com

Contents

Chapter 1

Shannon Theory

Shannon theory of information, published in 1948/1949, is made of three parts:

1. Sampling: it studies condition under which sampling a continuous function to obtain a discrete vector is invertible. The discrete real values representing the signal are then typically quantized to a finite precision to obtain a set of symbols in a finite alphabet.
2. Source coding: it studies optimal ways to represent (code) such a set of symbols as a binary sequence. It leverages the statistical distributions to obtain the most possible compact code.
3. Channel coding (not studied here): it studies how to add some redundancy to the coded sequence in order to gain robustness to errors or attacks during transmission (flip of certain bits with some probability). It is often named “error correcting codes theory”.

The main reference for this chapter is [?].

1.1 Analog vs. Discrete Signals

To develop numerical tools and analyze their performances, the mathematical modelling is usually done over a continuous setting (so-called “analog signals”). Such continuous setting also aims at representing the signal in the physical world, which are inputs to sensors hardwares such as microphone, digital cameras or medical imaging devices. An analog signal is a 1-D function $f_0 \in L^2([0, 1])$ where $[0, 1]$ denotes the domain of acquisition, which might for instance be time. An analog image is a 2D function $f_0 \in L^2([0, 1]^2)$ where the unit square $[0, 1]^2$ is the image domain.

Although these notes are focussed on the processing of sounds and natural images, most of the methods extend to multi-dimensional datasets, which are higher dimensional mappings

$$f_0 : [0, 1]^d \rightarrow [0, 1]^s$$

where d is the dimensionality of the input space ($d = 1$ for sound and $d = 2$ for images) whereas s is the dimensionality of the feature space. For instance, gray scale images corresponds to ($d = 2, s = 1$), videos to ($d = 3, s = 1$), color images to ($d = 2, s = 3$) where one has three channels (R, G, B). One can even consider multi-spectral images where ($d = 2, s \gg 3$) that is made of a large number of channels for different light wavelengths. Figures ?? and ?? show examples of such data.

1.1.1 Acquisition and Sampling

Signal acquisition is a low dimensional projection of the continuous signal performed by some hardware device. This is for instance the case for a microphone that acquires 1D samples or a digital camera that

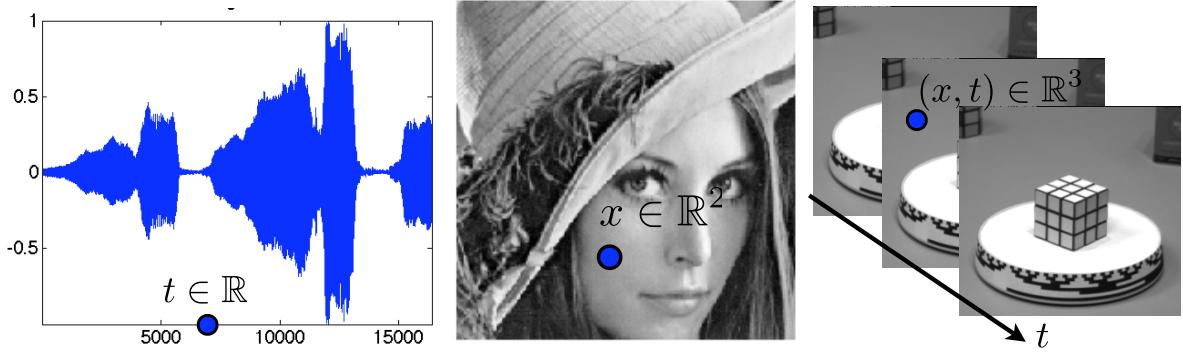


Figure 1.1: Examples of sounds ($d = 1$), image ($d = 2$) and videos ($d = 3$).

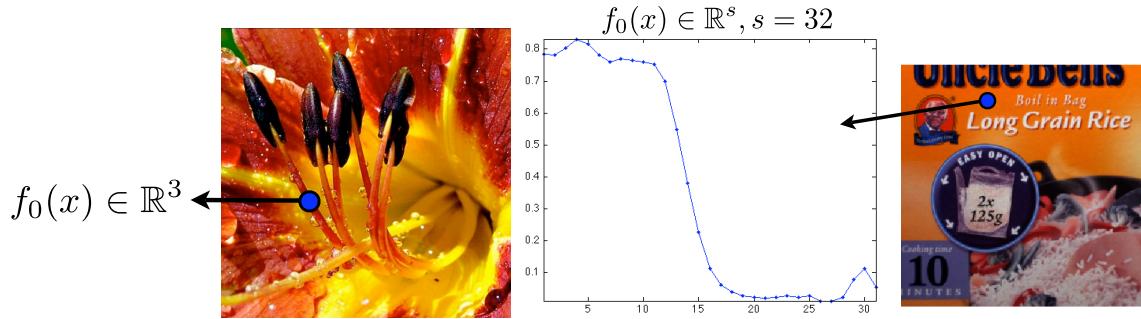


Figure 1.2: Example of color image $s = 3$ and multispectral image ($s = 32$).

acquires 2D pixel samples. The sampling operation thus corresponds to mapping from the set of continuous functions to a discrete finite dimensional vector with N entries.

$$f_0 \in L^2([0, 1]^d) \mapsto f \in \mathbb{C}^N$$

Figure ?? shows examples of discretized signals.

1.1.2 Linear Translation Invariant Sampler

A translation invariant sampler performs the acquisition as an inner product between the continuous signal and a constant impulse response h translated at the sample location

$$f_n = \int_{-S/2}^{S/2} f_0(x) h(n/N - x) dx = f_0 \star h(n/N). \quad (1.1)$$

The precise shape of $h(x)$ depends on the sampling device, and is usually a smooth low pass function that is maximal around $x = 0$. The size S of the sampler determines the precision of the sampling device, and is usually of the order of $1/N$ to avoid blurring (if S is too large) or aliasing (if S is too small).

Section ?? details how to reverse the sampling operation in the case where the function is smooth.

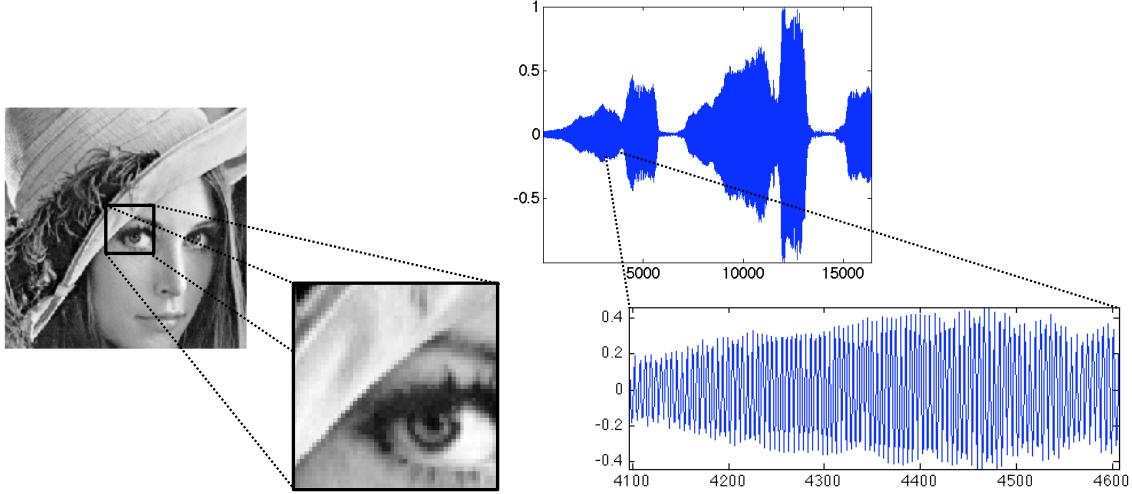


Figure 1.3: Image and sound discretization.

1.2 Shannon Sampling Theorem

Reminders about Fourier transform. For $f \in L^1(\mathbb{R})$, its Fourier transform is defined as

$$\forall \omega \in \mathbb{R}, \quad \hat{f}(\omega) \stackrel{\text{def.}}{=} \int_{\mathbb{R}} f(x) e^{-ix\omega} dx. \quad (1.2)$$

One has $\|\hat{f}\|^2 = (2\pi)^{-1} \|f\|^2$, so that $f \mapsto \hat{f}$ can be extended by continuity to $L^2(\mathbb{R})$, which corresponds to computing \hat{f} as a limit when $T \rightarrow +\infty$ of $\int_{-T}^T f(x) e^{-ix\omega} dx$. When $\hat{f} \in L^1(\mathbb{R})$, one can invert the Fourier transform so that

$$f(x) = \frac{1}{2\pi} \int_{\mathbb{R}} \hat{f}(\omega) e^{ix\omega} d\omega, \quad (1.3)$$

which shows in particular that f is continuous with vanishing limits at $\pm\infty$.

The Fourier transform $\mathcal{F} : f \mapsto \hat{f}$ exchanges regularity and decay. For instance, if $f \in C^p(\mathbb{R})$ with an integrable Fourier transform, then $\mathcal{F}(f^{(p)})(\omega) = (i\omega)^{-p} \hat{f}(\omega)$ so that $|\hat{f}(\omega)| = O(1/|\omega|^p)$. Conversely,

$$\int_{\mathbb{R}} (1 + |\omega|)^p |\hat{f}(\omega)| d\omega < +\infty \implies f \in C^p(\mathbb{R}). \quad (1.4)$$

For instance, if $\hat{f}(\omega) = O(1/|\omega|^{p+2})$, one obtains that $f \in C^p(\mathbb{R})$.

Reminders about Fourier series. We denote $\mathbb{T} = \mathbb{R}/2\pi\mathbb{Z}$ the torus. A function $f \in L^2(\mathbb{T})$ is 2π -periodic, and can be viewed as a function $f \in L^2([0, 2\pi])$ (beware that this means that the boundary points are glued together), and its Fourier coefficients are

$$\forall n \in \mathbb{Z}, \quad \hat{f}_n \stackrel{\text{def.}}{=} \frac{1}{2\pi} \int_0^{2\pi} f(x) e^{-ixn} dx.$$

This formula is equivalent to the computation of an inner-product $\hat{f}_n = \langle f, e_n \rangle$ for the inner-product $\langle f, g \rangle \stackrel{\text{def.}}{=} \frac{1}{2\pi} \int_{\mathbb{T}} f(x) \bar{g}(x) dx$. For this inner product, $(e_n)_n$ is orthonormal and is actually an Hilbert basis, meaning that one reconstructs with the following converging series

$$f = \sum_{n \in \mathbb{Z}} \langle f, e_n \rangle e_n \quad (1.5)$$

which means $\|f - \sum_{n=-N}^N \langle f, e_n \rangle e_n\|_{L^2(\mathbb{T})} \rightarrow 0$ for $N \rightarrow +\infty$. The pointwise convergence of (??) at some $x \in \mathbb{T}$ is ensured if for instance f is differentiable. The series is normally convergent (and hence uniform) if for instance f is of class C^2 on \mathbb{T} since in this case, $\hat{f}_n = O(1/n^2)$. If there is a step discontinuities, then there are Gibbs oscillations preventing uniform convergence, but the series still converges to the half of the left and right limit.

Poisson formula. The Poisson formula connects the Fourier transform and the Fourier series to sampling and periodization operators. For some function $h(t)$ defined on \mathbb{R} (typically the goal is to apply this to $h = \hat{f}$), its periodization reads

$$h_P(t) \stackrel{\text{def.}}{=} \sum_n h(t - 2\pi n). \quad (1.6)$$

This formula makes sense if $h \in L^1(\mathbb{R})$, and in this case $\|h_P\|_{L^1(\mathbb{T})} \leq \|h\|_{L^1(\mathbb{R})}$ (and there is equality for positive functions). The Poisson formula, stated in Proposition ?? below, corresponds to proving that the following diagram

$$\begin{array}{ccc} f(x) & \xrightarrow{\mathcal{F}} & \hat{f}(\omega) \\ \downarrow & & \downarrow \\ (\text{sampling}) & \xrightarrow{\text{Fourier serie}} & \sum_n f(n)e^{-i\omega n} \\ (f(n))_n & & \end{array} \quad \begin{array}{c} \text{periodization} \end{array}$$

is actually commutative.

Proposition 1 (Poisson formula). *Assume that \hat{f} has compact support and that $|f(x)| \leq C(1+|x|)^{-3}$ for some C . Then one has*

$$\forall \omega \in \mathbb{R}, \quad \sum_n f(n)e^{-i\omega n} = \hat{f}_P(\omega). \quad (1.7)$$

Proof. Since \hat{f} is compactly supported, \hat{f}_P is well defined (it involves only a finite sum) and since f has fast decay, using (??), $(\hat{f})_P$ is C^1 . It is thus the sum of its Fourier series

$$(\hat{f})_P(\omega) = \sum_k c_k e^{ik\omega}, \quad (1.8)$$

where

$$c_k = \frac{1}{2\pi} \int_0^{2\pi} (\hat{f})_P(\omega) e^{-ik\omega} d\omega = \frac{1}{2\pi} \int_0^{2\pi} \sum_n \hat{f}(\omega - 2\pi n) e^{-ik\omega} d\omega.$$

One has

$$\int_0^{2\pi} \sum_n |\hat{f}(\omega - 2\pi n) e^{-ik\omega}| d\omega = \int_{\mathbb{R}} |\hat{f}|$$

which is bounded because $\hat{f} \in L^1(\mathbb{R})$ (it has a compact support and is C^1), so one can exchange the sum and integral

$$c_k = \sum_n \frac{1}{2\pi} \int_0^{2\pi} \hat{f}(\omega - 2\pi n) e^{-ik\omega} d\omega = \frac{1}{2\pi} \int_{\mathbb{R}} \hat{f}(\omega) e^{-ik\omega} d\omega = f(-k)$$

where we used the inverse Fourier transform formula (??), which is legit because $\hat{f} \in L^1(\mathbb{R})$. \square

Shannon theorem. Shannon sampling theorem state a sufficient condition ensuring that the sampling operator $f \mapsto (f(ns))_n$ is invertible for some sampling step size $s > 0$. It requires that $\text{supp}(\hat{f}) \subset [-\pi/s, \pi/s]$, which, thanks to formula (??), implies that \hat{f} is C^∞ (in fact it is even analytic). This theorem was first proved by Whittaker in 1915. It was re-proved and put in perspective in electrical engineering by Nyquist in 1928. It became famous after the paper of Shannon in 1949, which put forward its importance in numerical communications. Figure ?? give some insight on how the proof works (left) and more importantly, on what happens when the compact support hypothesis fails (in which case aliasing occurs, see also Figure ??).

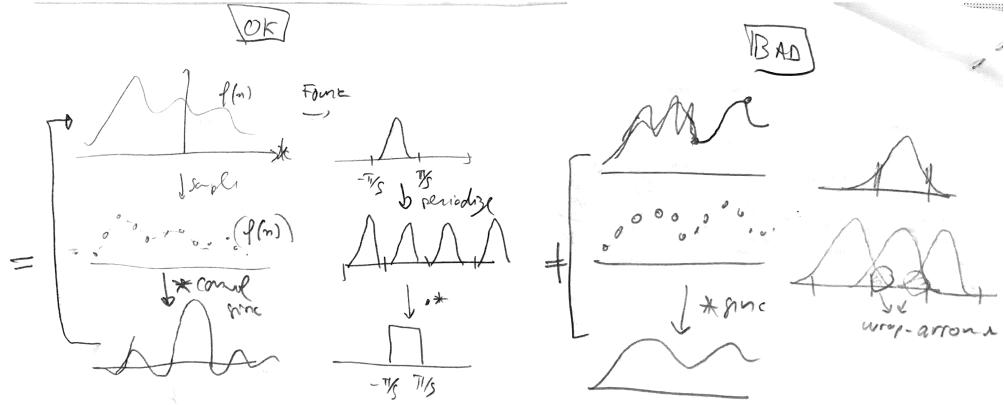


Figure 1.4: Schematic view for the proof of Theorem ??.

Theorem 1. If $|f(x)| \leq C(1 + |x|)^{-3}$ for some C and $\text{supp}(\hat{f}) \subset [-\pi/s, \pi/s]$, then one has

$$\forall x \in \mathbb{R}, \quad f(x) = \sum_n f(n) \text{sinc}(x/s - n) \quad \text{where} \quad \text{sinc}(u) = \frac{\sin(\pi u)}{\pi u} \quad (1.9)$$

with uniform convergence.

Proof. The change of variable $g \stackrel{\text{def.}}{=} f(s \cdot)$ results in $\hat{g} = 1/s \hat{f}(\cdot/s)$ so that we can restrict our attention to $s = 1$. The compact support hypothesis implies $\hat{f}(\omega) = 1_{[-\pi, \pi]}(\omega) \hat{f}_P(\omega)$. Combining the inversion formula (??) with Poisson formula (??)

$$f(x) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \hat{f}_P(\omega) e^{i\omega x} d\omega = \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_n f(n) e^{i\omega(x-n)} d\omega.$$

Since f has fast decay, $\int_{-\pi}^{\pi} |\sum_n f(n) e^{i\omega(x-n)}| d\omega = \sum_n |f(n)| < +\infty$, so that one can exchange summation and integration and obtain

$$f(x) = \sum_n f(n) \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{i\omega(x-n)} d\omega = \sum_n f(n) \text{sinc}(x - n).$$

□

One issue with this reconstruction formula is that it uses a slowly decaying and very oscillating sinc kernels. In practice, one rarely uses such a kernel for interpolation, and one prefers smoother and more localized kernel. If $\text{supp}(\hat{f}) \subset [-\pi/s', \pi/s']$ with $s' > s$ (i.e. have a more compact spectrum), one can re-do the proof of the theorem, and one gains some degree of freedom to design the reconstruction kernel, which now can be chosen smoother in Fourier and hence have exponential decay in time.

Spline interpolation are defined by considering $\varphi_0 = 1_{[-1/2, 1/2]}$ and $\varphi_k = \varphi_{k-1} * \varphi_0$ which is a piecewise polynomial of degree k and has bounded derivative of order k (and is of class C^{k-1}) with compact support on $[-(k+1)/2, (k+1)/2]$.

The reconstruction formula reads $f \approx \tilde{f} \stackrel{\text{def.}}{=} \sum_n a_n \varphi(\cdot - n)$ where $(a_n)_n$ is computed from the $(f(n))_n$ by solving a linear system (associated to the interpolation property $\tilde{f}(n) = f(n)$). It is only in the cases $k \in \{0, 1\}$ (piecewise constant and affine interpolations) that one has $a_n = f(n)$. In practice, one typically use the cubic spline interpolation, which corresponds to $k = 3$.

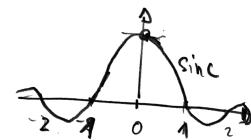


Figure 1.5: sinc kernel

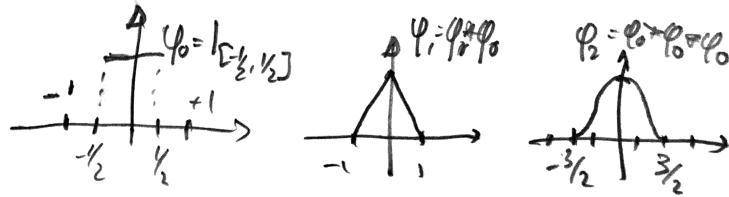


Figure 1.6: Cardinal splines as basis functions for interpolation.

Associated code: `test_sampling.m`

This theorem also explains what happens if \hat{f} is not supported in $[-\pi/s, \pi/s]$. This leads to aliasing, and high frequency outside this interval leads to low frequency artifacts often referred to as “aliasing”. If the input signal is not bandlimited, it is thus very important to pre-filter it (smooth it) before sampling to avoid this phenomena (of course this kills the high frequencies, which are lost), see Figure ??.

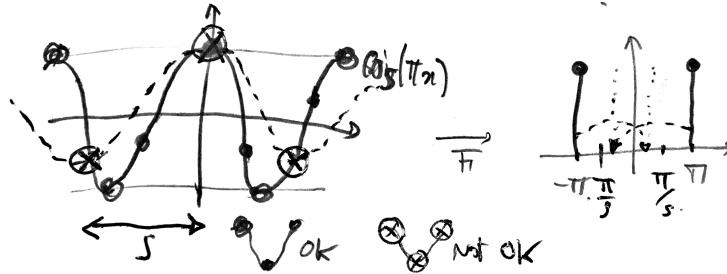


Figure 1.7: Aliasing in the simple case of a sine wave (beware however that this function does not have compact support).

Quantization. Once the signal has been sampled to obtain a discrete vector, in order to store it and transmit it, it is necessary to quantize the value to some finite precision. Section ?? presents transform coding, which is an efficient family of compression schemes which operate the quantization over some transformed domain (which correspond to applying a linear transform, usually orthogonal, to the sampled values). This is useful to enhance the performance of the source coding scheme. It is however common to operate directly the quantization over the sampled value.

Considering for instance a step size $s = 1/N$, one samples $(u_n \stackrel{\text{def.}}{=} f(n/N))_{n=1}^N \in \mathbb{R}^N$ to obtain a finite dimensional data vector of length N . Note that dealing with finite data corresponds to restricting the function f to some compact domain (here $[0, 1]$) and is contradictory with Shannon sampling theorem, since a function f cannot have a compact support in both space and frequency (so perfect reconstruction never holds when using finite storage).

Choosing a quantization step T , quantization $v_n = Q_T(u_n) \in \mathbb{Z}$ rounds to the nearest multiple of T , i.e.

$$\tilde{u} = Q_T(u) \Leftrightarrow v - \frac{1}{2} \leq u/T \leq v + \frac{1}{2}$$

[ToDo: put here a figure]. De-quantization is needed to restore a signal, and the best reconstruction (in average or in worse case) is defined by setting $D_T(v) \stackrel{\text{def.}}{=} Tv$. Quantizing and then de-quantizing introduce an error bounded by $T/2$, since $|D_T(Q_T(u)) - u| \leq T/2$. Up to machine precision, quantization is the only source of error (often called “lossy compression”) in Shannon’s standard pipeline.

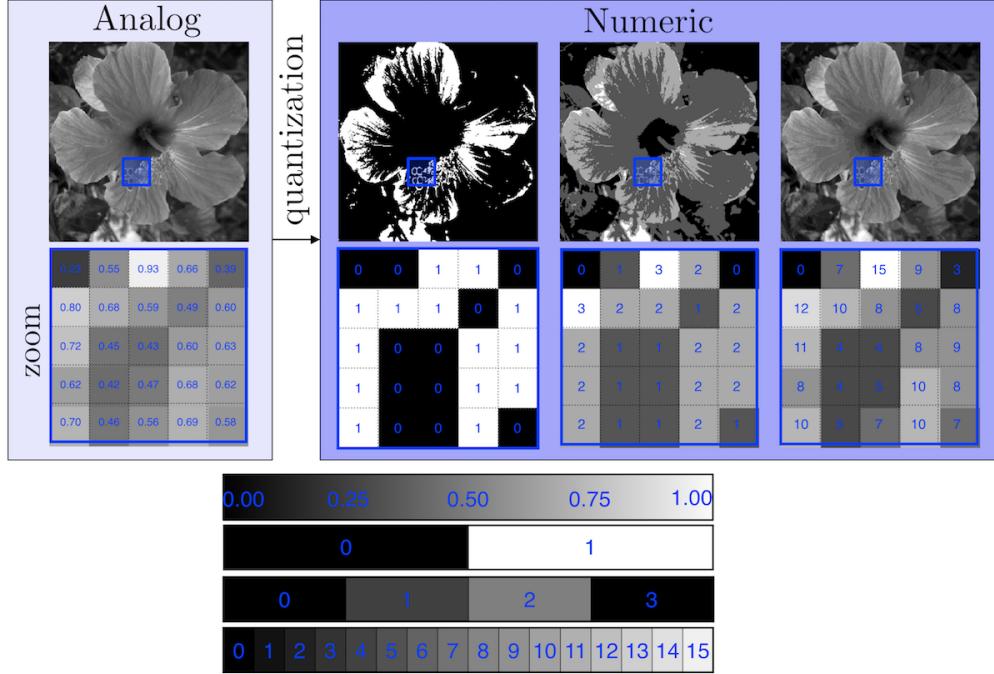


Figure 1.8: Quantizing an image using a decaying $T = 1/K$ where $K \in \{2, 3, 4, 16\}$ is the number of graylevels and the original image is normalized so that $0 \leq f_0 < 1$.

1.3 Shannon Source Coding Theorem

Uniform coding. We consider an alphabet (s_1, \dots, s_K) of K symbols. For instance, if one samples and quantize a bounded signal $0 \leq f_0 < 1$ using a step size $1/K$, then one can consider $s_k = k$ to be integer symbols. For text, these symbols include the letter plus extra punctuation symbols and blank. It is of course possible to code a sequence of such symbols using a uniform code (e.g. using the base 2 expansion) with $\lceil \log_2(K) \rceil$ bit per symbols. For instance if $K = 4$ and the symbols are $\{0, 1, 2, 3\}$, then the code words are $(c_0 = 00, c_1 = 01, c_2 = 10, c_3 = 11)$.

This uniform coding strategy is however extremely inefficient if the symbols are not uniformly distributed (i.e. if some symbols are more frequent than other, which is likely to be the case). We aim at designing better codes.

Prefix coding. A code $c_k = c(s_k)$ associate to each symbol s_k a code word $c_k \in \{0, 1\}^{\mathbb{N}}$ with a varying length $|c_k| \in \mathbb{N}^*$. A prefix code $c_k = c(s_k)$ is such that no word c_k is the beginning of another word c'_k . This is equivalent to be able to embed the $(c_k)_k$ as leaves of a binary tree T , with the code being output of a traversal from root to leaves (with a convention that going to a left (resp. right) child output a 0 (resp. a 1). We denote $c = \text{Leaves}(T)$ such prefix property.

This tree-based representation is useful to decode a binary stream by simply performing tree traversal. One follows the tree, from top to bottom, and outputs a symbol each time a leaf is reached (and then re-start at the top).

The following fundamental lemma describes the set of prefix codes using an inequality.

Lemma 1 (Kraft inequality). (i) For a code c , if there exists a tree T such that $c = \text{Leaves}(T)$ then

$$\sum_k 2^{-|c_k|} \leq 1. \quad (1.10)$$

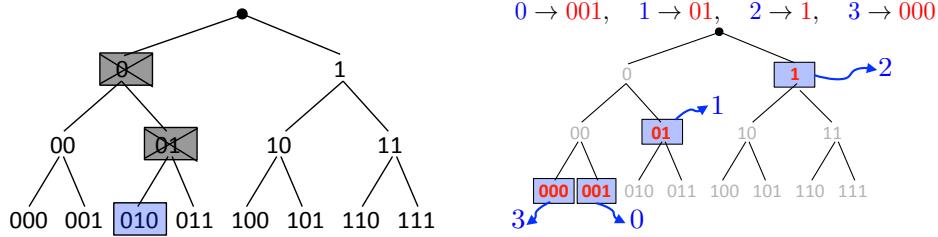


Figure 1.9: Left: complete tree of all codes of length 3; right: example of prefix code.

(ii) Conversely, if $(\ell_k)_k$ are such that

$$\sum_k 2^{-\ell_k} \leq 1 \quad (1.11)$$

then there exists a code $c = \text{Leaves}(T)$ such that $|c_k| = \ell_k$.

Proof. \Rightarrow We suppose $c = \text{Leaves}(T)$. We denote $m = \max_k |c_k|$ and consider the full binary tree. Below each c_k , one has a sub-tree of height $m - |c_k|$, see Figure ??, left. This sub-tree has $2^{m-|c_k|}$ leaves. Since all these sub-trees do not overlap, the total number of leaf do not exceed the total number of leaves 2^m of the full binary tree, hence

$$\sum_k 2^{m-|c_k|} \leq 2^m,$$

hence (??).

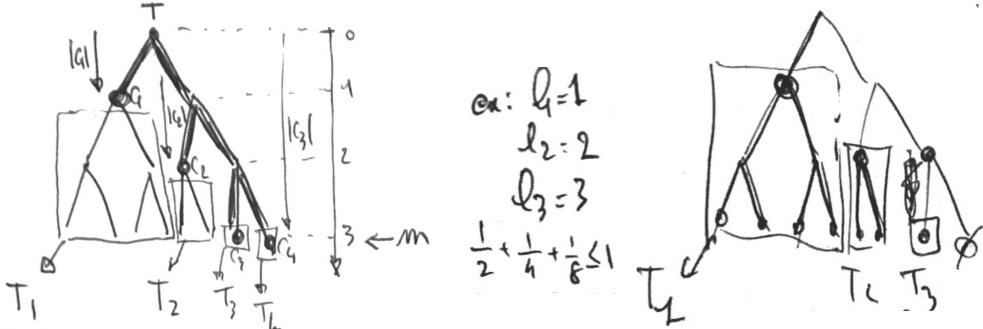


Figure 1.10: Left: full binary tree obtained by completing the tree associated to the code $(c_1 = 0, c_2 = 10, c_3 = 110, c_4 = 111)$. Right: packing sub-trees associated to code length to form the left part of the full tree.

\Leftarrow Conversely, we assume (??) holds. Without loss of generality, we assume that $|c_1| \geq \dots \geq |c_K|$. We start by putting a sub-tree of height $2^{m-|c_1|}$. Since the second tree is smaller, one can put it immediately aside, and continue this way, see Figure ??, right. Since $\sum_k 2^{m-|c_k|} \leq 2^m$, this ensure that we can stack side-by-side all these sub-tree, and this defines a proper sub-tree of the full binary tree. \square

Probabilistic modeling. We aim at designing the most possible compact code c_k . We assume at our disposal some probability distribution over this alphabet, which is just an histogram $p = (p_1, \dots, p_K) \in \mathbb{R}_+^K$ in the simplex, i.e. $\sum_k p_k = 1$. In practice, this probability is usually the empirical probability of appearance of the symbols x_k in the data to be coded.

The entropy of such an histogram is

$$H(p) \stackrel{\text{def.}}{=} - \sum_k p_k \log_2(p_k)$$

with the convention $0 \log_2(0) = 0$.

Denoting $h(u) = -u \log(u)$, $h''(u) = -1/u < 0$ so that H is strictly concave. The definition of the entropy extends to continuous density $f(x)$ for x on some measure space with reference measure dx (e.g. Lebesgue on \mathbb{R}^d) by setting $H(f) = \int f(x) \log(f(x))dx$.

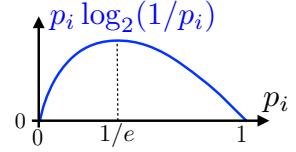


Figure 1.11:

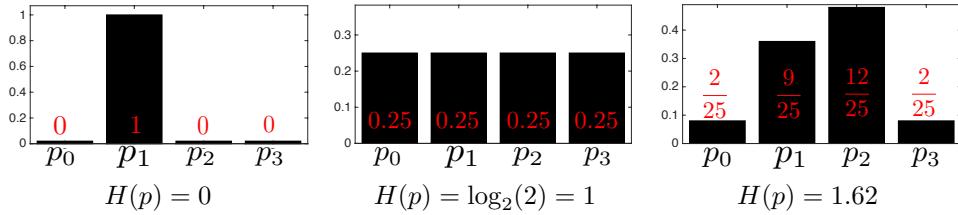


Figure 1.12: Three examples of probability distributions with corresponding entropies.

Lemma 2. One has

$$0 \leq H(p) \leq \log_2(K).$$

Proof. We consider the following constrained optimization problem

$$\min_p \left\{ f(p) ; g(p) = \sum_k p_k = 1 \right\}$$

where $f = -H$. According to the linked extrema theorem, at an optimum p^* , $\nabla f(p^*) = \lambda \nabla g(p^*)$ for some $\lambda \in \mathbb{R}$, so that here $\log(p_k^*) + 1 = \lambda$, i.e. $p_k^* = c$ is constant, and since $\sum_k p_k^* = 1$, one has $p_k^* = 1/K$ and thus $H(p) = \log_2(K)$. \square

Shannon theorem. Assuming that $(p_k)_k$ is the empirical probability of appearance of the symbols x_k in the data to be coded, the average symbol length associated to some code c is

$$L(c) \stackrel{\text{def.}}{=} \sum_k p_k |c_k|.$$

The goal is to design the best possible c so that $L(c)$ is as small as possible. Shannon theorem of entropic coding, proved below, give both lower and upper bound for this question.

Theorem 2. (i) If $c = \text{Leaves}(T)$ for some tree T , then

$$L(c) \geq H(p).$$

(ii) Conversely, there exists a code c with $c = \text{Leaves}(T)$ such that

$$L(c) \leq H(p) + 1.$$

Proof. First, we consider the following optimization problem

$$\min_{\ell=(\ell_k)_k} \left\{ f(\ell) \stackrel{\text{def.}}{=} \sum_k \ell_k p_k ; g(\ell) \stackrel{\text{def.}}{=} \sum_k 2^{-\ell_k} \leq 1 \right\}. \quad (1.12)$$

We first show that at an optimal ℓ^* , the constraint is saturated, i.e. $g(\ell^*) = 1$. Indeed, if $g(\ell^*) = 2^{-u} < 1$, with $u > 0$, we define $\ell'_k \stackrel{\text{def.}}{=} \ell_k^* - u$, which satisfies $g(\ell') = 1$ and also $f(\ell') = \sum_k (\ell_k - u)p_k < f(\ell^*)$, which is a contradiction. So we can restrict in (??) the constraint to $g(\ell) = 1$ and apply the linked extra theorem, which shows that necessarily, there exists $\lambda \in \mathbb{R}$ with $\nabla f(\ell^*) = \nabla g(\ell^*)$, i.e. $(p_k)_k = -\lambda \ln(2)(2^{-\ell_k^*})_k$. Since $\sum_k p_k = \sum_k 2^{-\ell_k^*} = 1$, we deduce that $\ell_k^* = -\log(p_k)$.

(i) If $c = \text{Leave}(T)$, the by Kraft inequality (??), necessarily $\ell_k = |c_k|$ satisfy the constraints of (??), and thus $H(p) = f(\ell^*) \leq f(\ell) = L(\ell)$.

(ii) We define $\ell_k \stackrel{\text{def.}}{=} \lceil -\log_2(p_k) \rceil \in \mathbb{N}^*$. Then $\sum_k 2^{-\ell_k} \leq \sum_k 2^{\log_2(p_k)} = 1$, so that these lengths satisfy (??). Thanks to Proposition ?? (ii), there thus exists a prefix code c with $|c_k| = \lceil -\log_2(p_k) \rceil$. Furthermore

$$L(c) = \sum_k p_k \lceil -\log_2(p_k) \rceil \leq \sum_k p_k (-\log_2(p_k) + 1) = H(p) + 1.$$

□

Note that this proof is constructing, i.e. it gives an algorithm that construct an almost optimal c , and this code is often called the Shannon-Fano code. It is usually a good code, although it is not necessarily the optimal code with the smallest $L(c)$. Such an optimal code can easily be computed in almost linear time (only sorting of the probability is needed, so it is $K(\log(K))$) by Huffman's dynamic programming algorithm (invented in 1952). The proof of correctness of this algorithm is however a bit tedious. Figure ?? shows an example of application of this algorithm.

In practice, such an entropic coding, although optimal, is not very efficient when one of the symbol has a large probability p_k . This is because then $2^{-p_k} \ll 1$ but one cannot allocate a fractional number of bit. This is why $L(c)$ can be as large as $H(p) + 1$. A simple workaround is to artificially increase the size of the alphabet from K to K^r by grouping together sets of r consecutive symbols, and thus reducing the gap to $H(p) + 1/r$. Constructing the code and coding however becomes very slow for large r . The standard way to achieve this without explicitly doing the grouping is by using arithmetic coding, invented in 1976, which uses interval arithmetic to allocate fractional number of bits and leveraging the fact that one usually code large sequence, thus approaching to arbitrary precision Shannon bound $H(p)$ as the length of the data increases.

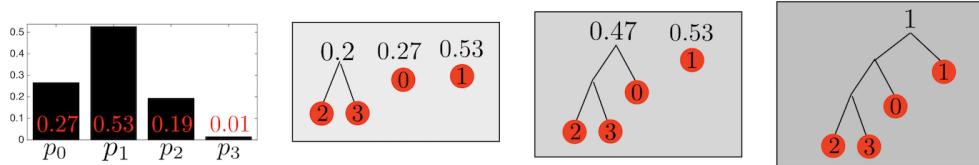


Figure 1.13: Huffman coding algorithm in action.

Note that while we give some statistical and probabilistic interpretation of entropy (measure of uncertainty) and of Shannon theorem, this theory is fully deterministic and give a bound for the actual length $NL(c)$ of coding some sequence of length N if the probability p are the empirical probability of the sequence.

If one choose a different probability q and use it to code the sequence, one necessarily obtain a worse average coding length, and this is reflected by the positivity of the so-called relative entropy (beware that it is a convex function while the entropy is concave), which is often called the Kulback-Leibler divergence

$$\text{KL}(p|q) = - \sum_k p_k \log q_k - H(p) = \sum_k p_k \log \frac{p_k}{q_k} \geq 0.$$

This KL divergence is similar to a distance in the sense that $\text{KL}(p|q) = 0$ if and only if $p = q$ (note however that KL is not symmetric and does not satisfies the triangular inequality). It also has the remarkable property that it is jointly convex in (p, q) . It is of paramount importance to compare probability distributions and measures, and form the basis of the fields of information theory and information geometry.

Doing better. One can wonder if it is possible to go below the entropy bound. By the virtue of Shannon theorem, it is not possible if only can only code in sequential order the symbols themselves. From a statistical perspective, it is as if the symbols were considered to be independent. If there is some redundancy in the sequence of symbol (for instance if they are discretization of a smooth function, so that consecutive symbols are likely to be equal), it is possible to re-transform (in a bijective way) the sequence to make them “more independent”. A simple illustration of this idea is given in Figure ??, where one computes successive difference of a 1D sequence of symbols (beware to also retain the initial symbol to be able to do the decoding).

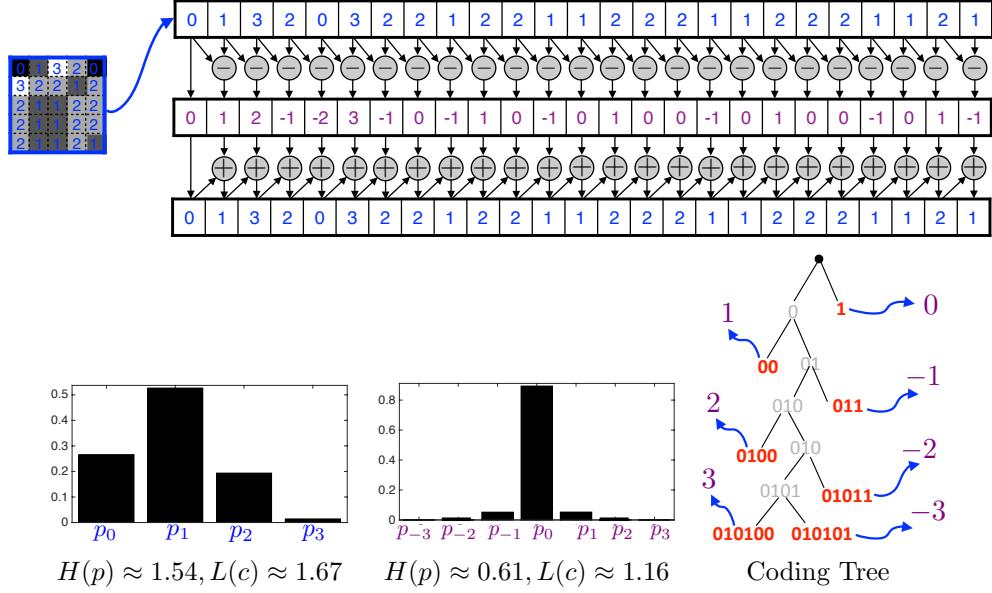


Figure 1.14: Top: retransformation by successive differences. Bottom: comparison of histograms of pixel values and differences, and a code tree for these differences.

Another, more systematic way to leverage such temporal redundancy is by performing run-length-coding, which operate by grouping together sequence of similar symbols and thus coding first a symbol and then the length of the associated group (which is coded entropically). If the sequence is generated by a Markov chain, this method can be shown to asymptotically reach the Shannon bound where now the entropy is the entropy associated to the distribution of the Markov chain on infinite sequences (which can be computed as the limit of the entropy for finite sequences).

Chapter 2

Fourier Transforms

The main references for this chapter is [?]. The Fourier transforms offers a perfect blend of analysis (solution of PDEs, approximation of functions), algebra (characters of groups, representation theory) and computer science (the FFT). It is the basics of signal processing because it allows to compute efficiently and study theoretically convolution operator, which are the shift-invariant operators. This chapter offers a glimpse of all these different facets.

2.1 Hilbert spaces and Fourier Transforms

2.1.1 Hilbertian bases.

A large class of method in data sciences (including for instance most signal processing tasks, but also data pre-processing for machine learning) operate by first projecting the input data on some basis. A particularly simple instance of this is when the basis is orthogonal, since in this case one has a simple reconstruction formula and conservation of energy, which is crucial to do a theoretical analysis. We explain this in a general Hilbert space, which can be for instance $\mathcal{H} = L^2([0, 1]^d)$ when dealing with continuous signal, or $\mathcal{H} = \mathbb{R}^N$ for discrete signal.

An (complex) Hilbert space $(\mathcal{H}, \langle \cdot, \cdot \rangle)$ is complete, where $\langle \cdot, \cdot \rangle$ is an hermitian inner product (i.e. $\langle f, g \rangle$ is the conjugate of $\langle g, f \rangle$). If it is separable, it can be equipped with an Hilbertian orthogonal basis $(\varphi_n)_{n \in \mathbb{N}}$, which means that one can expand any $f \in \mathcal{H}$ as

$$f = \sum_n \langle f, \varphi_n \rangle \varphi_n$$

where the convergence is in the sense of $\|f\|^2 \stackrel{\text{def.}}{=} \langle f, f \rangle$, i.e. $\|f - \sum_{n=0}^N \langle f, \varphi_n \rangle \varphi_n\| \rightarrow 0$ as $N \rightarrow +\infty$. One also have the conservation of energy

$$\|f\|^2 = \sum_n \langle f, \varphi_n \rangle^2.$$

A way to construct such an ortho-basis is using Gram-Schmidt orthogonalization procedure. On $L^2([0, 1])$ equipped with the usual inner product, orthogonalization of monomials defines the Legendre polynomials. On $L^2(\mathbb{R})$ equipped with a Gaussian measure, this leads to functions of the form $P_n(x)e^{-x^2}$ where P_n are Laguerre polynomials. Intuitively, orthogonality forces φ_n to have n “oscillations”, e.g. orthogonal polynomials have exactly n zeros.

Another way (that we do not detail here) to construct orthogonal bases is to consider the eigenvectors of some symmetric operator. We will show bellow that Fourier bases can be obtained this way, by considering translation-invariant operators (convolutions).

2.1.2 Fourier basis on $\mathbb{R}/2\pi\mathbb{Z}$.

There is a flurry of different Fourier basis depending on the space on which it is defined. To get an orthogonal basis, it is important to consider compact spaces (otherwise one obtain a notion of Fourier transform, which is not a decomposition on a basis).

On $L^2(\mathbb{T})$ where $\mathbb{T} = \mathbb{R}/2\pi\mathbb{Z}$, equipped with $\langle f, g \rangle \stackrel{\text{def.}}{=} \frac{1}{2\pi} \int_{\mathbb{T}} f(x)\bar{g}(x)dx$, one can use the Fourier basis

$$\varphi_n(x) \stackrel{\text{def.}}{=} e^{inx} \quad \text{for } n \in \mathbb{Z}. \quad (2.1)$$

One thus has

$$f = \sum_n \hat{f}_n e^{inx} \quad \text{where} \quad \hat{f}_n \stackrel{\text{def.}}{=} \frac{1}{2\pi} \int_0^{2\pi} f(x)e^{-inx}dx, \quad (2.2)$$

in $L^2(\mathbb{T})$ sense. Pointwise convergence is delicate, see Section ??.

Figure ??, left, shows examples of the real part of Fourier atoms.

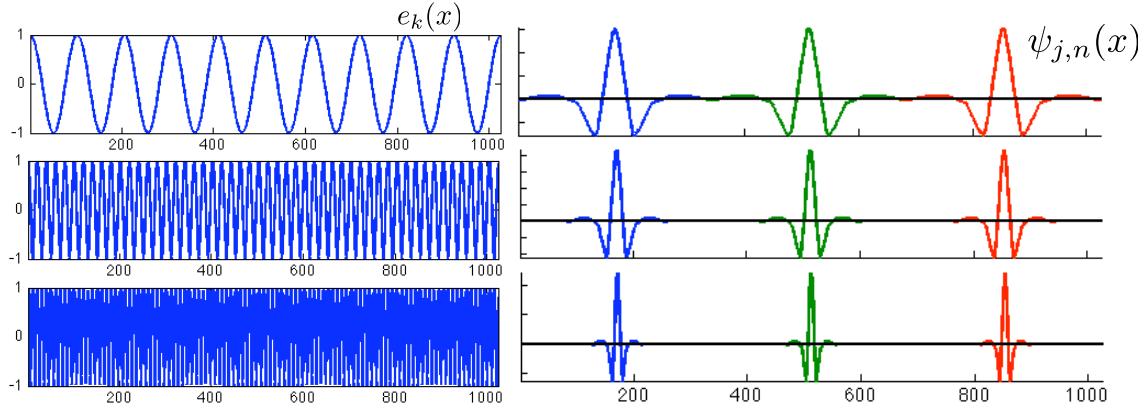


Figure 2.1: Left: 1D Fourier (real part), right: wavelet bases.

We recall that for $f \in L^1(\mathbb{R})$, its Fourier transform is defined as

$$\forall \omega \in \mathbb{R}, \quad \hat{f}(\omega) \stackrel{\text{def.}}{=} \int_{\mathbb{R}} f(x)e^{-ix\omega}dx.$$

and this is extended to $L^2(\mathbb{R})$ by density.

The connexion between the Fourier transform on \mathbb{R} and the Fourier coefficients on \mathbb{T} is given by the following diagram

$$\begin{array}{ccccc} f(x) & \xrightarrow{\mathcal{F}} & \hat{f}(\omega) & & \\ \downarrow & & \downarrow & & \\ \text{sampling} & & \text{Fourier serie} & & \text{periodization} . \\ (f(n))_n & \xrightarrow{\quad} & \sum_n f(n)e^{-i\omega n} & & \end{array}$$

Its commutativity states

$$\sum_n f(n)e^{-i\omega n} = \sum_n \hat{f}(\omega - 2\pi n) \quad (2.3)$$

and this is in fact the celebrated Poisson formula (Proposition ??).

2.2 Convolution on \mathbb{R} and \mathbb{T}

2.2.1 Convolution

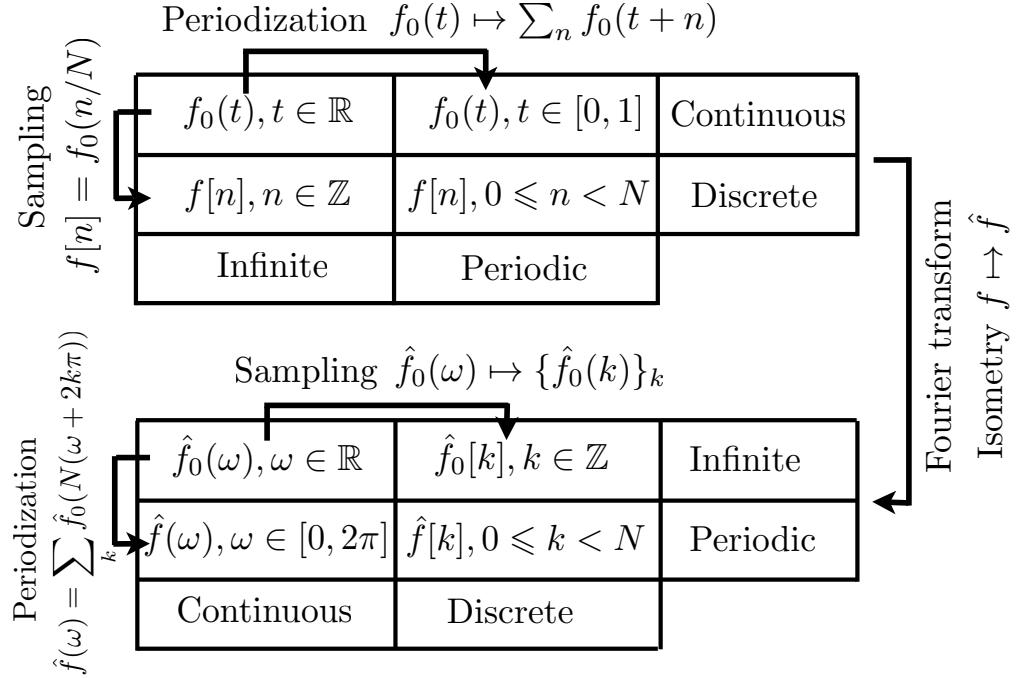


Figure 2.2: The four different settings for Fourier analysis, and the sampling-periodization relationship.

On $\mathbb{X} = \mathbb{R}$ or \mathbb{T} , one defines

$$f * g(x) = \int_{\mathbb{X}} f(t)g(x-t)dt. \quad (2.4)$$

Young's inequality shows that this quantity is well defined if $(f, g) \in L^p(\mathbb{X}) \times L^q(\mathbb{X})$

$$\frac{1}{p} + \frac{1}{q} = 1 + \frac{1}{r} \implies f * g \in L^r(\mathbb{X}) \quad \text{and} \quad \|f * g\|_{L^r(\mathbb{X})} \leq \|f\|_{L^p(\mathbb{X})} \|g\|_{L^q(\mathbb{X})}. \quad (2.5)$$

This shows that if $f \in L^1(\mathbb{X})$, then one has the map $g \in L^p(\mathbb{X}) \mapsto f * g \in L^p(\mathbb{X})$ is a continuous map on $L^p(\mathbb{X})$. Furthermore, when $r = \infty$, $f * g \in \mathcal{C}^0(\mathbb{X})$ is a continuous function (which shows some regularizing effect). Note that for $\mathbb{X} = \mathbb{T}$, $p < q \implies L^q(\mathbb{T}) \subset L^p(\mathbb{T})$, so that $L^\infty(\mathbb{X})$ is the smallest space.

Convolution is mostly used in order to regularize functions. For instance, if $f \in L^1(\mathbb{X})$ and $g \in \mathcal{C}^1(\mathbb{X})$ is bounded, then $f * g$ is differentiable and $(f * g)' = f * g'$. This is used to produce smooth approximate identity $(\rho_\varepsilon = \frac{1}{\varepsilon} \rho(\cdot/\varepsilon))_\varepsilon$ with convergence $f * \rho_\varepsilon \rightarrow f$ in $L^p(\mathbb{X})$ for $1 \leq p < +\infty$ of smooth approximations (and convergence in $L^\infty(\mathbb{X})$ if f is uniformly continuous). This is also used for denoising applications in signal and image processing.

For $(f, g) \in L^1(\mathbb{X})^2$ (so on $\mathbb{X} = \mathbb{T}$, also in any $L^p(\mathbb{X})$), one has

$$\mathcal{F}(f * g) = \hat{f} \odot \hat{g} \quad (2.6)$$

which means that \mathcal{F} is a morphism of algebra. For instance if $\mathbb{X} = \mathbb{R}$, its range is included in the algebra of continuous functions with vanishing limits in $\pm\infty$.

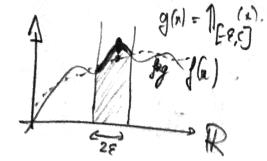


Figure 2.3: Convolution on \mathbb{R} .

$$\begin{array}{ccc} (\hat{f}, \hat{g}) & \xrightarrow{*} & f * g \\ \downarrow \hat{\mathcal{F}} & & \uparrow \mathcal{F}^{-1} \\ (\hat{f}, \hat{g}) & \xrightarrow{\odot} & \hat{f} \odot \hat{g} \end{array}$$

Figure 2.6: Commutative diagram of convolution-Fourier.

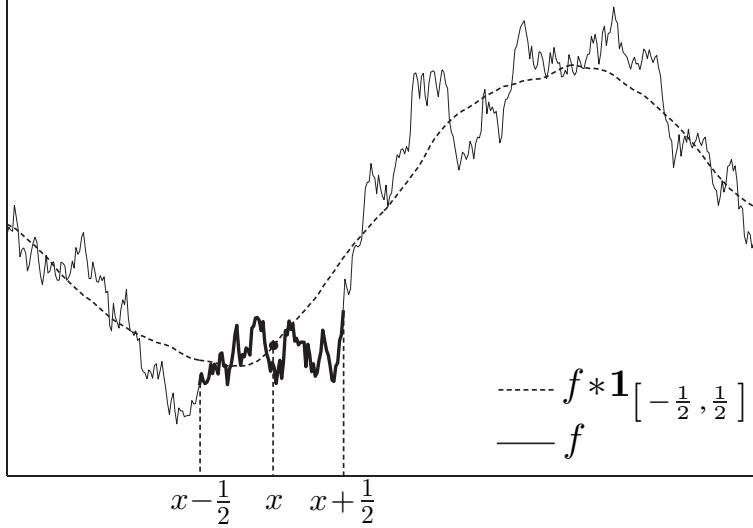


Figure 2.4: Signal filtering with a box filter (running average).

2.2.2 Translation Invariant Operators

Translation invariant operators (which commutes with translation) are fundamental because in most situations, input (signal, image, etc) should be processed without spatial preference. The following propositions shows that any translation invariant¹ (i.e. which commutes with translations) operator is actually a “convolution” against a distribution with bounded Fourier transform. The proof and conclusion (regularity of the convolution kernel) vary depending on the topology on the input and output spaces. We first study the case of convolution mapping to continuous functions.

Proposition 2. *We define $T_\tau f = f(\cdot - \tau)$. A bounded linear operator $H : (L^2(\mathbb{X}), \|\cdot\|_2) \rightarrow (C^0(\mathbb{X}), \|\cdot\|_\infty)$ is such that for all τ , $H \circ T_\tau = T_\tau \circ H$ if and only if*

$$\forall f \in L^2(\mathbb{T}), \quad H(f) = f \star g$$

with $g \in L^2(\mathbb{X})$.

Proof. Thanks to (??) (and the remark in the case $r = \infty$), $T : f \mapsto f \star g$ with $g \in L^2(\mathbb{X})$ is indeed a continuous operator from $L^2(\mathbb{X})$ to $C^0(\mathbb{X})$. Furthermore

$$(H \circ T_\tau)(f) = \int_{\mathbb{X}} f((\cdot - \tau) - y)g(y)d\tau = (f \star g)(\cdot - \tau) = T_\tau(Hf),$$

so that such an H is translation-invariant.

Conversely, we define $\ell : f \mapsto H(f)(0) \in \mathbb{R}$, which is legit since $H(f) \in C^0(\mathbb{X})$. Since H is continuous, there exists C such that $\|Hf\|_\infty \leq C\|f\|_2$, and hence $|\ell(f)| \leq C\|f\|_2$, so that f is a continuous linear form on the Hilbert space $L^2(\mathbb{X})$. Hence, according to Fréchet-Riesz theorem, there exists $h \in L^2(\mathbb{X})$ such that $\ell(f) = \langle f, h \rangle$. Hence, $\forall x \in \mathbb{X}$,

$$H(f)(x) = T_{-x}(Hf)(0) = H(T_{-x}f)(0) = \ell(T_{-x}f) = \langle T_{-x}f, h \rangle = \int_{\mathbb{X}} f(y + x)h(y)dy = f \star \bar{h}(x).$$

where $g \stackrel{\text{def.}}{=} \bar{h} = h(-\cdot) \in L^2(\mathbb{X})$. □

¹One should rather actually say “translation equivariant”.

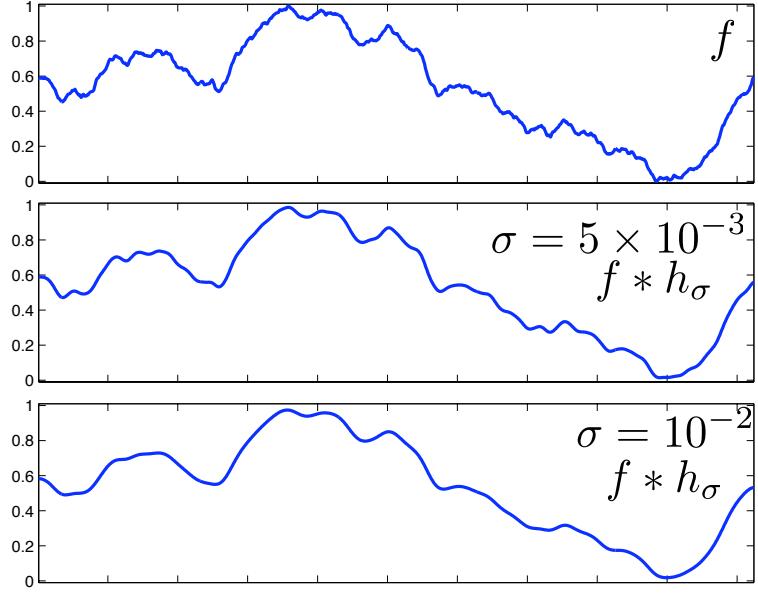


Figure 2.5: Filtering an irregular signal with a Gaussian filter of increasing filter size σ .

We now study, on \mathbb{T} , the case of convolution which can output non-continuous functions. In this case, the kernel can be a “distribution”, so the convolution is defined over the Fourier domain.

Proposition 3. *A bounded linear operator $H : L^2(\mathbb{T}) \rightarrow L^2(\mathbb{T})$ is such that for all τ , $H \circ T_\tau = T_\tau \circ H$ if and only if*

$$\forall f \in L^2(\mathbb{T}), \quad \mathcal{F}(H(f)) = \hat{f} \odot c$$

where $c \in \ell^\infty(\mathbb{Z})$ (a bounded sequence).

Proof. We denote $\varphi_n \stackrel{\text{def.}}{=} e^{in\cdot}$. One has

$$H(\varphi_n) = e^{in\tau} H(T_\tau(\varphi_n)) = e^{in\tau} T_\tau(H(\varphi_n)).$$

Thus, for all n ,

$$\langle H(\varphi_n), \varphi_m \rangle = e^{in\tau} \langle T_\tau \circ H(\varphi_n), \varphi_m \rangle = e^{in\tau} \langle H(\varphi_n), T_{-\tau}(\varphi_m) \rangle = e^{i(n-m)\tau} \langle H(\varphi_n), \varphi_m \rangle$$

So for $n \neq m$, $\langle H(\varphi_n), \varphi_m \rangle = 0$, and we define $c_n \stackrel{\text{def.}}{=} \langle H(\varphi_n), \varphi_n \rangle$. Since H is continuous, $\|Hf\|_{L^2(\mathbb{T})} \leq C\|f\|_{L^2(\mathbb{T})}$ for some constant C , and thus by Cauchy-Schwartz

$$|c_n| = |\langle H(\varphi_n), \varphi_n \rangle| \leq \|H(\varphi_n)\| \|\varphi_n\| \leq C$$

because $\|\varphi_n\| = 1$, so that $c \in \ell^\infty(\mathbb{Z})$. By continuity, recalling that by definition $\hat{f}_n \stackrel{\text{def.}}{=} \langle f, \varphi_n \rangle$,

$$H(f) = \lim_N H\left(\sum_{n=-N}^N \hat{f}_n \varphi_n\right) = \lim_N \sum_{n=-N}^N \hat{f}_n H(\varphi_n) = \lim_N \sum_{n=-N}^N c_n \hat{f}_n \varphi_n = \sum_{n \in \mathbb{Z}} c_n \hat{f}_n \varphi_n$$

so that in particular one has the desired result. \square

This theorem thus states that translation invariant operators are those which are “diagonal” in the Fourier ortho-basis.

$$\begin{array}{ccc} f & \xrightarrow{H} & H(f) \\ T_\tau \downarrow & & \downarrow T_\tau \\ f(-\cdot) & \xrightarrow{H} & H(f(-\cdot)) \end{array}$$

Figure 2.7:
Commutative
diagram for trans-
lation invariance.

2.2.3 Revisiting Poisson formula using distributions.

Informally, the Fourier series

$$\sum_n f(n)e^{-i\omega n}$$

can be thought as the Fourier transform $\mathcal{F}(\Pi_1 \odot f)$ of the discrete distribution

$$\Pi_1 \odot f = \sum_n f(n)\delta_n \quad \text{where} \quad \Pi_s = \sum_n \delta_{sn}$$

for $s \in \mathbb{R}$, where δ_a is the Dirac mass at location $a \in \mathbb{R}$, i.e. the distribution such that $\int f d(\delta_a) = f(a)$ for any continuous f . Indeed, one can multiply a distribution by a continuous function, and the definition of the Fourier transform of a distribution μ is a distributions $\mathcal{F}(\mu)$ such that that

$$\forall g \in \mathcal{S}(\mathbb{R}), \quad \int_{\mathbb{R}} g(x) d\mathcal{F}(\mu) = \int_{\mathbb{R}} \mathcal{F}^*(g) d\mu, \quad \text{where} \quad \mathcal{F}^*(g) \stackrel{\text{def.}}{=} \int_{\mathbb{R}} g(x) e^{ix\cdot} dx,$$

where $\mathcal{S}(\mathbb{R})$ are smooth and rapidly decaying (Schwartz class) functions.

The Poisson formula (??) can thus be interpreted as

$$\mathcal{F}(\Pi_1 \odot f) = \sum_n \hat{f}(\cdot - 2\pi n) = \int_{\mathbb{R}} \hat{f}(\cdot - \omega) d\Pi_{2\pi}(\omega) = \hat{f} \star \Pi_{2\pi}$$

Since $\mathcal{F}^{-1} = \frac{1}{2\pi} \mathcal{S} \circ \mathcal{F}$ where $\mathcal{S}(f) = f(-\cdot)$, one has, applying this operator on both sides

$$\Pi_1 \odot f = \frac{1}{2\pi} \mathcal{S} \circ \mathcal{F}(\hat{f} \star \Pi_{2\pi}) = \mathcal{S}\left(\frac{1}{2\pi} \mathcal{F}(\hat{f}) \odot \hat{\Pi}_{2\pi}\right) = \mathcal{S}\left(\frac{1}{2\pi} \mathcal{F}(\hat{f})\right) \odot \mathcal{S}(\hat{\Pi}_{2\pi}) = \hat{\Pi}_{2\pi} \odot f.$$

This can be interpreted as the relation

$$\hat{\Pi}_{2\pi} = \Pi_1 \implies \hat{\Pi}_1 = 2\pi \Pi_{2\pi}.$$

To intuitively understand this relation, one can compute a finite Fourier series

$$\sum_{n=-N}^N e^{-in\omega} = \frac{\sin((N+1/2)x)}{\sin(x/2)}$$

which is a smooth function which grows unbounded with $N \rightarrow +\infty$ as $N \rightarrow +\infty$.

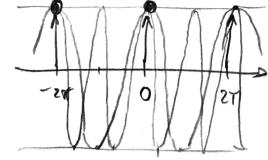


Figure 2.8: Sine wave being summed in the Poisson formula.

2.3 Finite Fourier Transform and Convolution

2.3.1 Discrete Ortho-bases

Discrete signals are finite dimensional vector $f \in \mathbb{C}^N$ where N is the number of samples and where each f_n is the value of the signal at a 1D or 2D location. For a 2-D images $f \in \mathbb{C}^N \simeq \mathbb{C}^{N_0 \times N_0}$, $N = N_0 \times N_0$, where N_0 is the number of pixels along each direction.

Discrete signals and images are processed using a discrete inner product that mimics the continuous L^2 inner product

$$\langle f, g \rangle = \sum_{n=0}^{N-1} f_n \bar{g}_n.$$

One thus defines a distance between discretized vectors as

$$\|f - g\|^2 = \sum_{n=0}^{N-1} |f_n - g_n|^2.$$

Exactly as in the continuous case, a discrete orthogonal basis $\{\psi_m\}_{0 \leq m < N}$ of \mathbb{C}^N , satisfies

$$\langle \psi_m, \psi_{m'} \rangle = \delta_{m-m'}. \quad (2.7)$$

The decomposition of a signal in such an ortho-basis is written

$$f = \sum_{m=0}^{N-1} \langle f, \psi_m \rangle \psi_m.$$

It satisfies a conservation of energy

$$\|f\|^2 = \sum_{n=0}^{N-1} |f_n|^2 = \sum_{m=0}^{N-1} |\langle f, \psi_m \rangle|^2$$

Computing the set of all inner product $\{\langle f, \psi_m \rangle\}_{0 \leq m < N}$ is done in a brute force way in $O(N^2)$ operations. This is not feasible for large datasets where N is of the order of millions. When designing an ortho-basis, one should keep this limitation in mind and enforce some structure in the basis elements so that the decomposition can be computed with fast algorithm. This is the case for the Fourier and wavelet bases, that enjoy respectively $O(N \log(N))$ and $O(N)$ algorithms.

2.3.2 Discrete Fourier transform

We denote $f = (f_n)_{n=0}^{N-1} \in \mathbb{R}^N$, but we insist that such vector should really be understood as being indexed by $n \in \mathbb{Z}/N\mathbb{Z}$, which is a finite commutative group for the addition. This corresponds to using periodic boundary conditions.

The discrete Fourier transform is defined as

$$\forall k = 0, \dots, N-1, \quad \hat{f}_k \stackrel{\text{def.}}{=} \sum_{n=0}^{N-1} f_n e^{-\frac{2i\pi}{N} kn} = \langle f, \varphi_k \rangle \quad \text{where} \quad \varphi_k \stackrel{\text{def.}}{=} (e^{\frac{2i\pi}{N} kn})_{n=0}^{N-1} \in \mathbb{C}^N \quad (2.8)$$

where the canonical inner product on \mathbb{C}^N is $\langle u, v \rangle = \sum_{n=1}^N u_n \bar{v}_n$ for $(u, v) \in (\mathbb{C}^N)^2$. This definition can intuitively be motivated by sampling the Fourier basis $x \mapsto e^{ikx}$ on $\mathbb{R}/2\pi\mathbb{Z}$ at equi-spaced points $(\frac{2\pi}{N}n)_{n=0}^{N-1}$. The following proposition shows that this corresponds to a decomposition in an ortho-basis.

Proposition 4. *One has*

$$\langle \varphi_k, \varphi_\ell \rangle = \begin{cases} N & \text{if } k = \ell, \\ 0 & \text{otherwise.} \end{cases}$$

In particular, this implies

$$\forall n = 0, \dots, N-1, \quad f_n = \frac{1}{N} \sum_k \hat{f}_k e^{\frac{2i\pi}{N} kn}. \quad (2.9)$$

Proof. One has, if $k \neq \ell$

$$\langle \varphi_k, \varphi_\ell \rangle = \sum_n e^{\frac{2i\pi}{N} (k-\ell)n} = \frac{1 - e^{\frac{2i\pi}{N} (k-\ell)N}}{1 - e^{\frac{2i\pi}{N} (k-\ell)}} = 0$$

which is the sum of a geometric serie (equivalently, sum of equi-spaced points on a circle). The inversion formula is simply $f = \sum_k \langle f, \varphi_k \rangle \frac{\varphi_k}{\|\varphi_k\|^2}$. \square

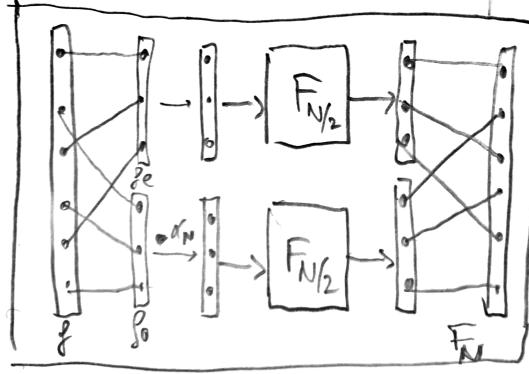


Figure 2.9: Diagram of one step of the FFT.

2.3.3 Fast Fourier transform

Assuming $N = 2N'$, one has

$$\begin{aligned}\hat{f}_{2k} &= \sum_{n=0}^{N'-1} (f_n + f_{n+N/2}) e^{-\frac{2i\pi}{N'} kn} \\ \hat{f}_{2k+1} &= \sum_{n=0}^{N'-1} e^{-\frac{2i\pi}{N'} n} (f_n - f_{n+N/2}) e^{-\frac{2i\pi}{N'} kn}.\end{aligned}$$

For the second line, we used the computation

$$e^{-\frac{2i\pi}{N}(2k+1)(n+N/2)} = e^{-\frac{2i\pi}{N}(2kn+kN+n+N/2)} = -e^{-\frac{2i\pi}{N}n} e^{-\frac{2i\pi}{N}kn}.$$

Denoting $\mathcal{F}_N(f) = \hat{f}$ the discrete Fourier transform on \mathbb{R}^N , and introducing the notation $f_e = (f_n + f_{n+N/2})_n \in \mathbb{R}^{N'}$, $f_o = (f_n - f_{n+N/2})_n \in \mathbb{R}^{N'}$ and $\alpha_N = (e^{-\frac{2i\pi}{N}n})_n \in \mathbb{R}^{N'}$, one has the following recursion formula

$$\mathcal{F}_N(f) = \mathcal{I}_N(\mathcal{F}_{N/2}(f_e), \mathcal{F}_{N/2}(f_o \odot \alpha_N))$$

where \mathcal{I}_N is the ‘‘interleaving’’ operator, defined by $\mathcal{I}_N(a, b) \stackrel{\text{def.}}{=} (a_1, b_1, a_2, b_2, \dots, a_{N'}, b_{N'})$. These iterations define the so-called Fast Fourier Transform algorithm, which works here when N is a power of 2. These iterations can be extended to arbitrary number N , but a workaround is to simply pad with 0 (or use more complicated extensions) to have vector with size that are power of 2.

Denoting $C(N)$ the numerical complexity (number of elementary operations) associated to the computation of \hat{f} , one thus has

$$C(N) = 2C(N/2) + NK \quad (2.10)$$

where K is the complexity of N complex additions and $N/2$ multiplications. Making the change of variable

$$\ell \stackrel{\text{def.}}{=} \log_2(N) \quad \text{and} \quad T(\ell) \stackrel{\text{def.}}{=} \frac{C(N)}{N}$$

i.e. $C(N) = 2^\ell T(\ell)$, the relation (??) becomes

$$2^\ell T(\ell) = 2 \times 2^{\ell-1} T(\ell-1) + 2^\ell K \implies T(\ell) = T(\ell-1) + K \implies T(\ell) = T(0) + K\ell$$

and using the fact that $T(0) = C(1)/1 = 0$, one obtains

$$C(N) = KN \log_2(N).$$

This complexity should be contrasted with the complexity $O(N^2)$ of directly computing the N coefficients (??), each involving a sum of size N .

2.3.4 Finite convolution

For $(f, g) \in (\mathbb{R}^N)^2$, one defines $f \star g \in \mathbb{R}^N$ as

$$\forall n = 0, \dots, N-1, \quad (f \star g)_n \stackrel{\text{def.}}{=} \sum_{k=0}^{N-1} f_k g_{n-k} = \sum_{k+\ell=n} f_k g_\ell \quad (2.11)$$

where one should be careful that here $+$ and $-$ should be understood modulo N (vectors should be seen as being defined on the group $\mathbb{Z}/N\mathbb{Z}$, or equivalently, one uses periodic boundary conditions). This defines an commutative algebra structure $(\mathbb{R}^N, +, \star)$, with neutral element the “Dirac” $\delta_0 \stackrel{\text{def.}}{=} (1, 0, \dots, 0)^\top \in \mathbb{R}^N$. The following proposition shows that $\mathcal{F} : f \mapsto \hat{f}$ is an algebra bijective isometry (up to a scaling by \sqrt{N} of the norm) mapping to $(\mathbb{R}^N, +, \odot)$ with neutral element $\mathbb{1}_N = (1, \dots, 1) \in \mathbb{R}^N$.

Proposition 5. *One has $\mathcal{F}(f \star g) = \hat{f} \odot \hat{g}$.*

Proof. We denote $T : g \mapsto f \star g$. One has

$$(T\varphi_\ell)_n = \sum_k f_k e^{\frac{2i\pi}{N}\ell(n-k)} = e^{\frac{2i\pi}{N}\ell n} \hat{f}_\ell.$$

This shows that $(\varphi_\ell)_\ell$ are the N eigenvectors of T with associated eigenvalues \hat{f}_ℓ . So T is diagonalizable in this basis. Denoting $F = (e^{-\frac{2i\pi}{N}kn})_{k,n}$ the matrix of the Fourier transform, the Fourier inversion formula (??) reads $F^{-1} = \frac{1}{N}F^*$ where $F^* = \bar{F}^\top$ is the adjoint matrix (trans-conjugate). The diagonalization of T now reads

$$T = F^{-1} \operatorname{diag}(\hat{f}) F = \implies \mathcal{F}(Tg) = \operatorname{diag}(\hat{f}) F g \implies \mathcal{F}(f \star g) = \operatorname{diag}(\hat{f}) \hat{g}.$$

□

This proposition shows that one can compute in $O(N \log(N))$ operation via the formula

$$f \star g = \mathbb{F}^{-1}(\hat{f} \odot \hat{g}).$$

This is very advantageous with respect to the naive implementation of formula (??), in the case where f and g have large support. In case where $|\operatorname{Supp}(g)| = P$ is small, then direct implementation is $O(PN)$ which might be advantageous. An example is $g = [1, 1, 0, \dots, 0, 1]/3$, the moving average, where

$$(f \star g)_n = \frac{f_{n-1} + f_n + f_{n+1}}{3}$$

needs $3N$ operations.

An example of application of the FFT is the multiplication of large polynomial, and thus of large integers (viewing the expansion in a certain basis as a polynomial). Indeed

$$\left(\sum_{i=0}^A a_i X_i \right) \left(\sum_{j=0}^B b_j X^j \right) = \sum_{k=0}^{A+B} \left(\sum_{i+j=k} a_i b_j \right) X^k$$

One can write $\sum_{i+j=k} a_i b_j = (\bar{a} \star \bar{b})_k$ when one defines $\bar{a}, \bar{b} \in \mathbb{R}^{A+B}$ by zero padding.

2.4 Discretisation Issues

Beside computing convolutions, another major application of the FFT is to approximate the Fourier transform and its inverse, thus leading to a computationally efficient spectral interpolation method.

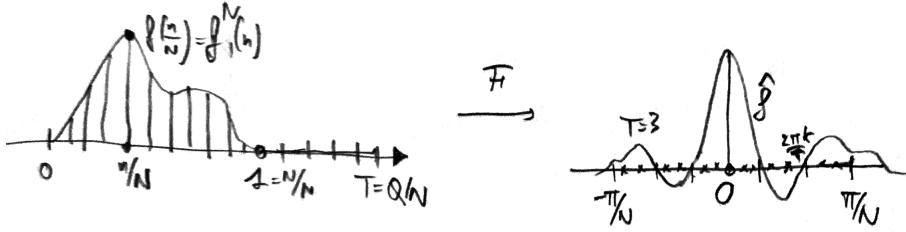


Figure 2.10: Fourier transform approximation by zero-padding in the spatial domain.

2.4.1 Fourier approximation via spatial zero padding.

It is possible to view the discrete finite Fourier transform (??) as a first order approximation to compute Fourier coefficients, or rather actually samples from the Fourier transform (??). Supposing that f is a smooth enough function supported on $[0, 1]$, we consider the discrete Fourier transform of the vector $f^Q \stackrel{\text{def.}}{=} (f(n/N))_{n=0}^{Q-1} \in \mathbb{R}^Q$ where $Q \geq N$ induced a padding by 0 (since $f(n/N) = 0$ for $n > N$)

$$\forall k \in [-\frac{Q}{2}, \frac{Q}{2}], \quad \frac{1}{N} \hat{f}_k^Q = \frac{1}{N} \sum_{n=0}^{N-1} f\left(\frac{n}{N}\right) e^{-\frac{2i\pi}{Q} nk} \approx \int_0^1 f(x) e^{-\frac{2ki\pi}{T} x} dx = \hat{f}\left(\frac{2k\pi}{T}\right) \quad \text{where } T \stackrel{\text{def.}}{=} \frac{Q}{N}.$$

The approximation is first order accurate, i.e. $O(1/N)$ for a C^1 function f . Increasing the amount Q of zero padding is a way to compute larger frequencies. Increasing the discretization precision N is on contrary a way to increase the precision of the Fourier sampling (using smaller step size $2\pi/T$).

2.4.2 Fourier approximation via spatial zero padding.

If one has at its disposal N uniform discrete samples $f^N = (f_n^N)_{n=0}^{N-1}$, one can compute its discrete Fourier transform $\mathcal{F}(f^N) = \hat{f}^N$ (in $O(N \log(N))$ operations with the FFT),

$$\hat{f}_k^N \stackrel{\text{def.}}{=} \sum_{n=0}^{N-1} f_n^N e^{-\frac{2i\pi}{N} nk},$$

and then zero-pad it to obtain a vector of length Q . For simplicity, we assume $N = 2N' + 1$ is odd, and this computation can be also done (but is more involved) with even size. Indexing the frequencies as $-N' \leq k \leq N'$ The padding vector is of the form,

$$\tilde{f}^Q \stackrel{\text{def.}}{=} (0, \dots, 0, \hat{f}^N, 0, \dots, 0) \in \mathbb{R}^Q$$

One can then compute the (with a normalization constant Q/N) inverse discrete Fourier transform of size Q (in $O(Q \log(Q))$ operations with the FFT) to obtain

$$\begin{aligned} \frac{Q}{N} \mathcal{F}^{-1}(\tilde{f}^Q)_\ell &= \frac{Q}{N} \times \frac{1}{N} \sum_{k=-N'}^{N'} \hat{f}_k^N e^{\frac{2i\pi}{Q} \ell k} = \frac{1}{N} \sum_{k=-N'}^{N'} \sum_{n=0}^{N-1} f_n^N e^{\frac{2i\pi}{N} nk} e^{\frac{2i\pi}{Q} \ell k} \\ &= \sum_{n=0}^{N-1} f_n^N \frac{1}{N} \sum_{k=-N'}^{N'} e^{2i\pi(-\frac{n}{N} + \frac{\ell}{Q})k} = \sum_{n=0}^{N-1} f_n^N \frac{\sin\left[\pi N \left(\frac{\ell}{Q} - \frac{n}{N}\right)\right]}{N \sin\left[\pi \left(\frac{\ell}{Q} - \frac{n}{N}\right)\right]} \\ &= \sum_{n=0}^{N-1} f_n^N \text{sinc}_N\left(\frac{\ell}{T} - n\right) \quad \text{where } T \stackrel{\text{def.}}{=} \frac{Q}{N} \quad \text{and} \quad \text{sinc}_N(u) \stackrel{\text{def.}}{=} \frac{\sin(\pi u)}{N \sin(\pi u/N)}. \end{aligned}$$

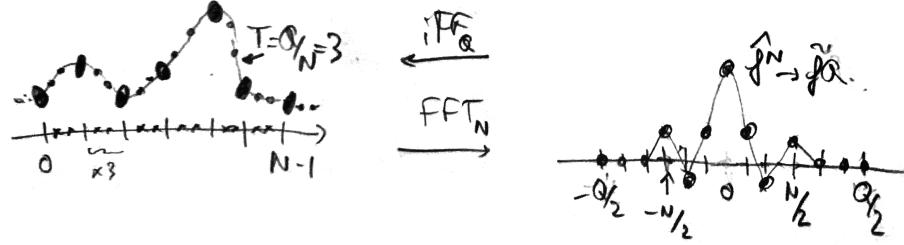


Figure 2.11: Interpolation by zero-padding in the frequency domain.

Here we use the following summation rule for geometric series for $\rho = e^{i\omega}$, $a = -b$, $\omega = 2\pi \left(-\frac{n}{N} + \frac{\ell}{Q} \right)$,

$$\sum_{i=a}^b \rho^i = \frac{\rho^{a-\frac{1}{2}} - \rho^{b+\frac{1}{2}}}{\rho^{-\frac{1}{2}} - \rho^{\frac{1}{2}}} = \frac{\sin((b+\frac{1}{2})\omega)}{\sin(\omega/2)}.$$

This zero-padding method leads to a discrete version of the Shannon interpolation formula (??), which allows to comput the interpolation on a grid of size Q are cost $O(Q \log(Q))$. Increasing N increases the accuracy of the formula, since $\text{sinc}_N \rightarrow \text{sinc}$ as $N \rightarrow +\infty$.

2.5 Fourier in Multiple Dimensions

The Fourier transform is extended from 1-D to arbitrary finite dimension $d > 1$ by tensor product.

2.5.1 On Continuous Domains

On \mathbb{R}^d . The crux of the power of Fourier transform in arbitrary dimension is that a product of elementary 1-D sine waves is still a sine wave

$$\prod_{\ell=1}^d e^{ix_\ell \omega_\ell} = e^{i\langle x, \omega \rangle}$$

moving orthogonally to the wave vector $\omega = (\omega_\ell)_{\ell=1}^d \in \mathbb{R}^d$. Here $\langle x, \omega \rangle = \sum_\ell x_\ell \omega_\ell$ is the canonical inner product on \mathbb{R}^d .

The definition of the Fourier transform and its inverse are

$$\begin{aligned} \forall \omega \in \mathbb{R}^d, \quad \hat{f}(\omega) &\stackrel{\text{def.}}{=} \int_{\mathbb{R}^d} f(x) e^{-i\langle x, \omega \rangle} dx, \\ \forall x \in \mathbb{R}^d, \quad f(x) &= \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} f(x) e^{i\langle x, \omega \rangle} d\omega, \end{aligned}$$

under hypotheses of integrability matching exactly those in 1-D.

On $(\mathbb{R}/2\pi\mathbb{Z})^d$. Given an Hilbertian basis $(\varphi_{n_1})_{n_1 \in \mathbb{N}}$ of $L^2(\mathbb{X})$, one construct an Hilbertian basis of $L^2(\mathbb{X}^d)$ by tensorization

$$\forall n = (n_1, \dots, n_d) \in \mathbb{N}^d, \quad \forall x \in \mathbb{X}^d, \quad \varphi_n(x) = \varphi_{n_1}(x_1) \dots \varphi_{n_d}(x_d). \quad (2.12)$$

Orthogonality is simple to check, and one can also prove convergence for sum of the form $\sum_{\|n\|_\infty \leq N} \langle f, \varphi_n \rangle \varphi_n \rightarrow f$ in $L^2(\mathbb{X}^d)$.

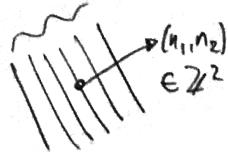


Figure 2.12: 2-D sine wave.

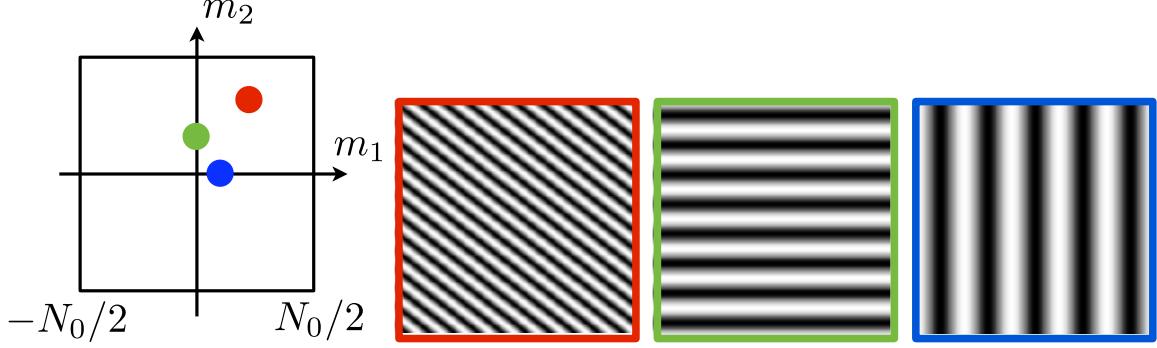


Figure 2.13: 2D Fourier orthogonal bases.

For the multi-dimensional torus $(\mathbb{R}/2\pi\mathbb{Z})^d$, using the Fourier basis (??), this leads to consider the basis

$$\forall n \in \mathbb{R}^d, \quad \varphi_n(x) = e^{i\langle x, n \rangle}$$

which is indeed an Hilbertian orthonormal basis for the inner product $\langle f, g \rangle \stackrel{\text{def.}}{=} \frac{1}{(2\pi)^d} \int_{\mathbb{T}^d} f(x)\bar{g}(x)dx$. This defines the Fourier transform and the reconstruction formula on $L^2(\mathbb{T}^d)$

$$\hat{f}_n \stackrel{\text{def.}}{=} \frac{1}{(2\pi)^d} \int_{\mathbb{T}^d} f(x)e^{-i\langle x, n \rangle} dx \quad \text{and} \quad f = \sum_{n \in \mathbb{Z}^d} \hat{f}_n e^{i\langle x, n \rangle}.$$



Figure 2.14: The 2-dimensional torus $\mathbb{T}^2 = (\mathbb{R}/2\pi\mathbb{Z})^2$

2.5.2 On Discrete Domains

Discrete Fourier Transform. On d -dimensional discrete domain of the form

$$n = (n_1, \dots, n_d) \in \mathbb{Y}_d \stackrel{\text{def.}}{=} [\![1, N_1]\!] \times \dots \times [\![1, N_d]\!]$$

(we denote $[\![a, b]\!] \stackrel{\text{def.}}{=} \{i \in \mathbb{Z} ; a \leq i \leq b\}$) of $N = N_1 \dots N_d$ points, with periodic boundary conditions, one defines an orthogonal basis $(\varphi_k)_k$ by the same tensor product formula as (??) but using the 1-D discrete Fourier basis (??)

$$\forall (k, n) \in \mathbb{Y}_d^2, \quad \varphi_k(n) = \varphi_{k_1}(n_1) \dots \varphi_{k_d}(n_d) = \prod_{\ell=1}^d e^{\frac{2i\pi}{N_\ell} k_\ell n_\ell} = e^{2i\pi \langle k, n \rangle_{\mathbb{Y}_d}}$$

(2.13)

where we used the (rescaled) inner product

$$\langle k, n \rangle_{\mathbb{Y}_d} \stackrel{\text{def.}}{=} \sum_{\ell=1}^d \frac{k_\ell n_\ell}{N_\ell}. \quad (2.14)$$

The basis $(\varphi_k)_k$ is indeed orthonormal for this inner product. The Fourier transform gathers inner products in this basis, and (similarly to the 1-D case) the convention is to not normalize them with $(N_\ell)_\ell$, so that

$$\begin{aligned} \forall k \in \mathbb{Y}_d, \quad \hat{f}_k &\stackrel{\text{def.}}{=} \sum_{n \in \mathbb{Y}_d} f_n e^{-i\langle k, n \rangle_{\mathbb{Y}_d}}, \\ \forall n \in \mathbb{Y}_d, \quad f_n &= \frac{1}{N} \sum_{k \in \mathbb{Y}_d} \hat{f}_k e^{i\langle k, n \rangle_{\mathbb{Y}_d}}. \end{aligned}$$

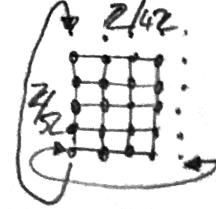


Figure 2.15: Discrete 2-D torus.

Fast Fourier Transform. We detail the algorithm in dimension $d = 2$ for notation simplicity, but this extends similarly in arbitrary dimension. The general idea is that if a fast algorithm is available to compute ortho-decompositions on two 1-D bases $(\varphi_{k_1}^1)_{k_1=1}^{N_1}$, $(\varphi_{k_2}^2)_{k_2=1}^{N_2}$, is extended to compute decomposition on the tensor product basis $(\varphi_{k_1}^1 \otimes \varphi_{k_2}^2)_{k_1, k_2}$ by applying successively the algorithm on the “rows” and then “columns” (the order does not matter) of the matrix $(f_n)_{n=(n_1, n_2)} \in \mathbb{R}^{N_1 \times N_2}$. Indeed

$$\forall k = (k_1, k_2), \quad \langle f, \varphi_{k_1}^1 \otimes \varphi_{k_2}^2 \rangle = \sum_{n=(n_1, n_2)} f_n \varphi_{k_1}^1(n_1) \varphi_{k_2}^2(n_2) = \sum_{n_1} \left(\sum_{n_2} f_{n_1, n_2} \varphi_{k_1}^1(n_2) \right) \varphi_{k_1}^1(n_1).$$

Denoting $C(N_1)$ the complexity of the 1-D algorithm on \mathbb{R}^{N_1} , the complexity of the resulting 2-D decomposition is $N_2 C(N_1) + N_1 C(N_2)$, and hence for the FFT, it is $O(N_1 N_2 \log(N_1 N_2)) = O(N \log(N))$ for $N = N_1 N_2$.

If we represent $f \in \mathbb{R}^{N_1 \times N_2}$ as a matrix, and denote $F_N = (e^{-\frac{2i\pi}{N} kn})_{k,n}$ the Fourier transform matrix (or the matrix where rows are the φ_k^*), then one can compute the 2-D Fourier transform as matrix-matrix products

$$\hat{f} = F_{N_1} \times f \times F_{N_2}^* \in \mathbb{R}^{N_1 \times N_2}.$$

But of course these multiplications are not computed explicitly (one uses the FFT).

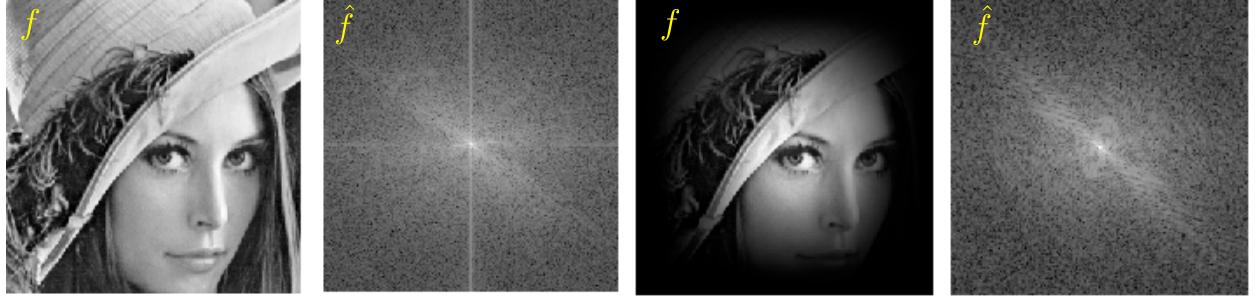


Figure 2.16: 2D Fourier analysis of an image (left), and attenuation of the periodicity artifact using masking (right).

2.5.3 Shannon sampling theorem.

The sampling Theorem ?? extends easily to \mathbb{R}^d by tensorization, assuming that the sampling is on a uniform Cartesian grid. In 2-D for instance, if $\text{supp}(\hat{f}) \subset [-\pi/s_1, \pi/s_1] \times [-\pi/s_2, \pi/s_2]$ and f is decaying fast enough,

$$\forall x \in \mathbb{R}^2, \quad f(x) = \sum_{n \in \mathbb{Z}^2} f(n_1 s_1, n_2 s_2) \text{sinc}(x_1/s_1 - n_1) \text{sinc}(x_2/s_2 - n_2) \quad \text{where} \quad \text{sinc}(u) = \frac{\sin(\pi u)}{\pi u}.$$

2.5.4 Convolution in higher dimension.

Convolution on \mathbb{X}^d with $\mathbb{X} = \mathbb{R}$ or $\mathbb{X} = \mathbb{R}/2\pi\mathbb{Z}$ are defined in the very same way as in 1-D (??) as

$$f \star g(x) = \int_{\mathbb{X}^d} f(t)g(x-t)dt.$$

Similarly, finite discrete convolution of vectors $f \in \mathbb{R}^{N_1 \times N_2}$ extend formula (??) as

$$\forall n \in \llbracket 0, N_1 - 1 \rrbracket \times \llbracket 0, N_2 - 1 \rrbracket, \quad (f \star g)_n \stackrel{\text{def.}}{=} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} f_k g_{n-k}$$

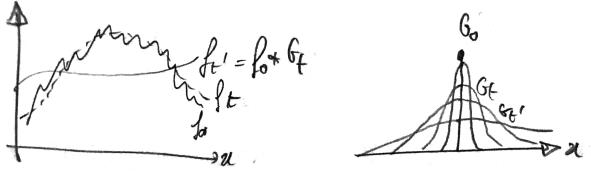


Figure 2.17: Heat diffusion as a convolution.

where additions and subtractions of vectors are performed modulo (N_1, N_2) .

The Fourier-convolution theorem is still valid in all this cases, namely $\mathcal{F}(f \star g) = \hat{f} \odot \hat{g}$. In the finite case, this offers a fast $O(N \log(N))$ method to compute convolutions even if f and g do not have small support.

2.6 Application to ODEs and PDEs

2.6.1 On Continuous Domains

We here give only the intuition without formal proofs.

One $\mathbb{X} = \mathbb{R}$ or \mathbb{T} , one has

$$\mathcal{F}(f^{(k)})(\omega) = (\mathrm{i}\omega)^k \hat{f}(\omega).$$

Intuitively, $f^{(k)} = f \star \delta^{(k)}$ where $\delta^{(k)}$ is a distribution with Fourier transform $\hat{\delta}^{(k)}(\omega) = (\mathrm{i}\omega)^k$. Similarly on $\mathbb{X} = \mathbb{R}^d$ (see Section ?? for the definition of the Fourier transform in dimension d), one has

$$\mathcal{F}(\Delta f)(\omega) = -\|\omega\|^2 \hat{f}(\omega) \quad (2.15)$$

(and similarly on \mathbb{T} replacing ω by $n \in \mathbb{Z}^d$). The Fourier transform (or Fourier coefficients) are thus powerful to study linear differential equations with constant coefficients, because they are turned into algebraic equations.

As a typical example, we consider the heat equation

$$\frac{\partial f_t}{\partial t} = \Delta f_t \implies \forall \omega, \quad \frac{\partial \hat{f}_t(\omega)}{\partial t} = -\|\omega\|^2 \hat{f}(\omega).$$

This shows that $\hat{f}_t(\omega) = \hat{f}_0(\omega) e^{-\|\omega\|^2 t}$ and by inverse Fourier transform and the convolution theorem

$$f_t = G_t \star f_0 \quad \text{where} \quad G_t = \frac{1}{(4\pi t)^{d/2}} e^{-\frac{\|x\|^2}{4t}}$$

which is a Gaussian of standard deviation $\sqrt{2t}$.

2.6.2 Finite Domain and Discretization

On $\mathbb{R}/N\mathbb{Z}$ (i.e. discrete domains with periodic boundary conditions), one typically considers forward finite differences (first and second order)

$$D_1 f \stackrel{\text{def.}}{=} N(f_{n+1} - f_n)_n = f \star d_1 \quad \text{where} \quad d_1 = [-1, 0, \dots, 0, 1]^\top \in \mathbb{R}^N, \quad (2.16)$$

$$D_2 f = D_1^\top D_1 f \stackrel{\text{def.}}{=} N^2(f_{n+1} + f_{n-1} - 2f_n)_n = f \star d_2 \quad \text{where} \quad d_2 = d_1 \star \bar{d}_1 = [2, -1, 0, \dots, 0, -1]^\top \in \mathbb{R}^N. \quad (2.17)$$

Thanks to Proposition ??, one can alternatively computes

$$\mathcal{F}(D_2 f) = \hat{d}_2 \odot \hat{f} \quad \text{where} \quad (\hat{d}_2)_k = N^2(e^{\frac{2i\pi}{N}} + e^{-\frac{2i\pi}{N}} - 2) = -4N^2 \sin\left(\frac{\pi k}{N}\right)^2. \quad (2.18)$$

For $N \gg k$, one thus has $(\hat{d}_2)_k \sim -(2\pi k)^2$ which matches the scaling of ??.

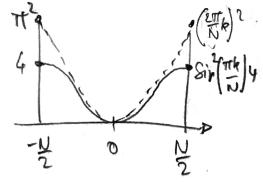


Figure 2.18: Comparison of the spectrum of Δ and D_2 .

2.7 A Bit of Group Theory

The reference for this section is [?].

2.7.1 Characters

For $(G, +)$ a commutative group, a character is a group morphism $\chi : (G, +) \rightarrow (\mathbb{C}^*, \cdot)$, i.e. it satisfies

$$\forall (n, m) \in G, \quad \chi(n + m) = \chi(n)\chi(m).$$

The set of characters is the so-called dual (\hat{G}, \odot) and is a group for the pointwise multiplication $(\chi_1 \odot \chi_2)(n) \stackrel{\text{def.}}{=} \chi_1(n)\chi_2(n)$. Indeed, the inverse of a character χ is $\chi^{-1}(n) = \chi(-n)$.

Note that for a finite group G with $|G| = N$, then since $N \times n = 0$ for any $n \in G$, then $\chi(n)^N = \chi(Nn) = \chi(0) = 1$, so that characters assume values in the unit circle, and more precisely

$$\chi(n) \in \left\{ e^{\frac{2i\pi}{N}k} ; 0 \leq k \leq N-1 \right\}. \quad (2.19)$$

So in particular \hat{G} is a finite group (since there is a finite number of applications between two finite sets) and $\chi^{-1} = \bar{\chi}$. In the case of a cyclic group, the dual is actually simple to describe.

Proposition 6. For $G = \mathbb{Z}/N\mathbb{Z}$, then $\hat{G} = (\varphi_k)_{k=0}^{N-1}$ where $\varphi_k = (e^{\frac{2i\pi}{N}nk})_n$ and $k \mapsto \varphi_k$ defines a (non-canonical) isomorphism $G \sim \hat{G}$.

Proof. The φ_k are indeed characters.

Conversely, for any $\chi \in \hat{G}$, according to ??, $\chi(1) = e^{\frac{2i\pi}{N}k}$ for some k . Then

$$\chi(n) = \chi(1)^n = e^{\frac{2i\pi}{N}kn} = \varphi_k(n).$$

Note that all these applications are different (because $\varphi_k(1)$ are all distincts) which shows that $|G| = |\hat{G}|$ so that they are isomorphic. \square

This proposition thus shows that characters of cyclic groups are exactly the discrete Fourier orthonormal basis defined in ??.

Commutative groups. For more general commutative groups with a finite number of generators, according to the celebrated structure theorem, one can “decompose” them as a product of cyclic groups (which are in some sense the basic building blocks), i.e. there is the following isomorphism of groups

$$G \sim (\mathbb{Z}/N_1\mathbb{Z}) \times \dots \times (\mathbb{Z}/N_d\mathbb{Z}) \times \mathbb{Z}^Q. \quad (2.20)$$

If G is finite, then $Q = 0$ and $N = N_1 \times N_d$. In this case, G is simply a discrete d -dimensional “rectangle” with periodic boundary conditions.

For two finite groups (G_1, G_2) one has

$$\widehat{G_1 \times G_2} = \hat{G}_1 \otimes \hat{G}_2 = \left\{ \chi_1 \otimes \chi_2 ; (\chi_1, \chi_2) \in \hat{G}_1 \times \hat{G}_2 \right\}. \quad (2.21)$$

Here \otimes is the tensor product of two functions

$$\forall (n_1, n_2) \in G_1 \times G_2, \quad (\chi_1 \otimes \chi_2)(n_1, n_2) \stackrel{\text{def.}}{=} \chi_1(n_1)\chi_2(n_2).$$

Indeed, one verifies that $\chi_1 \otimes \chi_2$ is a morphism, and in fact one has the factorization $\chi = \chi(\cdot, 0) \otimes \chi(0, \cdot)$ because one decomposes $(n_1, n_2) = (n_1, 0) + (0, n_2)$.

This construction, thanks to the structure theorem, leads to a constructive proof of the isomorphism theorem.

Proposition 7. *If G is commutative and finite then $\hat{G} \sim G$.*

Proof. The structure theorem (??) for $Q = 0$ and the dual of a product (??) shows that

$$\hat{G} \sim \hat{G}_1 \otimes \dots \otimes \hat{G}_d$$

where we denoted $G_\ell \stackrel{\text{def.}}{=} \mathbb{Z}/N_\ell\mathbb{Z}$. One then remark that $\hat{G}_1 \otimes \hat{G}_2 \sim G_1 \times G_2$. One conclude thanks to Proposition ??, since one has $\hat{G}_k \sim G_k$. \square

Note that the isomorphism $\hat{G} \sim G$ is not “canonical” since it depends on the indexing of the roots of unity on the circle. Similarly to the case of duality of vector space, the isomorphism $\hat{G} \sim G$ can be made canonical by considering the evaluation map

$$g \in G \longmapsto e_g \in \hat{G} \quad \text{where} \quad (e_g : \chi \in \hat{G} \mapsto \chi(g) \in \mathbb{C}^*).$$

Discrete Fourier transform from character's point of view. One can be even more constructive by remarking that characters in \hat{G}_ℓ are the discrete Fourier atoms (??), i.e. are of the form

$$(e^{\frac{2i\pi}{N_\ell} k_\ell n_\ell})_{n_\ell=0}^{N_\ell-1} \quad \text{for some } 0 \leq k_\ell < N_\ell.$$

Identifying G and $G_1 \times \dots \times G_d$, by tensorizing these functions together, one thus obtains that the characters composing \hat{G} are exactly the orthogonal multi-dimensional discrete Fourier basis (??).

2.7.2 More General cases

Infinite groups. For an infinite group with a finite number of generator, one has $Q > 0$, and the definition of \hat{G} should impose the continuity of the characters (and also use an invariant measure on G to define inner products). In the case $G = \mathbb{Z}$, the dual are indexed by a continuous parameter,

$$\hat{\mathbb{Z}} = \{\varphi_\omega : n \mapsto e^{in\omega} \in L^2(\mathbb{R}/2\pi\mathbb{Z}) ; \omega \in \mathbb{R}/2\pi\mathbb{Z}\}$$

so that $\hat{\mathbb{Z}} \sim \mathbb{R}/2\pi\mathbb{Z}$. The case $G = \mathbb{Z}^Q$ follows by tensorization. The $(\varphi_\omega)_\omega$ are “orthogonal” in the sense that $\langle \varphi_\omega, \varphi_{\omega'} \rangle_{\mathbb{Z}} = \delta(\omega - \omega')$ can be understood as a Dirac kernel (this is similar to the Poisson formula), where $\langle u, v \rangle_{\mathbb{Z}} \stackrel{\text{def.}}{=} \sum_n u_n \bar{v}_n$. The “decomposition” of a sequence $(c_n)_{n \in \mathbb{Z}}$ on the set of characters is equivalent to forming a Fourier series $\sum_n c_n e^{-in\omega}$.

Similarly, for $G = \mathbb{R}/2\pi\mathbb{Z}$, one has $\hat{G} = \mathbb{Z}$, with orthonormal characters $\varphi_n = e^{in\omega}$, so that the decomposition of functions in $L^2(G)$ is the computation of Fourier coefficients.

Non-commutative groups. For non-commutative group, one also observes that G is not isometric to \hat{G} . A typical example is the symmetric group Σ_N of N elements, where one can show that $\hat{G} = \{\text{Id}, \varepsilon\}$ where $\varepsilon(\sigma) = (-1)^q$ is the signature, where q is the number of permutations involved in a decomposition of $\sigma \in \Sigma_N$.

In order to study non-commutative groups, one has to replace morphisms $\chi : G \rightarrow \mathbb{C}^*$ by morphisms $\rho : G \rightarrow \text{GL}(\mathbb{C}^{n_\rho})$ for some n_ρ , which are called “representations” of the group G . For $(g, g') \in G$ (denoting now multiplicatively the operation on G), one should thus have $\rho(gg') = \rho(g) \circ \rho(g')$. When $n_\rho = 1$,

identifying $\mathrm{GL}(\mathbb{C}) \sim \mathbb{C}^*$, one retrieve the definition of characters. Note that if ρ is a representation, then $\chi(g) \stackrel{\text{def.}}{=} \mathrm{tr}(\rho(g))$, where tr is the trace, defines a character.

If there is a subspace V stable by all the $\rho(g)$, one can build W such that $\mathbb{R}^{n_\rho} = V + W$ which is also stable, thus reducing the study of ρ to the study on a small space (matrices have a block diagonal form). It suffice to consider the inner product

$$\langle x, y \rangle \stackrel{\text{def.}}{=} \sum_{g \in G} \langle \rho(g)x, \rho(g)y \rangle_{\mathbb{R}^{n_\rho}}$$

and select the orthogonal V^\perp for this product. Note that when using an ortho-basis of \mathbb{R}^{n_ρ} for this inner product, the matrices associated to the $\rho(g)$ are unitary. In order to limit the set of such representations, one is only interested in “elementary” ones, which does not have invariant sub-spaces, and are called “irreducible” (otherwise one could create arbitrary large representation by stacking others in a block diagonal way). One also only consider these irreducible representation up to isomorphism, where two representation (ρ, ρ') are said to be isomorphic if $\rho(g) = U^{-1}\rho(g)U$ where $U \in \mathrm{GL}(\mathbb{C}^d)$ is a change of basis. The set of all these representations up to isomorphisms is called the dual group and denoted \hat{G} .

For the symmetric group, there is an explicit description of the set of irreducible representations. For instance, for $G = \Sigma_3$, $|G| = 6$, and there is two representation of dimension 1 (the identity and the signature, which are the caracters) and one representation of dimension 2, which is obtained by identifying G with the isometric of the equilateral triangle in \mathbb{R}^2 , and the dimensions indeed satisfy $6 = |G| = \sum n_\rho^2 = 1 + 1 + 2^2$.

One can show that the dimensions n_ρ of these irreducible representations $\rho \in \hat{G}$ satisfies $\sum_{\rho \in \hat{G}} n_\rho^2 = N$ and that the entries of the matrices involved in these representation define an orthogonal basis of the space of functions $f : G \rightarrow \mathbb{C}$ (note however that this set of basis function is not canonical since it depends on a particular choice of basis for each representation up to isomorphism). The associated Fourier transform of a function $f : G \rightarrow \mathbb{C}$ is defined as

$$\forall \rho \in \hat{G}, \quad \hat{f}(\rho) \stackrel{\text{def.}}{=} \sum_{g \in G} f(g)\rho(g) \in \mathbb{C}^{n_\rho \times n_\rho}.$$

This corresponds to computing inner products with the aforementioned ortho-basis. It is an invertible linear transform, whose invert is given next.

Proposition 8. *One has*

$$\forall g \in G, \quad f(g) = \frac{1}{|G|} \sum_{\rho \in \hat{G}} n_\rho \mathrm{tr}(\hat{f}(\rho)\rho(g^{-1})).$$

Proof. The proof relies on the following formula, which states that for any $g \in G$

$$A_g \stackrel{\text{def.}}{=} \sum_{\rho \in \hat{G}} n_\rho \mathrm{tr}(\rho(g)) = \delta_g$$

where $\delta_g = 0$ for $g \neq \mathrm{Id}_G$ and $\delta_{\mathrm{Id}_G} = 1$. We will not prove this formula, and refer to the book of Diaconis p.13 for a proof, which is based on the decomposition of the character of the so-called standard representation (which corresponds to permuting by the action of G the canonical basis of $\mathbb{R}^{|G|}$) on the set of characters, which are orthogonal. One then has

$$\begin{aligned} \frac{1}{|G|} \sum_{\rho \in \hat{G}} n_\rho \mathrm{tr}(\hat{f}(\rho)\rho(g^{-1})) &= \frac{1}{|G|} \sum_{\rho \in \hat{G}} n_\rho \mathrm{tr}\left(\sum_u f(u)\rho(u)\rho(g^{-1})\right) \\ &= \sum_u f(u) \frac{1}{|G|} \sum_{\rho \in \hat{G}} n_\rho \mathrm{tr}(\rho(ug^{-1})) = \sum_u f(u)G_{ug^{-1}} = f(g). \end{aligned}$$

□

One can define the convolution of two functions $f, h : G \rightarrow \mathbb{C}$ as

$$(f \star h)(a) \stackrel{\text{def.}}{=} \sum_{bc=a} f(b)h(c)$$

(beware that it is not commutative anymore). The Fourier transform diagonalize these convolution operators, has stated next.

Proposition 9. Denoting $\mathcal{F}(f) = \hat{f}$

$$\mathcal{F}(f \star h)(\rho) = \hat{f}(\rho) \times \hat{h}(\rho)$$

where \times is the matrix multiplication.

Proof.

$$\mathcal{F}(f \star h)(\rho) = \sum_x \sum_y f(y)h(y^{-1}x)\rho(yy^{-1}x) = \sum_x \sum_y f(y)\rho(y)h(y^{-1}x)\rho(y^{-1}x) = \sum_y f(y)\rho(y) \sum_z h(z)\rho(z)$$

where we made the change of variable $z = y^{-1}x$. \square

For certain groups, there exists fast Fourier transforms to compute \hat{f} , an example being the permutation group, but the structure of these algorithm is much more involved than in the case $G = \mathbb{Z}/N\mathbb{Z}$.

This theory extends to compact group by considering a discrete but infinite set of representations. A typical example is to analyze signals defined on the rotation groups $\mathrm{SO}(3)$, on which one can compute explicitly the representation using the basis of spherical harmonics detailed in Section ?? bellow. In this case, one has a representation of dimension $2\ell + 1$ for each frequency index ℓ .

2.8 A Bit of Spectral Theory

In order to define Fourier methods on general domains \mathbb{X} , one can use the aforementioned group-theoretic approach if $\mathbb{X} = G$ is a group, or also if a group acts transitively on \mathbb{X} . An alternative way is to describe the equivalent of Fourier basis functions as diagonalizing a specific differential operator (as we have seen in Section ?? that it is in some sense a way to characterise the Fourier basis). Of particular interest is the Laplacian, since it is the lowest order rotation-invariant differential operator, and that there exists natural generalization on domains such as surfaces or graphs.

2.8.1 On a Surface or a Manifold

The presentation here is very informal. One can define the Laplacian of a smooth function $f : \mathbb{X} \rightarrow \mathbb{C}$ defined on a “surface” \mathbb{X} as

$$\forall x \in \mathbb{X}, \quad (\Delta f)(x) \stackrel{\text{def.}}{=} \lim_{\varepsilon \rightarrow 0} \frac{1}{\mathrm{Vol}(B_\varepsilon(x))} \int_{B_\varepsilon(x)} f(x) d\mu(x) - f(x).$$

Here $\mu(x)$ is the area measure on \mathbb{X} , $\mathrm{Vol}(B) \stackrel{\text{def.}}{=} \int_B d\mu(x)$, and $B_\varepsilon(x) = \{y ; d_{\mathbb{X}}(x, y) \leq \varepsilon\}$ is the geodesic ball of radius ε at x , where $d_{\mathbb{X}}$ is the geodesic distance on \mathbb{X} (length of the shortest path).

If the surface \mathbb{X} is smooth, compact and connected, then it is possible to show that Δ is itself a compact operator with a negative spectrum $0 > \lambda_1 > \lambda_2 > \dots$ and an orthogonal set of eigenvectors $(\varphi_n)_{n \geq 0}$ where $\varphi_1 = 1$. Here the inner product is $\langle f, g \rangle_{\mathbb{X}} \stackrel{\text{def.}}{=} \int_{\mathbb{X}} f(x)g(x) d\mu(x)$ on $L^2(\mathbb{X})$. In the case of a flat torus $\mathbb{X} = (\mathbb{R}/\mathbb{Z})^d$, then writing $x = (x_1, \dots, x_d)$,

$$\Delta f = \sum_{s=1}^d \frac{\partial^2 f}{\partial^2 x_s}.$$

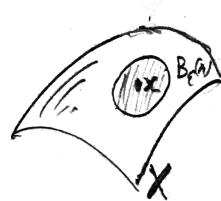


Figure 2.19: Computing Laplacian on a surface

Similarly to (??) (which was for an unbounded domain), then one can chose for this eigen-functions φ_n the Fourier basis (??) and $\lambda_n = -\|n\|^2$

2.8.2 Spherical Harmonics

Of particular interest is the special case of the previous construction on the $(d - 1)$ -dimensional sphere $\mathbf{S}^{d-1} = \{x \in \mathbb{R}^d ; \|x\|_{\mathbb{R}^d} = 1\}$. In this case, there exists a closed form expression for the eigenvectors of the Laplacian. In the 3-D case $d = 3$, they are indexed by $n = (\ell, m)$

$$\forall \ell \in \mathbb{N}, \quad \forall m = -\ell, \dots, \ell, \quad \varphi_{\ell,m}(\theta, \varphi) = e^{im\varphi} P_{\ell}^m(\cos(\theta))$$

and then the eigenvalue of the Laplacian is $\lambda_{\ell,m} = -\ell(\ell + 1)$. Here P_{ℓ}^m are associated Legendre polynomials, and we used spherical coordinates $x = (\cos(\varphi), \sin(\varphi) \sin(\theta), \sin(\varphi) \cos(\theta)) \in \mathbf{S}^3$ for $(\theta, \varphi) \in [0, \pi] \times [0, 2\pi]$. The index ℓ is analogous to the amplitude of Fourier frequencies in 2-D. For a fixed ℓ , the space $V_{\ell} = \text{span}(\varphi_{\ell,m})$ is an eigenspace of Δ , and is also invariant under rotation.



Figure 2.20: Spherical coordinates.

2.8.3 On a Graph

We assume \mathbb{X} is a graph of N vertices, simply indexed $\{1, \dots, N\}$. Its “geometry” is described by a connectivity matrix of weights $W = (w_{i,j})_{i \sim j}$ where we denote $i \sim j$ to indicate that (i, j) is an edge of the graph for $(i, j) \in \mathbb{X}^2$. We assume that this weight matrix and the connectivity is symmetric, $w_{i,j} = w_{j,i}$.

The graph Laplacian $\Delta : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is computing the difference between the average of values around a point and the value at this point

$$\forall f \in \mathbb{R}^N, \quad (\Delta f)_i \stackrel{\text{def.}}{=} \sum_{j \sim i} w_{i,j} f_j - \left(\sum_{j \sim i} w_{i,j} \right) f_i \quad \Rightarrow \quad \Delta = W - D$$

where $D \stackrel{\text{def.}}{=} \text{diag}_i(\sum_{j \sim i} w_{i,j})$. In particular, note $\Delta \mathbf{1} = 0$

For instance, if $\mathbb{X} = \mathbb{Z}/N\mathbb{Z}$ with the graph $i \sim i-1$ and $i \sim i+1$ (modulo N), then Δ is the finite difference Laplacian operator $\Delta = D_2$ defined in (??). This extends to any dimension by tensorization.

Proposition 10. Denoting $G : f \in \mathbb{R}^N \mapsto (\sqrt{w_{i,j}}(f_i - f_j))_{i < j}$ the graph-gradient operator, one verifies that

$$-\Delta = G^T G \quad \Rightarrow \quad \forall f \in \mathbb{R}^N, \quad \langle \Delta f, f \rangle_{\mathbb{R}^N} = -\langle Gf, Gf \rangle_{\mathbb{R}^P}.$$

where P is the number of (ordered) edges $E = \{(i, j) ; i \sim j, i < j\}$.

Proof. One has

$$\begin{aligned} \|Gf\|^2 &= \sum_{(i,j) \in E} w_{i,j} |f_i - f_j|^2 = \sum_{i < j} w_{i,j} f_i^2 + \sum_{i < j} w_{i,j} f_j^2 - 2 \sum_{i < j} w_{i,j} f_i f_j \\ &= \sum_{i < j} w_{i,j} f_i^2 + \sum_{i > j} w_{i,j} f_i^2 - \sum_{i,j} w_{i,j} f_i f_j = \sum_j f_i^2 \sum_{i,j} w_{i,j} - \sum_i f_i \sum_j w_{i,j} f_j \\ &= \langle Df, f \rangle - \langle Lf, f \rangle = -\langle Lf, f \rangle. \end{aligned}$$

□

Figure 2.21: Weighted graph.



This proposition shows that Δ is a negative semi-definite operator, which thus diagonalizes in an ortho-basis $(\varphi_n)_{n=1}^N$, with $\varphi_1 = 1$, with eigenvalues $0 \geq \lambda_1 \geq \lambda_N$. If \mathbb{X} is connected, one can show that $\lambda_1 < 0$. In the case of a regular graph associated to a uniform grid, one retrieves the discrete Fourier basis (??).

More details and application of Laplacians on graphs can be found in Chapter ??, see in particular Section ??.

Bibliography

- [1] P. Alliez and C. Gotsman. Recent advances in compression of 3d meshes. In N. A. Dodgson, M. S. Floater, and M. A. Sabin, editors, *Advances in multiresolution for geometric modelling*, pages 3–26. Springer Verlag, 2005.
- [2] P. Alliez, G. Ucelli, C. Gotsman, and M. Attene. Recent advances in remeshing of surfaces. In *AIM@SHAPE report*. 2005.
- [3] Amir Beck. *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB*. SIAM, 2014.
- [4] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.
- [5] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [6] E. Candès and D. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities. *Commun. on Pure and Appl. Math.*, 57(2):219–266, 2004.
- [7] E. J. Candès, L. Demanet, D. L. Donoho, and L. Ying. Fast discrete curvelet transforms. *SIAM Multiscale Modeling and Simulation*, 5:861–899, 2005.
- [8] A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20:89–97, 2004.
- [9] Antonin Chambolle, Vicent Caselles, Daniel Cremers, Matteo Novaga, and Thomas Pock. An introduction to total variation for image analysis. *Theoretical foundations and numerical methods for sparse recovery*, 9(263-340):227, 2010.
- [10] Antonin Chambolle and Thomas Pock. An introduction to continuous optimization for imaging. *Acta Numerica*, 25:161–319, 2016.
- [11] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1999.
- [12] F. R. K. Chung. Spectral graph theory. *Regional Conference Series in Mathematics, American Mathematical Society*, 92:1–212, 1997.
- [13] Philippe G Ciarlet. Introduction à l’analyse numérique matricielle et à l’optimisation. 1982.
- [14] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *SIAM Multiscale Modeling and Simulation*, 4(4), 2005.
- [15] P. Schroeder et al. D. Zorin. Subdivision surfaces in character animation. In *Course notes at SIGGRAPH 2000*, July 2000.

- [16] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. on Pure and Appl. Math.*, 57:1413–1541, 2004.
- [17] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, 4(3):245–267, 1998.
- [18] D. Donoho and I. Johnstone. Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, 81:425–455, Dec 1994.
- [19] Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.
- [20] M. Figueiredo and R. Nowak. An EM Algorithm for Wavelet-Based Image Restoration. *IEEE Trans. Image Proc.*, 12(8):906–916, 2003.
- [21] M. S. Floater and K. Hormann. Surface parameterization: a tutorial and survey. In N. A. Dodgson, M. S. Floater, and M. A. Sabin, editors, *Advances in multiresolution for geometric modelling*, pages 157–186. Springer Verlag, 2005.
- [22] Simon Foucart and Holger Rauhut. *A mathematical introduction to compressive sensing*, volume 1. Birkhäuser Basel, 2013.
- [23] I. Guskov, W. Sweldens, and P. Schröder. Multiresolution signal processing for meshes. In Alyn Rockwood, editor, *Proceedings of the Conference on Computer Graphics (Siggraph99)*, pages 325–334. ACM Press, August 8–13 1999.
- [24] A. Khodakovsky, P. Schröder, and W. Sweldens. Progressive geometry compression. In *Proceedings of the Computer Graphics Conference 2000 (SIGGRAPH-00)*, pages 271–278, New York, July 23–28 2000. ACM Press.
- [25] L. Kobbelt. $\sqrt{3}$ subdivision. In Sheila Hoffmeyer, editor, *Proc. of SIGGRAPH'00*, pages 103–112, New York, July 23–28 2000. ACM Press.
- [26] M. Lounsbery, T. D. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Trans. Graph.*, 16(1):34–73, 1997.
- [27] S. Mallat. *A Wavelet Tour of Signal Processing, 3rd edition*. Academic Press, San Diego, 2009.
- [28] Stephane Mallat. *A wavelet tour of signal processing: the sparse way*. Academic press, 2008.
- [29] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Commun. on Pure and Appl. Math.*, 42:577–685, 1989.
- [30] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.
- [31] Gabriel Peyré. *L’algèbre discrète de la transformée de Fourier*. Ellipses, 2004.
- [32] J. Portilla, V. Strela, M.J. Wainwright, and Simoncelli E.P. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Trans. Image Proc.*, 12(11):1338–1351, November 2003.
- [33] E. Praun and H. Hoppe. Spherical parametrization and remeshing. *ACM Transactions on Graphics*, 22(3):340–349, July 2003.
- [34] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992.
- [35] Otmar Scherzer, Markus Grasmair, Harald Grossauer, Markus Haltmeier, Frank Lenzen, and L Sirovich. *Variational methods in imaging*. Springer, 2009.

- [36] P. Schröder and W. Sweldens. Spherical Wavelets: Efficiently Representing Functions on the Sphere. In *Proc. of SIGGRAPH 95*, pages 161–172, 1995.
- [37] P. Schröder and W. Sweldens. Spherical wavelets: Texture processing. In P. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95*. Springer Verlag, Wien, New York, August 1995.
- [38] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.
- [39] A. Sheffer, E. Praun, and K. Rose. Mesh parameterization methods and their applications. *Found. Trends. Comput. Graph. Vis.*, 2(2):105–171, 2006.
- [40] Jean-Luc Starck, Fionn Murtagh, and Jalal Fadili. *Sparse image and signal processing: Wavelets and related geometric multiscale analysis*. Cambridge university press, 2015.
- [41] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Applied and Computation Harmonic Analysis*, 3(2):186–200, 1996.
- [42] W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546, 1997.