A course on signal processing

Fernando Chamizo

Universidad Autónoma de Madrid
and
Instituto de Ciencias Matemáticas

Contents

What is this course about?

This course is primarily about signals. And what is a signal? And what is the need for processing? A signal is a function carrying some kind of information. This is too general and any fanatic mathematician could complain saying that any function carries information, the one about itself. There is not a more concrete mathematical definition because the term signal is closer to engineering. If you prefer something in the middle (physics?) you can stick to the following definition grabbed from [MI11]: "*For our purposes a* signal *is defined as any physical quantity that varies as a function of time, space, or any other variable or variables. Signals convey information in their patterns of variation*". To add something on my own, in many instances a signal is a function that it is used to transmit something informative from a place into another, for instance the amplitude of the human voice in terms of time or the frames of a video. Signal processing concerns the manipulation of the information carried by the signal. In general this is not a devious manipulation, we look for improving quality or gaining storage efficiency. Think for instance in a CD, usually it contains (contained?) up to 80 minutes of audio. The signal giving the sound, a longitudinal wave of pressure transmitted through air has to be treated to fit in 700 megas, it means around $5.87 \cdot 10^9$ bits, zeros and ones, represented by tiny pits in a plastic layer. Something must be done to convert the complicated wave sound into the individually simple zeros and ones.

My primary intention when I started to think about this course was to elaborate material close to the applications for students with an education biased to pure mathematics. I shall be utterly satisfied if some mathematicians feel that they learned something on engineering or if some engineers feel that they learned something on mathematics. I have tried to keep low prerequisites, for instance the basic Fourier analysis is explained from scratch. By personal taste, I have focused especially on digital signals. I encourage the reader to revise the complementary material mentioned in the final part. Although it is not very polished, I think that it can be useful to develop many teaching activities. Some experiments mentioned in the sections "Experience mathematics by yourself" are indebted to [TG15], [MVG16] and `http://bohr.inf.um.es/miembros/rgm/`.

I am completely aware about my limited English proficiency and I want to apologize sincerely and deeply for it.

It is my pleasure to acknowledge the help provided by the student Carlos Manada during the second semester of the course 2017/2018 pointing out misprints and making sharp comments on the notes that were the starting point for this document.

Madrid, April 2020

Fernando Chamizo

Glossary of notation

This is a brief list of some notations employed in this course trying to respect loosely the alphabetical order.

$\langle a, b\rangle$	Inner product.
C	Continuous functions (real variable functions if it is no otherwise indicated).
C^k	Functions with a continuous k derivative.
C_0^k	Compactly supported functions with a continuous k derivative.
DCT	Discrete Cosine Transform (II).
DFT	Discrete Fourier Transform.
DWT	Discrete Wavelet Transform.
D_N	Dirichlet kernel.
Δ	Laplace operator.
$\delta(x)$	Dirac delta.
$\delta_P(x)$	Periodic Dirac delta.
$e(x)$	$e^{2\pi i x}$.
L^p	Functions with $\lvert f\rvert^p$ Lebesgue integrable.
ℓ_C	Average length of the code C.
$\ell(s)$	Length of the bit string s.
$\ell^2(\mathbb{Z})$	Sequences $\{a_n\}_{n\in\mathbb{Z}}$ with $\sum \lvert a_n\rvert^2 < \infty$.
MAD	Median absolute deviation.
$O(f)$	A function F with $\limsup \lvert f\rvert/\lvert F\rvert < \infty$.
$\operatorname{sinc}(x)$	The cardinal sinus $\sin(\pi x)/(\pi x)$.
$\mathbb{T}$	Unit 1D torus (S^1).
Tr	Trace.
$\mathbb{Z}_N$	Integers modulo N.
$[A, B]$	The commutator $AB - BA$.
$\widehat{f}$	Fourier transform of f.
$\widehat{G}$	Character group of G.
$\widehat{x}_n$	Discrete Fourier coefficient.
$\lVert f\rVert_p$	The p-norm $\big(\int \lvert f\rvert^p\big)^{1/p}$.
$\lVert f\rVert_\infty$	The essential supremum of f.
$\lfloor x\rfloor$	Integral part of x.
$\nabla\cdot$	Divergence.
$\nabla\times$	Curl.
$\otimes$	Kronecker product.

Chapter 1

Simple waves

1.1 Physical principles

1.1.1 Harmonic oscillators

In several physical situations when a particle in the real line is displaced to $x = x(t)$ it suffers an attractive force towards its equilibrium position at the origin that is proportional to x. A couple of academic examples in which this model gives a reasonably good approximation are a particle attached to a spring and the small oscillations of a simple pendulum. Recalling $F = ma$ the differential equation of motion is

$$x'' + \omega^2 x = 0 \tag{1.1}$$

where $-\omega^2$ is the proportionality constant ($\omega > 0$). A physical system driven by this equation is called a *harmonic oscillator*. The minus sign imposes that the force is attractive. Physicists very often prefer to write $\ddot{x}$ instead of x'' but we advocate here the mathematical notation. You know how to solve this ODE (ordinary differential equation) or you already know by heart the general solution

$$x(t) = A\cos(\omega t) + B\sin(\omega t) \tag{1.2}$$

where A and B are constants to be adjusted according to the initial conditions. It is said that ω is the *angular frequency*, the number of radians per unit of time. In daily life it is more common to express the *frequency* ν as the number of repetitions of an event per unit of time, so $\omega = 2\pi\nu$. The *hertz*, abbreviated Hz, is a natural unit of frequency meaning one cycle per second. For instance $120\,rpm$ (revolutions per minute) correspond to $\nu = 2\,Hz$ and to $\omega = 4\pi\,rad/s$.

In the examples mentioned above (spring and pendulum) in practice we do not see endless oscillations like (1.2). It may be a good approximation for short periods of time but without external help the friction eventually stops the oscillations. For a fluid like air, the friction is with some approximation proportional to the velocity (*Stokes' law*). It suggests to improve the model (1.1) to get the *damped harmonic oscillator* ruled by

$$x'' + 2ax' + \omega^2 x = 0 \tag{1.3}$$

where $2a$ with $a > 0$ is the new proportionality constant. You should know also how to solve this equation. The fancy and usual way is to try a solution $x(t) = e^{rt}$ to conclude that $r \in \{r_-, r_+\}$ with $r_\pm = -a \pm \sqrt{a^2 - \omega^2}$. If the friction is not very large (for your curiosity this is called the *underdamped harmonic oscillator*, the only one we are going to consider), $a < \omega$ and hence $r_\pm$ are complex numbers. When we choose A and B in $Ae^{r_- t} + Be^{r_+ t}$ to match the (real) initial conditions, the imaginary parts must cancel and the general solution becomes

$$(1.4) \qquad x(t) = Ae^{-at}\cos(\widetilde{\omega}t) + Be^{-at}\sin(\widetilde{\omega}t) \qquad \text{with} \quad \widetilde{\omega} = \sqrt{\omega^2 - a^2}.$$

The *amplitudes* (the coefficients of the oscillatory terms) decay exponentially in time, as seen in practice, however if a is much smaller than ω we have that the relative frequency shift $(\omega - \widetilde{\omega})/\omega$ is approximately as small as $\frac{1}{2}(a/\omega)^2$. This approximate invariance of the frequency is in part responsible for the precision of the pendulum clocks that caused a revolution in time measurement accuracy in the 17th century. I know, I know, (1.4) is far from modeling the oscillations of the pendulum of a clock because of the exponential decay. One needs an external force to maintain the oscillations. In those clocks it was provided by the *anchor escapement* [BB05, §10.2.4]. We have learned in previous courses and we shall recall once again in this one, that all respectable functions can be expressed in terms of sines and cosines then we restrict ourselves to external forces of the form $F_e\cos(\omega_e t - \varphi_e)$ with $F_e, \omega_e > 0$. Note that for $\varphi_e = 0$ we have a pure cosine and for $\varphi_e = \pi/2$ a pure sine. The corresponding *driven harmonic oscillator* is consequently modeled by the equation

$$(1.5) \qquad x'' + 2ax' + \omega^2 x = F_e\cos(\omega_e t - \varphi_e).$$

The solution is the sum of the solution of the homogeneous equation (1.3), given by (1.4), and a particular solution that could be computed using variation of the parameters, however the calculations become complicated because the final result depends on a, ω, ω_e and φ_e in a messy way. Let us use again a complex variable trick considering the complex differential equation

$$(1.6) \qquad y'' + 2ay' + \omega^2 y = Fe^{i\omega_e t} \qquad \text{with} \quad F = F_e e^{-i\varphi_e}.$$

If we find a particular solution y_p then, taking real parts, $x_p = \Re y_p$ will be a particular solution of (1.5). Let us try $y_p = Ge^{i\omega_e t}$ that forces $G = F/(-\omega_e^2 + 2ia\omega_e + \omega^2)$. If we write this as $Ce^{-i\varphi}$ with $C = |G|$, then

$$(1.7) \qquad x_p(t) = C\cos(\omega_e t - \varphi) \qquad \text{with} \quad C = \frac{F_e}{\sqrt{(\omega^2 - \omega_e^2)^2 + 4a^2\omega_e^2}}.$$

This is the *steady-state solution* of (1.5), the limit when $t \to +\infty$ of any solution under any initial conditions, because (1.4) goes to 0. The denominator in C measures the *gain*. If $(\omega^2 - \omega_e^2)^2 + 4a^2\omega_e^2$ is small then $C > F_e$, the external force is somewhat amplified. Probably everybody has experienced this phenomenon in children playgrounds: if one pushes periodically a swing at the right moment ($\omega \approx \omega_e$) then the amplitude of the oscillation increases. In this case we say that *resonance* has occurred.

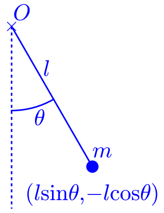
The extreme mathematical resonance case in which ω is exactly ω_e and $a \to 0$ (no friction) causes a problem because $x_p \to \infty$ in (1.7). Do not worry, the swing is not going to explode or something like that if you are close to this situation. In this extreme case of resonance, our ansatz for the form of y_p is wrong. A valid particular solution involves a factor t that remains bounded at any finite value of t, although the amplitude of any solution increases arbitrarily when $t \to \infty$. Recall Tacoma Narrows Bridge! (or look up on the internet if you do not know what I am talking about).

In part, the abstract study of the harmonic oscillator is motivated by the pendulum which does not follow exactly (1.1). Instead of deriving the right equation using the ubiquitous Newton's second law $F = ma$, we are going to employ the easier Lagrangian formulation. If you do not know anything about *Lagrangian mechanics*, I hope this example to provide an incentive to read something about it. The main asset is that, like in differential geometry, you can choose the coordinates $\{q_k\}$ you wish. Once you have done it, you have to construct with them the *Lagrangian* $L = T - V$ where T is the kinetic energy and V is the potential energy. It is a function of $\{q_k\}$ and $\{\dot{q}_k\}$ where we have suspended temporarily our preference of q_k' instead of $\dot{q}_k$ to indicate a time derivative. The Lagrangian depends implicitly on t, through $q_k = q_k(t)$ and may also depend explicitly on it. The law of motion in the chosen coordinates is given by the *Euler-Lagrange equations*

$$\frac{d}{dt}\Big(\frac{\partial L}{\partial \dot{q}_k}\Big) = \frac{\partial L}{\partial q_k}. \tag{1.8}$$

For a system of particles of masses m_j at $\mathbf{x}_j$ under gravitational forces near Earth's surface, the kinetic energy is $\frac{1}{2}\sum m_j\|\mathbf{x}_j\|^2$ and the potential energy is $\sum m_j z_j$, where g is the gravitational acceleration. The Lagrangian is got writing these quantities in our coordinates. For a pendulum (a point mass m suspended from the origin through a massless rigid rod of length l) the natural coordinate is the angle θ and we have

$$\begin{cases} L(\theta,\dot{\theta}) &= \frac{1}{2}m\big((l\dot{\theta}\cos\theta)^2 + (l\dot{\theta}\sin\theta)^2\big) + mgl\cos\theta \\ &= \frac{1}{2}ml^2\dot{\theta}^2 + mgl\cos\theta \\ \\ & \frac{d}{dt}\Big(\frac{\partial L}{\partial \dot{\theta}}\Big) = ml^2\ddot{\theta}, \quad \frac{\partial L}{\partial \theta} = -mgl\sin\theta. \end{cases} \tag{1.9}$$



And *voilà*, in the blink of an eye we have the pendulum equation without using fictitious invisible forces like the tension of the rod that puzzled you in Physics 101,

$$\ddot{\theta} + \frac{g}{l}\sin\theta = 0. \tag{1.10}$$

When θ is small, (1.1) with $\omega^2 = \sqrt{g/l}$ approximates (1.10) giving the formula $2\pi\sqrt{l/g}$ for the period of oscillation. If you are not happy with the approximations, multiply by $\dot{\theta}$ and integrate to get for a certain constant E (related to the energy), under $\theta(0) = 0$,

$$t\sqrt{\frac{g}{l}} = \int_0^\theta \frac{d\theta}{\sqrt{2(E+\cos\theta)}} \quad \underset{ku=\sin(\theta/2)}{\overset{E=2k^2-1}{\Longleftrightarrow}} \quad t\sqrt{\frac{g}{l}} = \int_0^u \frac{du}{\sqrt{(1-u^2)(1-k^2u^2)}}. \tag{1.11}$$

The last integral is the famous Jacobi elliptic integral enjoying wonderful properties. The theory assures that $u = u(t)$ extends to a meromorphic function with two periods. This is cumbersome from the mathematical point of view but, following [Bae05], there is a convincing physical easy explanation: The pendulum is the epitome of boring repetitive oscillations, so you have a real period there. Changing in (1.10) $t \mapsto it$ the equation stands except for changing g by $-g$, but reversing the direction of gravity is like putting the pendulum upside down and it is still a pendulum, so we have also a complex period.

The patient reader may forgive or skip a final brief physical aside about non classical oscillations.

The currently official physical explanation of reality, quantum field theory, postulates that there is quantum harmonic oscillator at each point of vacuum. In certain units and with a criminal notation (t is now position and x^2 probability density) the quantum harmonic oscillator is ruled by a nontrivial solution $x = x(t)$ of

$$x'' + (2\omega - t^2)x = 0. \tag{1.12}$$

This looks as a kind of harmonic oscillator (1.1) with frequency changing on time. If we look for solutions $x = x(t)$ smooth square-integrable and not identically zero (as dictated by quantum mechanics), it can be proved that they exist if and only if 2ω is a positive odd integer. This lies more or less deep (not an exercise!) and physically indicates a quantization of the energy. The smallest value $\omega = 1/2$ corresponds to the solution $x(t) = Ae^{-t^2/2}$ that does not oscillate at all and the same happens for higher values of ω. Are not you intrigued by the name harmonic "oscillator"? Good! You have a challenging reason to enter into the exciting realm of quantum mechanics but this is not the right course ([Zwi13] is a good one). A last aside of the aside is that $1/2 \neq 0$ causes something awkward: after the postulate of quantum field theory, each point carries a positive energy and there are infinitely many points, then the energy of the vacuum, that has to be minimal, is infinite!

Suggested Readings. The different flavors of classic harmonic oscillations are discussed in almost any physics book for undergraduates (for instance [AF67]). In [Sim17] you can learn about the methods for solving differential equations with an eye to applications. A quick and mathematically spotless discussion of the quantum harmonic oscillator is in §3.4 of [Fol08]. The original book [BB05] is an accessible, comprehensive and historical study about oscillations surrounding pendulums or taking them as motivation.

1.1.2 Electromagnetic waves and circuits

A substantial part of the information that reaches us employs electromagnetic waves to travel, at least in a part of its trip from the source. Even DTT (Digital Terrestrial Television) contradicts its "terrestrial" qualifier making use of conventional television antennas. I must confess that the study of electromagnetic waves is unrelated the rest of the course but I consider this subsection as general knowledge for graduate students in mathematics.

In the beginning... it was "*A treatise on electricity and magnetism*" published in 1873 and authored by J.C. Maxwell [Max54]. OK, it was not the beginning, it was rather a culmination. A way of summarizing in a mathematical compact form, called *Maxwell equations*, experiments and laws stated by several researchers. One of the main contributors was M. Faraday and he probably would have freaked out (he died 6 years before the publication of the treatise) because he was reluctant to use mathematical arguments while the classic form of Maxwell equations in vacuum is

$$\nabla \cdot \vec{E} = 0, \qquad \nabla \cdot \vec{B} = 0, \qquad \nabla \times \vec{E} = -\frac{1}{c}\frac{\partial \vec{B}}{\partial t}, \qquad \nabla \times \vec{B} = \frac{1}{c}\frac{\partial \vec{E}}{\partial t} \tag{1.13}$$

where c is the speed of light and, as usual, $\nabla\cdot$ and $\nabla\times$ are the divergence and the curl. The main characters are the *electric field* $\vec{E}$ and the *magnetic field* $\vec{B}$. Roughly speaking they are a way of measuring the strength of the forces produced by charges and magnets.

It is hard to believe that (1.13) are "experimental formulas". How on earth can you measure the curl? The answer is that after applying Stokes' theorem they become integral formulas harder to manage from the mathematical point of view but more intuitive.

Let us focus on the third equation (the so-called Maxwell-Faraday equation). If we move a magnet through a wire loop enclosing a surface S, an electric current appears in the wire. This is the principle of the wind or water turbines that bring electricity to our homes. If the magnet stops there is no current and when the magnet moves quickly the current is greater. The strength of the magnet, represented by $\vec{B}$ also matters. In this context the following relation, with K a constant, sounds more or less natural

$$\frac{d}{dt}\int_S \vec{B} = K \int_{\partial S} \vec{E}. \tag{1.14}$$

The left hand side is something like the variation of the "total magnetic field" through the surface S and the right hand side is the total electric field through the wire. With standard (Gaussian) units $K = -c$. This is not essential, choosing other units we could give any nonzero value to K. The minus sign indicates an old convention about what is called north and south pole of a magnet. If we believe (1.14) with $K = -c$, by Stokes' theorem applied to the right side, we have

$$\int_S \Big(\frac{\partial \vec{B}}{\partial t} + c\nabla \times \vec{E}\Big) = 0. \tag{1.15}$$

As the surface S is arbitrary (as the wire loop is) we deduce the third equation of (1.13).

One may wonder (as Faraday would have done) why the mathematically abstruse formulation (1.13) is important. The answer is that mathematics is usually easier than real life (especially if you are a mathematician). For instance, in some way the existence of electromagnetic waves and even to some extent special relativity are encoded in (1.13) and Maxwell equations predicted them before any experimental test did (for the latter, check the title or the content of the celebrated 1905 paper by A. Einstein introducing relativity [Ein05]). Let us see where the waves are.

Imagine an evil professor posing the following problem to you as a freshman: *If $\vec{F}$ is a vector field with $\nabla \cdot \vec{F} = 0$, compute $\nabla \times (\nabla \times \vec{F})$.* In principle it is a calculation, but the formula for $\nabla \times \vec{F}$ is involved and consequently $\nabla \times (\nabla \times \vec{F})$ is expected to be super-involved. It turns out that the condition $\nabla \cdot \vec{F} = 0$ allows to simplify the mess to get

$$(1.16) \qquad \nabla \times (\nabla \times \vec{F}) = -\Delta \vec{F} \qquad \text{where} \quad \Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}.$$

Would you dare to get a proof of this as simple as you can?

Taking the curl of the third and fourth equations in (1.13), one deduces from (1.16)

$$(1.17) \qquad c^2 \Delta \vec{E} = \frac{\partial^2 \vec{E}}{\partial t^2} \qquad \text{and} \qquad c^2 \Delta \vec{B} = \frac{\partial^2 \vec{B}}{\partial t^2}.$$

It means that each coordinate of $\vec{E}$ and $\vec{B}$ is a solution of the wave equation for speed c

$$(1.18) \qquad c^2 \Delta u = \frac{\partial^2 u}{\partial t^2}.$$

We conclude that there are electromagnetic waves and they travel at speed c. These waves were mentioned firstly in 1865 in a work of Maxwell [Max65]. Around 1887, H. Hertz was able to produce electromagnetic waves with high voltage sparks and detect them some meters further [Her90]. The rest is history. Next time you switch on the TV, say loudly: Thank you Maxwell!

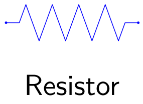
In many situations one has to deal with a non-vacuum environment, with charges. The extension of (1.13) to this case is

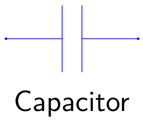
$$(1.19) \qquad \nabla \cdot \vec{E} = 4\pi\rho, \qquad \nabla \cdot \vec{B} = 0, \qquad \nabla \times \vec{E} = -\frac{1}{c}\frac{\partial \vec{B}}{\partial t}, \qquad \nabla \times \vec{B} = \frac{4\pi}{c}\vec{\jmath} + \frac{1}{c}\frac{\partial \vec{E}}{\partial t}$$

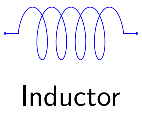
where ρ is the charge density and $\vec{\jmath} = \rho\vec{v}$ with $\vec{v}$ the velocity field (flow velocity) of the charges.

In practice, most of the signals are treated electronically then it does not harm to learn something about very basic components and circuits. Surely you have heard the names *voltage* (electric potential difference) and *electric current* referred to electricity. They are the line integral of $\vec{E}$ between two points and the flux of $\vec{\jmath}$. Fortunately the so-called hydraulic analogy gives an intuitive way of thinking about them. Basically, if you consider electricity as a fluid and the conductors in the circuits as pipes, the voltage between two points is the difference of pressure and the current is the volume flow rate.

Although we live in the era of silicon (I wrote silicon, not silicone), for the sake of brevity we are going to consider only passive components, meaning that they do not involve semiconductors. The most basic are the *resistor*, the *capacitor* and the *inductor*. Their symbols and their characteristic relations between the time-depending voltage $V = V(t)$ (within their terminals) and electric current $I = I(t)$ are

(1.20) 

$V = IR$ 

$I = CV'$ 

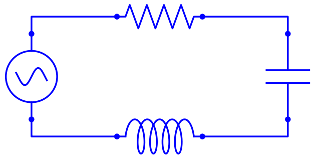
$V = LI'$.

Here the *resistance* R, the *capacitance* C and the *inductance* L are constants associated to the specifications of the particular component. As the symbols suggest, capacitors are essentially a pair of very close conducting plates and the inductor is a conducting spring, a coil (you can construct and test both by yourself [Fie03]). Believe or not, the equations of (1.20) in these two cases are quite direct consequences of Maxwell equations (1.19), as explained below. A resistor is made of materials that are not so good conductors and the theoretical explanation of its equation (the famous *Ohm's law*) belongs to solid state physics. In the hydraulic analogy a resistor is a constricted pipe. You can look up the analogs for capacitors and inductors that I do not mention here. For the interested reader (skip to the RLC circuit if you are not), let us see how to deduce their equations without rigor and without entering into details.

Integrating the first formula of (1.19) on the surface S of the cylinder determined by the plates of a capacitor, we have by the divergence theorem $\int_S \vec{E} = 4\pi Q$ where Q is the total charge on the plates. This suggests that $|\vec{E}|$ is proportional to Q. As the voltage is electric field times length, in this case the separation between the plates, the charge is proportional to the voltage. Denoting C the proportionality constant and using $I = dQ/dt$, the equation $I = CV'$ is deduced. For the inductor there is a little dirty trick. It turns out that for the frequencies appearing in electric circuits, the last term in the last formula of (1.19) is negligible[1]. Let us consider a one-loop inductor determined by a disk D with boundary C of length l. Integrating the reduced equation on S, by Stoke's theorem, $\int_C \vec{B} = \frac{4\pi}{c}\int_D \vec{\jmath}$. This suggests that $|\vec{B}|l$ and I are proportional. The variation of the flux of $\vec{B}$ in time gives the voltage thanks to Maxwell-Faraday equation (1.14). Then the voltage should be proportional to the variation of I, that is written $V = LI'$.

The simplest *RLC circuit* corresponds to the following scheme

(1.21)



$$V_R + V_C + V_L = V$$

where the rightmost symbol means a source of voltage V. The equation is an instance of *Kirchhoff's law*, like the conservation of energy, where V_R, V_C and V_L are the voltages between the terminals of each component. Taking the derivative and using (1.20),

$$RI' + C^{-1}I + LI'' = V'. \tag{1.22}$$

If our voltage source produce a usual sine wave $V = V_0\sin(\omega t)$ (as our power plugs at home), we have

$$I'' + RL^{-1}I' + (LC)^{-1}I = L^{-1}\omega V_0\cos(\omega t). \tag{1.23}$$

[1] In fact, this term was the only part of the equations not supported with experiments at Maxwell's time. He introduced it using purely theoretical arguments.

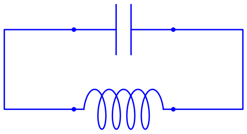
According to our previous study, the resonance happens for $\omega = \omega_0$ with $\omega_0 = (LC)^{-1/2}$ and at large, the current behaves as $I = I_0 \sin(\omega t - \delta)$ for some δ. With our coefficients

$$(1.24) \qquad V_0 = I_0 Z \qquad \text{where} \qquad Z = \sqrt{R^2 + L^2\omega^2(1-\omega_0^2/\omega^2)^2}.$$

If $\omega = \omega_0$ then $Z = R$ considering only the amplitudes (forgetting the phases) the circuit behaves as if the capacitor and the inductance does not furnish any resistance. On the other hand if ω is not close to ω_0, then Z is much bigger than R. This is the principle to tune a specific radio station or TV channel. In the mathematical context, this primitive machine that allows to select with certain precision specific frequencies is a gateway to an electronic computation of Fourier expansions.

Note that if you omit the source and the resistor in (1.21) the new equation is

$$(1.25) \qquad I'' + (LC)^{-1} I = 0$$



and we have in theory a tireless harmonic oscillator if the capacitor is initially charged. In practice, there is always some resistance in the conductors and, as in a free pendulum, the oscillations fade away quickly. To achieve a real electronic oscillator you have to introduce some kind of amplification. In the early days it was achieved with vacuum tubes (valves) and later with transistors.

Suggested Readings. For a basic mathematically oriented introduction to the Maxwell equations and its relation to modern theoretical physics, I recommend the recent book [Gar15]. The solution of the equations and its meaning is very well explained in the modern classic [FLS64].

1.1.3 Sound waves

The sound consists of changes of pressure that can be detected by the human ear. With some approximations and basic physics we are going to convince ourselves that it is transmitted as a wave.

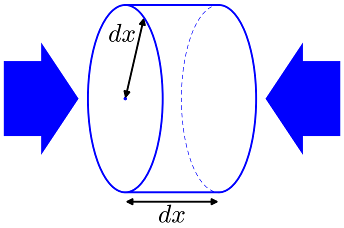
If a particle of the air is in a certain position we want to study its displacement u when time evolves and it is disturbed by the sound. We assume that the perturbation acts in the same way at every horizontal line, in other words, $u = u(x,t)$ and we can focus on the X axis. The changes in the pressure p are related to changes in the density ρ. For the sound there are not big variations with respect to the normal pressure and density, say p_0 and ρ_0, then we can write

$$(1.26) \qquad p(x,t) = p_0 + p_\epsilon(x,t) \qquad \text{and} \qquad \rho(x,t) = \rho_0 + \rho_\epsilon(x,t)$$

meaning that p_ϵ and ρ_ϵ are much smaller than the constants p_0 and ρ_0. They express some kind of perturbation.

It seems natural than p and ρ have to related in some way by an equation of state[2]. Whatever it is, everything is linear at small scales and then it is not far from the truth to assume $p_\epsilon = \kappa\rho_\epsilon$ for some constant $\kappa > 0$ that measures in some way the elasticity of the air.

We proceed using $F = ma$, as usual in very basic mechanic. The mass of a small cylinder of radius and height (along the X axis) of length dx is approximately $\pi\rho_0(dx)^3$. The forces to each side are $p(x,t)\pi(dx)^2$ and $-p(x+dx,t)\pi(dx)^2$ (recall pressure = force/surface).

(1.27) $F = p(x,t)\text{Area}$  $F = p(x+dx,t)\text{Area}.$

Dividing by $\pi(dx)^3$, the approximation of $F = ma$ becomes when $dx \to 0$

$$-\frac{\partial p_\epsilon}{\partial x} = \rho_0 \frac{\partial^2 u}{\partial t^2}. \tag{1.28}$$

On the other hand, when time evolves, by effect of the displacement from the undisturbed situation, the height of the cylinder pass from dx to $d(x+u)$. The mass must be preserved, then $\rho_0\pi(dx)^3 = \rho d(x+u)\pi(dx)^2$. As $\rho_0/\rho = 1 - \rho_\epsilon/\rho \sim 1 - (\kappa\rho)^{-1}p_\epsilon$ we can approximate

$$-p_\epsilon = \kappa\rho\frac{\partial u}{\partial x} \sim \kappa\rho_0\frac{\partial u}{\partial x} \tag{1.29}$$

and substituting in (1.28) we obtain the wave equation

$$\kappa\frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial t^2}. \tag{1.30}$$

Probably your mathematical brain is complaining "too many approximations for me". Do not forget that models are models, a sometimes lousy approximation to reality. Anyway it is amazing that we can say something. Gravitational waves have been recently detected and they come from making up a wave equation out of a linear approximations of a scaring nonlinear equation involving the curvature tensor that we have no idea how to solve.

It is not necessary going that far to meet nonlinearity. For instance, if we consider water waves in a thin channel, the waves are transverse and a small oscillation analysis as presented before for sound but taking into account the pressure due to the vertical columns of water to each side of an element of fluid, leads to (1.30) with $\kappa = gh$ where g is the

[2] Just to boast: In cosmology the Universe as a whole is considered to be a fluid with equation of state $p = 0$ since 47000 years after Big Bang (matter dominated era) and $p = \rho/3$ in the early stages of Universe (radiation dominated era). The galaxy separation distance and other astronomical data are explained with the acoustic waves generated in the primordial plasma. If you want to lose your faith, read [Ste16, §16–§19].

gravitational acceleration and h is the deepness of the channel. It implies that the velocity of water waves is $\sqrt{gh}$ according to the model. This is more or less precise for shallow water but it is completely unrealistic in the middle of the ocean where the deepness is of the order of kilometers. The value of $\sqrt{gh}$ does not match with the actual speed and it sounds unnatural that the velocity could be affected in any way by the bottom of the ocean. A revised model taking also into account the vertical acceleration, gives in (1.30) a κ depending on the frequency of the waves. In this way the coefficient of the equation depends on the solution and we have a highly complicated example of non-linearity that is an active field of research for theoreticians.

Suggested Readings. This is classic material that can be found in many books for undergraduates (for instance [AF67]). Talking about classics, perhaps it is worth to have a look to the translations in `http://www.17centurymaths.com/` of early works by L. Euler, specially E305.

1.2 Mathematical methods and results

1.2.1 Basic Fourier series and integrals

J. Fourier defended in his celebrated memoir [Fou88] that periodic functions can be analyzed in terms of cosine and sine functions but he was not able to provide a proof, this task was completed by P.G.L. Dirichlet in 1828, six years after the publication of [Fou88]. Nowadays we consider it as an important result not so difficult to prove for regular functions. Before giving any proof, we are going to try to understand why it should be true.

A formal simplification occurs unifying cosine and sine into the complex exponential. As it is going to appear everywhere, we use a special notation:

$$e(x) := e^{2\pi i x} = \cos(2\pi x) + i\sin(2\pi x). \tag{1.31}$$

It is also convenient to introduce the notation $\mathbb{T}$ to mean $\mathbb{R}/\{x \mapsto x+1\}$ with the quotient topology that we can visualize as any interval of length 1 gluing together the extreme points (it is usually called *torus* but in this 1D case it is rather a circle). In this way, a function $f : \mathbb{T} \longrightarrow \mathbb{C}$ can be unwrapped as a 1-periodic function $f : \mathbb{R} \longrightarrow \mathbb{C}$ and we identify such interpretations. For instance $f \in C^k(\mathbb{T})$ means $f \in C^k$ and 1-periodic. For 1-periodic functions, the integral over any unit interval is often denoted as an integral over $\mathbb{T}$ because there is no ambiguity in the choice of the interval.

Let g be a function $g : \mathbb{Z} \longrightarrow \mathbb{C}$ such that $g(n+N) = g(n)$; if you prefer so, $\{g(n)\}_{n\in\mathbb{Z}}$ is a two-sided N-periodic sequence. It is a simple exercise to check

$$g(n) = \sum_{m\in I} a_m e(mn/N) \qquad \text{with} \quad a_m = \frac{1}{N}\sum_{k\in I} g(k)e(-mk/N) \tag{1.32}$$

where I is any set of N consecutive integers. Is it really simple? Yes, it is. Just substitute the formula for a_m and use that for $n, k \in I$

$$\frac{1}{N}\sum_{m\in I} e\big(m(n-k)/N\big) = \begin{cases} 1 & \text{if } n = k, \\ 0 & \text{if } n \neq k, \end{cases} \tag{1.33}$$

because the roots of the unity cancels, they are forces pulling in symmetric directions and no net force results.

If you substitute $g(n) = f(n/N)$ and let $N \to \infty$ then f comes a generic 1-periodic function $f : \mathbb{R} \longrightarrow \mathbb{C}$, at least in some intuitive way. The formula for a_m is a Riemann sum and we imagine that in the first formula of (1.32), I becomes $\mathbb{Z}$. With these hand waving manipulations we infer that for 1-periodic functions

$$f(x) = \sum_{m\in\mathbb{Z}} a_m e(mx) \qquad \text{with} \quad a_m = \int_{\mathbb{T}} f(t)e(-mt)\, dt. \tag{1.34}$$

The sum is the famous *Fourier series*. We must humbly recognize that this is not a rigorous proof but we can proudly trumpet that it is closer to it than Fourier attempts. Even more, with the standards of rigor at the first quarter of the 19th century perhaps it would have been admitted as a proof. From the modern point of view, it is possible to stain the argument with some ϵ's and δ's and turn it into an actual proof when f is regular, for instance $f \in C^\infty(\mathbb{T})$.

If we admit (1.34) for 1-periodic functions, the T-periodic functions are covered by a change of variables

$$f(x) = \frac{1}{T}\sum_{m\in\mathbb{Z}} a_m e(mx/T) \qquad \text{where} \quad a_m = \int_{-T/2}^{T/2} f(t)e(-nt/T)\, dt. \tag{1.35}$$

As before, we can read the sum as a Riemann sum and one expects in the limit $T \to \infty$ the *Fourier inversion formula*

$$f(x) = \int_{-\infty}^{\infty} \widehat{f}(\xi)e(x\xi)\, d\xi \qquad \text{where} \quad \widehat{f}(\xi) = \int_{-\infty}^{\infty} f(t)e(-t\xi)\, dt. \tag{1.36}$$

This already appears in Fourier's memoir [Fou88]. Doing justice to him, the operator $\mathcal{F} : f \longmapsto \widehat{f}$ is called the *Fourier transform*.

Now we are going to tell a fairy tale suggesting a line of attack to get (1.34) and (1.36). Do you remember the *Dirac delta*? It was introduced by P.A.M. Dirac when mathematizing quantum mechanics. In physics it is managed all the time as a "function" δ satisfying $\int_{\mathbb{R}} \delta = 1$ and $\int_{\mathbb{R}} f\delta = f(0)$. For pure mathematicians writing something like this is heretical and they tell the same thing in a wordy way. They say that is a *distribution* (function is tabooed here), a certain type of operator acting on smooth functions and it is represented by an *approximation to the identity*, a collection of functions $\eta_\epsilon(x) = \epsilon^{-1}\eta(x/\epsilon)$ with $\eta \in L^1 \cap C^\infty$ and $\int \eta = 1$. For $\epsilon > 0$ small, we shrink the X-axis and stretch the Y-axis, then $\int_{\mathbb{R}} \eta_\epsilon = 1$ and it is not difficult to prove $\lim_{\epsilon\to 0^+} \int_{\mathbb{R}} \eta_\epsilon f = f(0)$ for $f \in C_0^\infty$. Although pure mathematicians do not say it in public, in the darkness of their offices, the Dirac delta δ is the function $\lim_{\epsilon\to 0^+} \eta_\epsilon$.

The bottom line is now that (1.32) was very easy to prove because we had the simple formula (1.33) for a displaced finite Dirac delta, whatever it means. Let us state two spooky analogues that suit our needs:

$$\delta_P(x) = \sum_{n\in\mathbb{Z}} e(nx) \qquad \text{and} \qquad \delta(x) = \int_{-\infty}^{\infty} e(x\xi)\, d\xi \tag{1.37}$$

where δ_P is the 1-periodic Dirac delta $\delta_P(x) = \sum_{n\in\mathbb{Z}} \delta(x-n)$, sometimes called *Dirac comb*. Pure mathematicians rip their garments apart when see (1.37) but for theoretical physicists it is the daily bread. In signal analysis you should have (1.37) in mind even if you do not dare to write it.

If we believe in fairies and in (1.37), for f 1-periodic

$$f(x) = \int_{\mathbb{T}} f(t)\delta_P(x-t)\,dt = \sum_{m\in\mathbb{Z}} \int_{\mathbb{T}} f(t)e(-nt)e(nx)\,dt = \sum_{m\in\mathbb{Z}} a_m e(mx) \tag{1.38}$$

and we have (1.34). In the same way, we get (1.36) from

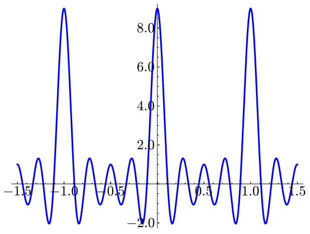
$$f(x) = \int_{-\infty}^{\infty} f(t)\delta(x-t)\,dt = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t)e(-t\xi)e(x\xi)\,dtd\xi = \int_{-\infty}^{\infty} \widehat{f}(\xi)e(x\xi)\,d\xi. \tag{1.39}$$

The fairies are hidden in changing the order of summation or integration.

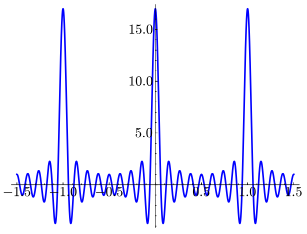
Although it is not explicitly mentioned in the most of the textbooks for mathematicians, (1.37) and our proof by vigorous handwaving [RD05, p.28] guide the actual proof of the basic convergence theorems in Fourier analysis. The most expeditious way of dodging our qualms about infinity in the first formula, is cutting the series to the so-called *Dirichlet kernel*

$$D_N(x) = \sum_{n=-N}^{N} e(nx), \qquad D_N(x) = \begin{cases} \dfrac{\sin\left(\pi(2N+1)x\right)}{\sin(\pi x)} & \text{if } x \notin \mathbb{Z}, \\ 2N+1 & \text{if } x \in \mathbb{Z}. \end{cases} \tag{1.40}$$

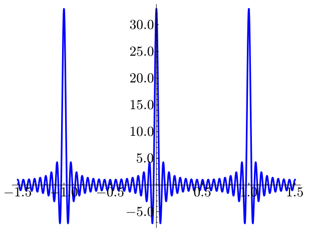
The explicit formula to the right follows from the sum of a geometric finite sequence. When N grows, it has a big peak around each integer and the integral of D_N equals 1 on $\mathbb{T}$, then it sounds like an approximation to the identity. The following plots show the aspect of the graph of D_N for some values of N.



N = 4



N = 8



N = 16

A proxy of the first equation in (1.38) is

$$f(x) = \int_{-1/2}^{1/2} f(t)D_N(x-t)\,dt + \int_{-1/2}^{1/2} \left(f(x)-f(t)\right)D_N(x-t)\,dt. \tag{1.41}$$

The first integral is

$$S_N f(x) = \sum_{m=-N}^{N} a_m e(mx) \qquad \text{with} \quad a_m = \int_{\mathbb{T}} f(t)e(-mt)\,dt, \tag{1.42}$$

the partial sum of the Fourier series (1.34). The pointwise convergence $S_N f(x) \to f(x)$ is equivalent to the vanishing of the second integral in the limit. By the periodicity, this can be written as

$$\lim_{N\to\infty} \int_{-1/2}^{1/2} \big(f(x) - f(x-t)\big) \frac{\sin\big(\pi(2N+1)t\big)}{\sin(\pi t)}\, dt = 0. \tag{1.43}$$

If $f \in C^1(\mathbb{T})$, for t very close to 0, $\big(f(x)-f(x-t)\big)/\sin(\pi t)$ can be extended to a continuous, in particular bounded, function, while for t not very close close to 0, we can integrate by parts to get an $O(N^{-1})$ when integrating $\sin\big(\pi(2N+1)t\big)$. This scheme proves

Theorem 1.2.1. *If $f \in C^1(\mathbb{T})$ then $S_N f \to f$ uniformly.*

One can squeeze the argument incorporating some technical tricks. One of them deserves a proper name (two, to say the full truth), the *Riemann-Lebesgue lemma*. It seems that it was proved firstly by B. Riemann [Bré02, p.15] whose outstanding contribution to Fourier series has been somewhat eclipsed for other of his famous works [Cór08].

Lemma 1.2.2 (Riemann-Lebesgue)**.** *If $f \in L^1$ then $\widehat{f}(\xi) \to 0$ when $\xi \to \infty$.*

The proof is easy: Integrable functions can be approximated by step functions and step functions can be approximated by functions in C_0^1 and for the latter the result is obvious integrating by parts.

By Riemann-Lebesgue lemma, if $\big(f(x) - f(x-t)\big)/\sin(\pi t)$ is integrable for a given x (as a function of t), we have (1.43) and then $S_N f(x) \to f(x)$. Except for some variations [Bré02, §1.3] [Zyg88], this is called *Dini's theorem*. It implies that if $f \in C(\mathbb{T})$ satisfies a *Hölder condition* of order α, i.e. $|f(x) - f(y)| = O\big(|x-y|^\alpha\big)$ for some $0 < \alpha \le 1$, then $S_N f \to f$ pointwise. It is known that the continuity of f is not enough to assure everywhere convergence[3] [Kör88, §15].

If f has lateral limits everywhere, for each x define

$$M(x) = \frac{f(x^+) + f(x^-)}{2}. \tag{1.44}$$

Then proceeding as in (1.41) and (1.43), $S_N f(x) \to M(x)$ if and only if

$$\int_{-1/2}^{1/2} \big(M(x) - f(x-t)\big) \frac{\sin\big(\pi(2N+1)t\big)}{\sin(\pi t)}\, dt \to 0. \tag{1.45}$$

Equivalently,

$$\int_{0}^{1/2} \big(M(x) - f(x-t) - f(x+t)\big) \frac{\sin\big(\pi(2N+1)t\big)}{\sin(\pi t)}\, dt \to 0. \tag{1.46}$$

[3] In fact in [KK66] it is proved that given a zero measure set $E \subset \mathbb{T}$ there exists a continuous function such that for each $x \in E$, $S_N f(x)$ does not converge as $N \to \infty$. The proof is shorter than one might think but, as somebody said, this margin is too small to include it.

If we assume also the existence of lateral derivatives everywhere and that they are integrable, we conclude that $S_N f(x)$ converges to $M(x)$ everywhere. The existence of the lateral derivatives can be relaxed in a set of measure zero. Essentially *bounded variation functions* are almost everywhere differentiable functions with integrable derivative [Rud74]. Then the most general result that we can get with this line of reasoning is [Kat68, II.2]

Theorem 1.2.3. *If f is of bounded variation in $\mathbb{T}$ then we have $S_N f(x) \to M(x)$ when $N \to \infty$ for any x and $M(x)$ as in* (1.44).

When one tries to extend this kind of arguments to deal with (1.36) a new problem appears due to the fact we have to worry about the convergence of two infinite integrals. Taking into account the second formula in (1.37), the natural analogue of the Dirichlet kernel is

$$\int_{-N}^{N} e(x\xi)\, d\xi = \frac{\sin(2\pi N x)}{\pi x}. \tag{1.47}$$

The formula that parallels (1.41) is

$$f(x) = \int_{-\infty}^{\infty} f(t) \int_{-N}^{N} e((x-t)\xi)\, d\xi dt + \int_{-\infty}^{\infty} \big(f(x) - f(t)\big) \frac{\sin\big(2\pi N(x-t)\big)}{\pi(x-t)}\, dt. \tag{1.48}$$

If $f(x) = 0$ everything is OK and we can repeat the argument at these value of x for instance when $f \in L^1 \cap C^1$ but if $f(x) \neq 0$ the last integral does not make sense as a Lebesgue integral even if $f(t) \to 0$ smoothly when $t \to \infty$ because the function under the integral decays as $f(x)/\pi t$ which is not integrable. Then extra subtleties must be introduced to give a meaning to the previous expression and conclude an analogue of Theorem 1.2.1, namely [Hel91, §1.2]

Theorem 1.2.4. *If $f \in L^1 \cap C^1$ then*

$$f(x) = \lim_{N\to\infty} \int_{-N}^{N} \widehat{f}(\xi) e(x\xi)\, d\xi. \tag{1.49}$$

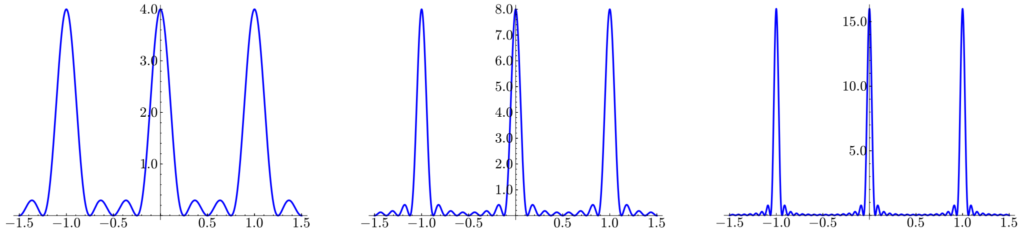
Establishing minimal conditions to have

$$S_N f(x) \to f(x) \qquad \text{and} \qquad \int_{-N}^{N} \widehat{f}(\xi) e(x\xi)\, d\xi \to f(x) \tag{1.50}$$

with "few exceptions" is a big topic in classical harmonic analysis with some deep results. The most celebrated and the culmination of the theory is the *Carleson-Hunt theorem* [Car66] [Hun68] [LT00] which states $S_N f(x) \to f(x)$ almost everywhere for any $f \in L^p(\mathbb{T})$, $p > 1$. In signal processing this is not really very important in practice because the sharp truncation of the Fourier series or the Fourier integral may be natural for our mathematical mind but it is a bad idea if we want to approximate a slightly rough function by sines and cosines. To give a theoretical basis to this assertion, one can say that there are better ways than truncation to get Dirac deltas as limits in (1.37). For instance, we have

$$\sum_{n=-N}^{N} \Big(1 - \frac{|n|}{N}\Big) e(nx) = \frac{\sin^2(\pi N x)}{N \sin^2(\pi x)} \qquad \text{and} \qquad \int_{-N}^{N} \Big(1 - \frac{|\xi|}{N}\Big) e(x\xi)\, d\xi = \frac{\sin^2(\pi N x)}{N\pi^2 x^2}. \tag{1.51}$$

The advantage of these functions is that they are integrable and with L^1-norm uniformly bounded in N. If we compare the graphs of the first function for several values of N with those of $D_N(x)$ we guess a better approximation to our idea of the Dirac comb δ_P



N = 4 N = 8 N = 16

Usually this function is called *Fejér kernel*. It mollifies Dirichlet's kernel avoiding sharp cut high frequencies. The outcome is a more clear, and practical, convergence theorem

Theorem 1.2.5 (Fejér)**.** *For $f \in C(\mathbb{T})$, we have*

$$f(x) = \lim_{N\to\infty} \sum_{m=-N}^{N} \Big(1 - \frac{|m|}{N}\Big) a_m e(mx) \tag{1.52}$$

uniformly in x, with a_m as in (1.34)*.*

The same can be stated for the Fourier transform but the integrability is not a consequence of continuity in this case and it must be imposed.

Theorem 1.2.6. *For $f \in C(\mathbb{R})$ integrable, we have*

$$f(x) = \lim_{N\to\infty} \int_{-N}^{N} \Big(1 - \frac{|\xi|}{N}\Big) \widehat{f}(\xi) e(x\xi)\, d\xi \tag{1.53}$$

uniformly over compact sets.

Formally, putting $N = \infty$ in these results we obtain the formulas (1.34) and (1.36). The proofs of these theorems follow the scheme sketched before. We include them here anyway.

Proof of Theorem 1.2.5. Let $F_N(x)$ be the first function in (1.51). Clearly $\int_{\mathbb{T}} F_N = 1$ then we can write as in (1.41)

$$f(x) = \int_{\mathbb{T}} f(t) F_N(x-t)\, dt + \int_{\mathbb{T}} \big(f(x) - f(t)\big) F_N(x-t)\, dt. \tag{1.54}$$

The first integral gives the sum appearing in the statement and it remains to prove that the last integral goes to zero. Substitute $\mathbb{T}$ by the interval $[x - 1/2, x + 1/2]$. As it is compact, f is uniformly continuous and then for every $\epsilon > 0$ there exist $0 < \delta < 1/2$ such that $|f(x) - f(t)| < \epsilon$ whenever $|x - t| < \delta$ and these values contribute to the integral less than ϵ because $\int F_N = \int |F_N| = 1$. On the other hand, $F_N(u) \to 0$ uniformly when $\delta < |u| < 1/2$. □

Proof of Theorem 1.2.6. If $\mathcal{F}_N(x)$ is the second function in (1.51), we mimic the previous proof starting with

$$f(x) = \int_{-\infty}^{\infty} f(t)\mathcal{F}_N(x-t)\,dt + \int_{-\infty}^{\infty} \big(f(x)-f(t)\big)\mathcal{F}_N(x-t)\,dt. \tag{1.55}$$

As $f \in L^1$, we can change the order of integration (Fubini's theorem) and prove that the first integral coincides with the integral in the statement. Now we have to prove that the second integral goes to zero. For each x, let $\mathcal{C}_x = \{t \,:\, |x-t| > 1/2\}$. If the values of x are restricted to a compact set, $f(x)$ remains bounded, hence $\int_{\mathcal{C}_x} f(x)\mathcal{F}_N(x-t)\,dt = O(N^{-1})$. On the other hand, $\int_{\mathcal{C}_x} f(t)\mathcal{F}_N(x-t)\,dt = O(N^{-1})$ because $\mathcal{F}_N(x-t) = O(N^{-1})$ and f is integrable. Therefore it remains to prove that the contribution of $|x-t| < 1/2$ is negligible and it follows as in the previous proof. Note that the functions $F_N(u)$ and $\mathcal{F}_N(u)$ are comparable when $|u| < 1/2$. □

Let us move to some explicit examples of Fourier series. In our pocket calculator we do not find many periodic functions different from cosine and sine or easily related to them. Perhaps the only exception is the fractional part $x - \lfloor x \rfloor$ where

$$\lfloor x \rfloor := \min\{n \in \mathbb{Z} \,:\, n \le x\} \tag{1.56}$$

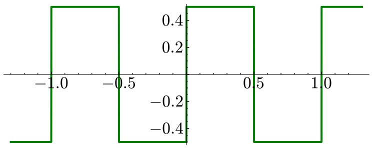
is the integral part. With it we are going to construct some real 1-periodic functions that are famous enough to deserve a name. We call them signals and write t instead of x to emphasize that they appear in engineering. We present the expansions in terms of sines and cosines, the complex exponential form would be obtained substituting the *Euler formulas*

$$\sin(2\pi t) = \frac{e(t)-e(-t)}{2i} \qquad \text{and} \qquad \cos(2\pi t) = \frac{e(t)+e(-t)}{2}. \tag{1.57}$$

The plots of the signals are drawn with continuous lines as it would be seen in an oscilloscope. Recall that in jump discontinuities the Fourier series converges to the middle point by Theorem 1.2.3. If you play these signals as sounds (there are online applications to do it), you will note that they sound differently although they have the same frequency. The shape of a sound wave gives the *timbre* (tone quality), its frequency the *pitch* and its amplitude the *volume*.

The first example is the *square wave*, the simplest periodic digital signal taking the values -1 and 1.

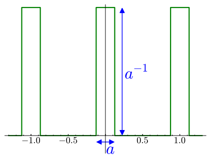
(1.58) $$f(t) = \frac{1}{2}(-1)^{\lfloor 2t \rfloor}$$



$$= \frac{2}{\pi}\sum_{n=1}^{\infty} \frac{\sin\big(2\pi(2n-1)t\big)}{2n-1}.$$

The *pulse wave* is... guess it! a pulse of width $a < 1$ that tends to the Dirac comb when $a \to 0$. Note that this limit is formally coherent with the expansion of δ_P in (1.37).
(1.59)

$$f(t) = \frac{\max\Big(0, (-1)^{\lfloor 2t+a \rfloor} - (-1)^{\lfloor 2t-a \rfloor}\Big)}{2a}$$



$$= \sum_{n=-\infty}^{\infty} \operatorname{sinc}(an)\cos(2\pi n t)$$

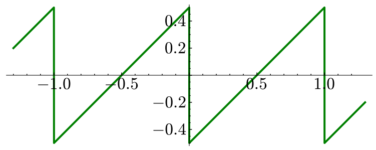
where sinc is a useful abbreviation used in signal processing to mean

$$\operatorname{sinc}(x) = \int_{-1/2}^{1/2} e(x\xi)\, d\xi = \begin{cases} \dfrac{\sin(\pi x)}{\pi x} & \text{if } x \neq 0, \\ 1 & \text{if } x = 0. \end{cases} \tag{1.60}$$

The “c” in sinc comes because in a famous early paper [Whi15] it was said to be a *cardinal function*. According to some authors sinc is an abbreviation for *sinus cardinal* or *sinus cardinalis* but this seems that the use of this term is more recent.

The *sawtooth wave* has again a self-explanatory name, at least until you see the next example. It can be proved that the series is truncated to $n \le N$, it produces an error $O\big((1 + N|t - \lfloor t + 1/2 \rfloor|)^{-1}\big)$.

$$f(t) = t - \lfloor t \rfloor - \frac{1}{2} \tag{1.61}$$

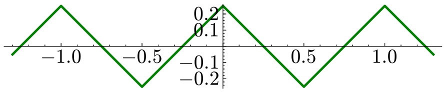


$$= -\frac{1}{\pi}\sum_{n=1}^{\infty} \frac{\sin(2\pi n t)}{n}.$$

Choosing $t = 1/4$ we get Leibniz series $\pi/4 = 1 - 3^{-1} + 5^{-1} - 7^{-1} + 9^{-1} - \dots$ with is really pretty but useless for precise numerical approximations of π.

In the *triangle wave* there are not discontinuities and the convergence is quicker. If you plot few terms in the Fourier series you will notice a very good approximation with small differences near the corners.
(1.62)

$$f(t) = \frac{1}{4} - \Big|t - \lfloor t + \frac{1}{2} \rfloor\Big|$$



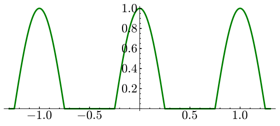
$$= \frac{2}{\pi^2}\sum_{n=1}^{\infty} \frac{\cos\big(2\pi(2n-1)t\big)}{(2n-1)^2}.$$

In amplitude modulated telecommunications (today they are old-fashioned) the information is given by the amplitude of a highly oscillating signal. We cannot plug in directly our headphones to this input because the average signal is zero in short periods (for instance in AM broadcasting the frequency is/was comparable to $10^6\, Hz$) and we would not hear anything. A primary need is to rectify the signal, choosing for instance the upper half. In this way short time averages are proportional to the amplitude and our headphones are responsive to them. This is achieved in practice with a *diode*, an electronic component allowing the current to flow only in one direction. In the early days they were valves and now they are semiconductors[4].

This story give us an excuse to add to the examples the *rectified cosine wave.*

(1.63)

$$f(t) = \max\left(0, \cos(2\pi t)\right)$$



$$= \frac{\cos(2\pi t)}{2} - \frac{1}{\pi}\sum_{n=-\infty}^{\infty} \frac{(-1)^n \cos(4\pi n t)}{4n^2-1}.$$

In the case of the Fourier transforms there are several examples with non artificial closed expressions. By the inversion formula, the Fourier transform operator $\mathcal{F}$ verifies $\mathcal{F}(\mathcal{F}f)(x) = f(-x)$. As the following examples are even, we can read them in both directions. It does not add anything new in the self-reciprocal examples.

The first example is arguably the most important, the *Gaussian function*

$$e^{-ax^2} \overset{\mathcal{F}}{\longleftrightarrow} \sqrt{\frac{\pi}{a}}e^{-\pi^2\xi^2/a} \qquad \text{for } a > 0. \tag{1.64}$$

Of course e^{-ax} has an ugly exponential growth for $x < 0$ but we can amend it introducing an absolute value. We lose the differentiability at $x = 0$.

$$e^{-a|x|} \overset{\mathcal{F}}{\longleftrightarrow} \frac{2a}{4\pi^2\xi^2 + a^2} \qquad \text{for } a > 0. \tag{1.65}$$

A smooth variant with exponential decay is given by the hyperbolic trigonometric function sech.

$$\operatorname{sech}(ax) \overset{\mathcal{F}}{\longleftrightarrow} \frac{\pi}{a}\operatorname{sech}\left(a^{-1}\pi^2\xi\right) \qquad \text{for } a > 0. \tag{1.66}$$

No power is integrable because a problem appears either at 0 or at ∞. On the other hand, it is well-known that $\int_0^\infty x^{-\nu}\cos x\, dx$ is meaningful and finite as a Riemann improper integral for $0 < \nu < 1$ (it is not as a Lebesgue integral). In this way, we can define the Fourier transform of $|x|^{-\nu}$ except at the origin.

$$|x|^{-\nu} \overset{\mathcal{F}}{\longleftrightarrow} \frac{\pi|2\pi\xi|^{\nu-1}}{\Gamma(\nu)\cos(\pi\nu/2)} \qquad \text{for } 0 < \nu < 1. \tag{1.67}$$

Here $\Gamma(\nu) = \int_0^\infty x^{\nu-1}e^{-x}\, dx$ is the usual *Gamma function.*

Although it is not regular, the Fourier transform of the characteristic function of the centered unit interval plays an important role in signal processing. This was the reason to introduce the definition (1.60).

$$\chi_{[-1/2,1/2]}(x) \overset{\mathcal{F}}{\longleftrightarrow} \operatorname{sinc}(x). \tag{1.68}$$

[4] Well, this use of semiconductors is not so modern if you have heard about the handmade crystal radio your great-great grandparent built with a mineral of galena and no batteries!

Again $\operatorname{sinc}(x)$ is not a Lebesgue integrable function and then for the left arrow we have to use the Riemann improper integral. With it we recover $\chi_{[-1/2,1/2]}(x)$ except at $x = \pm 1/2$ where the value is $1/2$.

Suggested Readings. There are many and excellent books on basic Fourier analysis and preferring one or another is a question of personal taste. Among my favorites are [Kör88], [DM72], [Kat68], [Fol92] and [Hel91]. The classic [Zyg88] is certainly old and encyclopedic although still very advisable because it reflects the freshness of some natural basic questions. A very basic online tone generator can be found in `http://www.szynalski.com/tone-generator/`.

1.2.2 Some properties

A simple and useful property is that we can integrate by parts to get the Fourier coefficients or the Fourier transforms of the k-derivative of a function. Namely for $f \in C^k(\mathbb{T})$ and for $f \in C^k(\mathbb{R})$ with $f^{(k)}$ integrable, we have respectively

$$(1.69) \qquad a_n = (2\pi i n)^{-k} \int_{\mathbb{T}} f^{(k)}(x) e(-nx)\, dx \qquad \text{and} \qquad \widehat{f}(\xi) = (2\pi i \xi)^{-k} \widehat{f^{(k)}}(\xi)$$

for $n \neq 0$, $\xi \neq 0$, where a_n are the Fourier coefficients of f.

These formulas allow to extend our list of explicit examples. The first one justifies in general term by term integration of the Fourier series of a zero average function. As an example of the second, the Fourier transform of $-2\pi x e^{-\pi x^2}$ is $2\pi i \xi e^{-\pi \xi^2}$ because the former is the derivative of $e^{-\pi x^2}$ whose Fourier transform is itself by (1.64).

An interesting theoretical and not so theoretical consequence of (1.69) is that, under the stated hypotheses, we have

$$(1.70) \qquad a_n = O(|n|^{-k}) \qquad \text{and} \qquad \widehat{f}(\xi) = O(|\xi|^{-k})$$

when $n \to \infty$ and $\xi \to \infty$, respectively. In this way, the Fourier coefficients of $f \in C^\infty(\mathbb{T})$ decay quicker than any negative power and this is good news for numerical methods because we can approximate f with few terms of the Fourier series[5]. In the case of the Fourier transform, if f is in the Schwartz class then $\widehat{f}$ is also there. Actually this is the property that gives relevance to this space of functions, it provides a simple environment to work which is preserved by Fourier transforms.

In $\mathbb{T}$ and $\mathbb{R}$, the translations $x \mapsto x + \beta$ preserve the homogeneous structure of the space. If f and g are related by $f(x) = g(x + \beta)$ then we have

$$(1.71) \qquad a_n = e(\beta n) b_n \qquad \text{and} \qquad \widehat{f}(\xi) = e(\beta\xi)\widehat{g}(\xi),$$

where a_n and b_n are, respectively, the Fourier coefficients of f and g.

[5] As an aside, this is very relevant for calculations in celestial mechanics because periodic models or combinations of periodic models are quite accurate. Think for instance in Kepler laws.

An arbitrary dilation in general does not preserve 1-periodicity, then we only consider the case of the Fourier transform. We have

$$g(x) = f(x/\delta) \qquad \text{implies} \qquad \widehat{g}(\xi) = \delta \widehat{f}(\delta\xi) \qquad \text{for } \delta > 0. \tag{1.72}$$

This inverse scaling under Fourier transform is something to keep in mind for the next subsection.

The functions $\{e(nx)\}_{n\in\mathbb{Z}}$ are orthonormal with respect to the natural functional scalar product $\langle f, g\rangle = \int \bar{f}g$. For regular functions Theorem 1.2.1 and (1.70) assure that we can express this scalar product in terms of the Fourier coefficients integrating term by term. Namely, if $\{a_n\}_{n\in\mathbb{Z}}$ and $\{b_n\}_{n\in\mathbb{Z}}$ are the Fourier coefficients of f and g, we have

$$\int_{\mathbb{T}} |f|^2 = \sum_{n\in\mathbb{Z}} |a_n|^2 \qquad \text{and} \qquad \int_{\mathbb{T}} \bar{f}g = \sum_{n\in\mathbb{Z}} \overline{a}_n b_n. \tag{1.73}$$

This has a tremendous significance in the theoretical side because $L^2(\mathbb{T})$ is a *Hilbert space* with the scalar product $\langle f, g\rangle$, it means that we can take limits (it is complete) and the same happens with $\ell^2(\mathbb{Z})$, the bilateral sequences with bounded square norm. Each function in $L^2(\mathbb{T})$ can be approximated by regular functions, hence (1.73) holds true for any square integrable function. The convergence of the Fourier series arises in $L^2(\mathbb{T})$ without further conditions.

Theorem 1.2.7. *If $f \in L^2(\mathbb{T})$ then $S_N f \to f$ in $L^2(\mathbb{T})$ i.e., $\lim_{N\to\infty} \|S_N f - f\|_2 = 0$.*

In this way, the L^2 theory becomes natural and easy. As well as their theoretical importance, the formulas (1.73) are the source of many impressive identities. For instance, when we apply the first to the sawtooth wave (1.61), we obtain $\int_0^1 (x-1/2)^2\,dx = \sum_{n\neq 0}(2\pi n)^{-2}$ that gives readily $\sum_{n=1}^{\infty} n^{-2} = \pi^2/6$, a result that (proved in a quite different way) boosted the fame of Euler when he was 28 years old.

For Fourier transforms of functions in the Schwartz class, we have similar formulas

$$\int_{-\infty}^{\infty} |f|^2 = \int_{-\infty}^{\infty} |\widehat{f}|^2 \qquad \text{and} \qquad \int_{-\infty}^{\infty} \bar{f}g = \int_{-\infty}^{\infty} \overline{\widehat{f}}\widehat{g}. \tag{1.74}$$

To prove the second (the first corresponds to $f = g$), it is enough to apply the inversion formula (1.36) twice

$$\int_{-\infty}^{\infty} \bar{f}g = \int_{-\infty}^{\infty} \bar{f}(x) \int_{-\infty}^{\infty} \widehat{g}(\xi)e(x\xi)\,dxd\xi = \int_{-\infty}^{\infty} \overline{\widehat{f}}\widehat{g}. \tag{1.75}$$

Again, (1.74) allows to construct a nice $L^2(\mathbb{R})$ theory of the Fourier transform except for... our definition of $\widehat{f}$ may be nonsensical for $f \in L^2(\mathbb{R})$. We have $L^1(\mathbb{T}) \subset L^2(\mathbb{T})$ and for $f \in L^2(\mathbb{T})$ its Fourier coefficients always exist while on the other hand, due to the lack of compactness, $L^1(\mathbb{R}) \not\subset L^2(\mathbb{R})$. If $f \in L^2(\mathbb{R}) - L^1(\mathbb{R})$, we have to give sense to $\widehat{f}$ and if we want to save (1.74) the procedure is clear: $\widehat{f}$ should be the limit in $L^2(\mathbb{R})$ of $\widehat{f}_n$ with $\|f_n - f\|_2 \to 0$. Three methods to define $\widehat{f}$ in $L^2(\mathbb{R})$ keeping (1.74) and the inversion formula are discussed in [DM72, §2.3-2.5].

The relations (1.73) and (1.74) are called generically *Parseval identity* or *Plancherel identity*. Some authors apply both names indistinctly and some others establish differences.

Arguably the most important property for signal processing is the behavior of the *convolution* $f * g$ under Fourier analysis. In $\mathbb{T}$ and in $\mathbb{R}$ the convolution is defined respectively as

$$(1.76)\qquad (f * g)(x) = \int_{\mathbb{T}} f(t)g(x-t)\,dt \qquad \text{and} \qquad (f * g)(x) = \int_{-\infty}^{\infty} f(t)g(x-t)\,dt.$$

Note that it is commutative $f * g = g * f$. If you look up the convergence theorems, you will see that we have already used convolutions without defining them. In signal processing they allow to introduce factors in Fourier series and integrals acting as filters. Mathematically, if a_n, b_n and c_n are the Fourier coefficients of f, g and $f * g$ with $f, g \in L^2(\mathbb{T})$ or if $f, g \in L^2(\mathbb{R})$, we have respectively

$$(1.77)\qquad c_n = a_n b_n \qquad \text{and} \qquad (f * g)\widehat{\ } = \widehat{f}\,\widehat{g}.$$

For instance, the convolution with the Dirichlet kernel (1.40) cuts the Fourier series to give the partial sum S_N in (1.42). In general, if we want to select a certain set of frequencies we must consider the convolution with a function such that its Fourier transform vanishes outside this set.

Suggested Readings. The topics discussed here still belong to the basic theory of Fourier series and integrals and then they are covered by the monographs suggested in the previous subsection.

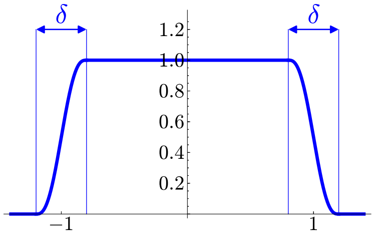
1.2.3 Uncertainty

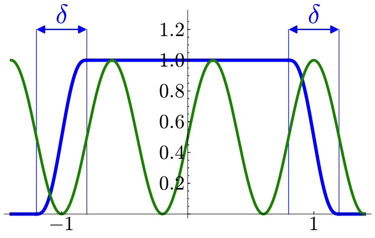
In few words, the essence of the *uncertainty principle* is that with waves of frequencies less than ν one misses details of size much smaller than ν^{-1}.

For instance, if one wants a good approximation of a function $f = f(x)$ in such a way that its variation in intervals of length δ is well represented, then the truncated Fourier series $\sum_{|n|\le N} a_n e(nx)$ or the truncated Fourier integral $\int_{-N}^{N} \widehat{f}(\xi)e(x\xi)\,d\xi$ are useless to mimic f with this detail if $N\delta$ is small. The range of frequencies is at least the inverse of the required precision.

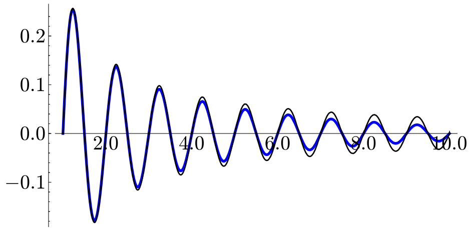
Let us focus on regular functions, for instance in the Schwartz space, and in Fourier integrals instead of Fourier series. The "inverse law" is linked to the simple scaling property (1.72). Say that we have a function $F = F(x)$ and we add a "detail" of size δ not modifying the mass of F. We can model this detail as adding a function $\varphi(x/\delta)$ with φ compactly supported in an interval of length 1 and $\int \varphi = 0$ to preserve the mass. By (1.72), the Fourier transform of $G(x) = F(x) + \varphi(x/\delta)$ differs from that of F in $\delta\widehat{\varphi}(\delta\xi)$. We know that $\widehat{\varphi}(0) = \int \varphi = 0$ then this term is expected to be negligible when $\delta\xi$ is small and we need larger values of ξ to notice a difference between F and G. If the "detail" has size δ but it is oscillatory, it does not match the model $\varphi(x/\delta)$ of the previous analysis and it might resonate only with large frequencies and it could take even larges values ξ to detect that there is something there. We shall illustrate the situation with a highly oscillatory example later.

Let us expand the idea to cover a common situation. Imagine that we want to use Fourier analysis for $f = \chi_{[-1,1]}$. It is not even continuous and its Fourier transform shows a poor decay like $|\xi|^{-1}$. We decide to regularize it as a function f_δ in the Schwartz space such that $f = f_\delta$ except in the intervals $[-1-\delta/2, -1+\delta/2]$ and $[1-\delta/2, 1+\delta/2]$ and preserving the mass (the integral) in each of them. With an eye to applications, we can think that this is a winning strategy because $\widehat{f_\delta}$ is now rapidly decreasing by (1.70) and f and f_δ are very close, for instance in L^2 norm. The drawback is that a wavelength greater than δ skips the intervals and no change is noticed.

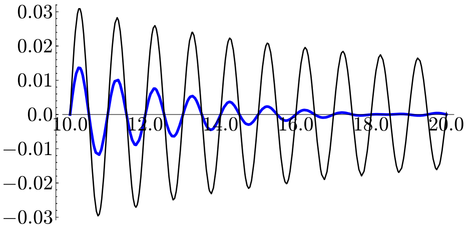




If we do not employ frequencies greater than δ^{-1}, $\widehat{f_\delta} - \widehat{f_{\delta'}}$ with $\delta' < \delta$ is like the zero function. If we think in f as f_0, the conclusion is that using Fourier transforms we cannot distinguish f from its regularization until we do not reach high frequencies. This is the uncertainty in this case. In particular, we cannot profit from the quick decay of $\widehat{f_\delta}$ until ξ is very large. These are actual graphics for $\delta = 0.1$ (using a C^3 regularization instead of C_0^∞ to ease some computational aspects).



$\widehat{\mathsf{f}}$ and $\widehat{\mathsf{f}_\delta}$ in $[1,10]$

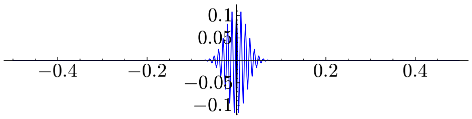


$\widehat{\mathsf{f}}$ and $\widehat{\mathsf{f}_\delta}$ in $[10,20]$

Note that both Fourier transforms are quite similar when $|\xi|$ is much less than δ^{-1} and we only notice the expected quick decay of $\widehat{f_\delta}$ when we have past δ^{-1}.

Let us see in an example with Fourier series that we do not have a "certainty" on the range of frequencies, it depends on the case. Consider the 1-periodic function

$$f(x) = \sum_{k=1}^{\infty} e^{-5-(k-100)^2/100} \cos(2\pi k x) \tag{1.78}$$



Its plot in $[-0.5, 0.5]$ shows that it is essentially the zero function except in an interval of length like 0.1. One can guess that a truncation of the Fourier series to $|n| \le 10$ or so is enough to show that there is something at this scale. This guess is wrong because the definition of f shows readily that its Fourier coefficients are given by $a_0 = 0$ and

$$a_n = \frac{1}{2} e^{-5-(|n|-100)^2/100} \qquad \text{for } n \in \mathbb{Z} - \{0\}. \tag{1.79}$$

This is infinitesimally small when $|n| \le 10$, being its maximal value in this range less than $5 \cdot 10^{-38}$. If we extend the truncated Fourier series to $|n| \le 20$ or $|n| \le 30$, still we get a near to zero function, we do not see anything suggesting the previous plot. In fact (1.79) shows that the Fourier coefficients are only noticeable where $|n|$ differs from 100 in something comparable to 10. We need these frequencies to recover the aspect of the function. This is due to the internal oscillations of f. There is a similar situation with the Fourier transform: If we multiply a function by $e(\beta x)$, its Fourier transform is shifted by β by (1.71). Adding several of this multiplications we can force the Fourier transform to live in many intervals.

Another way of thinking about the *uncertainty principle* and its most popular formulation is that one cannot localize simultaneously a function and its Fourier transform. There is a nice result due to G.H. Hardy [Har33] quantifying this property when the decay of f is controlled by a Gaussian function.

Theorem 1.2.8 (Hardy)**.** *If $f(x) = O\big(e^{-\alpha\pi x^2}\big)$ and $\widehat{f}(x) = O\big(e^{-\beta\pi x^2}\big)$ with $\alpha, \beta > 0$ and f is not identically zero, then $\alpha\beta \le 1$. Moreover the equality is reached if and only if f is a constant multiple of $e^{-\alpha\pi x^2}$.*

The proof is very short if one applies a (false) result of complex variables waving hands. To cover the loose ends see [Tao] or [DM72].

Proof (with a gap). We can assume $\alpha = 1$ by (1.72). In this case, it is enough to prove that if $f(x)$ and $\widehat{f}(x)$ are $O\big(e^{-\pi x^2}\big)$, then f is a constant multiple of $e^{-\pi x^2}$.

Under $f(x) = O\big(e^{-\pi x^2}\big)$, the function

$$F(z) = e^{\pi z^2} \int_{-\infty}^{\infty} f(t) e(-tz)\, dt \qquad \text{with} \qquad z = x + iy \in \mathbb{C}, \tag{1.80}$$

defines an entire function because the function under the integral is $O\big(e^{-\pi t^2 + 2\pi t y}\big)$.

If $y = 0$, using $\widehat{f}(x) = O\big(e^{-\pi x^2}\big)$ we have $F(z) = e^{\pi x^2} O\big(e^{-\pi x^2}\big) = O(1)$. On the other hand, if $x = 0$, $F(z) = e^{-\pi y^2} O\Big(\int_{-\infty}^{\infty} e^{-\pi t^2 + 2\pi t y}\, dt\Big) = O(1)$, just changing variables $t \mapsto t + y$.

Let us say (or dream) that maximum modulus principle can be applied in each quadrant of $\mathbb{C}$ (here it is the gap). Then as F is bounded in the real and imaginary axes, we have that F is bounded in each quadrant and consequently it is constant (Liouville's theorem). For each $z = x$ real, we have const. $= e^{\pi x^2} \widehat{f}(x)$ and eliminating $\widehat{f}$ and taking the inverse transform, $f(x) =$ const. $e^{-\pi x^2}$. □

If you are curious about the gap but not so curious to go to the bibliography you will like to know that maximum modulus principle is not true in full generality for a quadrant but it is true if we assume that the growth is under control. The proxy of the maximum modulus principle for unbounded regions is the *Phragmén-Lindelöf principle*.

Even third grade kids know that uncertainty principle is something of quantum physics with philosophical consequences. How is that? Does the third grade syllabus include now Fourier transforms? You can find a big pile of books with the nonsensical things said in the media about quantum physics and repeated by everybody but here they have a point. The epitome of the uncertainty principle is *Heisenberg's uncertainty principle* and somebody can accuse engineers and mathematicians of hijacking the term. In quantum physics the space of moments is related to the Fourier transform and one of the mathematical statements of this principle is the inequality in the following result. To be precise, actually W. Heisenberg did not state it in this way, he obtained the idea working with Fourier series of an anharmonic oscillator [SR01] and presented it as a lack of commutativity [Hei27], as we shall do later.

Theorem 1.2.9 (Heisenberg inequality). *For any $a, b \in \mathbb{R}$ and $f \in L^2(\mathbb{R})$*

$$16\pi^2 \int_{-\infty}^{\infty} (x-a)^2 |f(x)|^2\, dx \cdot \int_{-\infty}^{\infty} (\xi - b)^2 |\widehat{f}(\xi)|^2\, d\xi \ge \|f\|_2^4 \tag{1.81}$$

if the integrals exist. Moreover the equality is reached if and only if f is a constant multiple of $e(bx)e^{-c(x-a)^2}$ for some $c > 0$.

Except in the trivial case $f = 0$, we can always assume under scaling that $\|f\|_2 = 1$, hence $|f(x)|^2\, dx$ and $|\widehat{f}(\xi)|^2\, d\xi$ are probability measures. The minimum of the left hand side is reached for the expectations $a = \int x|f(x)|^2\, dx$ and $b = \int \xi |\widehat{f}(\xi)|^2\, d\xi$. In this situation, Heisenberg inequality states that the product of the variances is always greater than $1/16\pi^2$. This is close to the quantum mechanics interpretation that we shall briefly consider later. Translating at the same time f and $\widehat{f}$ one can always force $a = b = 0$ (see the proof).

Proof. Define $f(x) = g(x-a)e(bx)$. It is easy to see that $\widehat{f}(\xi) = \widehat{g}(\xi - b)e\big(a(b-\xi)\big)$ and then Heisenberg inequality is equivalent to

$$16\pi^2 \int_{-\infty}^{\infty} x^2 |g(x)|^2\, dx \cdot \int_{-\infty}^{\infty} \xi^2 |\widehat{g}(\xi)|^2\, d\xi \ge \|g\|_2^4, \tag{1.82}$$

i.e., we can restrict ourselves to the case $a = b = 0$.

Assume firstly that g is in the Schwartz space of rapidly decreasing functions to avoid convergence problems. Integrating by parts

$$\int_{-\infty}^{\infty} |g(x)|^2\, dx = -\int_{-\infty}^{\infty} x\big(|g(x)|^2\big)'\, dx = -2\Re \int_{-\infty}^{\infty} x g(x)\overline{g'(x)}\, dx. \tag{1.83}$$

Cauchy-Schwarz inequality and Parseval identity (1.74) prove

$$\|g\|_2^4 \le 4\int_{-\infty}^{\infty} x^2 |g(x)|^2\, dx \cdot \int_{-\infty}^{\infty} |g'(x)|^2\, dx = 4\int_{-\infty}^{\infty} x^2 |g(x)|^2\, dx \cdot \int_{-\infty}^{\infty} |\widehat{g'}(\xi)|^2\, d\xi \tag{1.84}$$

and it is enough to use (1.69). The inequality saturates (becomes an equality) if and only if xg and g' are proportional and solving a simple ODE this is the same as saying that g is a Gaussian function.

To extend the proof when g is not in the Schwartz class it is enough to approximate by a sequence $\{g_n\}_{n=1}^\infty$in this space in such a way that $\lim \int_{-\infty}^{\infty}(1+\xi^2)|\widehat{g}(\xi)-\widehat{g}_n(\xi)|^2\,d\xi = 0$. See the details in [DM72, §2.8]. □

If you talk to your physicist friend and proudly mention you now know that Heisenberg uncertainty in quantum mechanics is a property of the Fourier transform the physicist will reply "Of course it is not, it is about the noncommutativity of two operators" and will be able to show you tons of books and web pages in which the *commutation relation*

$$[\mathbf{x},\mathbf{p}] = i\hbar \qquad \text{or} \qquad [\mathbf{x},\mathbf{p}] = i \tag{1.85}$$

is printed with big types and boxed. We focus on the second which is not other than the first one written in *natural units* in which $\hbar = 1$. After asking and asking your friend, you will figure out that at least in the 1D setting the *position* $\mathbf{x}$ and the *momentum* $\mathbf{p}$ are the operators acting on $\Psi : \mathbb{R} \longrightarrow \mathbb{C}$ given by

$$\mathbf{x} : \Psi \longmapsto x\Psi \qquad \text{and} \qquad \mathbf{p} : \Psi \longmapsto -i\frac{d}{dx}\Psi. \tag{1.86}$$

Strange, isn't it? Then the commutation relation (1.85), with $[\cdot,\cdot]$ the commutator, as usual, turns to be rather easy:

$$[\mathbf{x},\mathbf{p}]\Psi = \mathbf{x}(\mathbf{p}\Psi) - \mathbf{p}(\mathbf{x}\Psi) = x\Big(-i\frac{d}{dx}\Psi\Big) + i\frac{d}{dx}(x\Psi) = i\Psi. \tag{1.87}$$

It goes without saying that the giant step in quantum mechanics is to model positions and momentum with operators, not checking almost trivial calculations like this.

A mantra for physicists is that for two Hermitian operators A and B a commutation relation of the form $[A,B] = iC$ gives the uncertainty relation

$$\Delta A\,\Delta B \ge \frac{1}{2}|\langle C\rangle|. \tag{1.88}$$

They say, if A and B commute $C = 0$ and you can measure A and B with arbitrarily small *uncertainties* ΔA and ΔB, but if they do not, then there is a limit in the precision of the simultaneous measurement of A and B. For position $\mathbf{x}$ and momentum $\mathbf{p}$ in natural limits the lower limit for $\Delta A\,\Delta B$ is $1/2$ and in "normal" units $\hbar/2$, around $5.27\cdot 10^{-35} m^2kg/s$, something infinitesimal for our daily experience but relevant at atomic scale. Wait, wait, wait… we urgently need a physics-mathematics translator. What is the uncertainty ΔA of an operator? What are those funny vertical lines and brackets around C in (1.88)?

First of all, physicists apply their Hermitian operators (they say *observables*) to functions very often denoted by Ψ (they say *wave functions*) usually with $|\Psi|^2$ a probability density function, although it is not relevant for (1.88). To fix ideas you can think in

$\Psi : \mathbb{R} \longmapsto \mathbb{C}$, as before, or $\Psi : \mathbb{R}^n \longmapsto \mathbb{C}$. Once a wave function (a *state*) is fixed, the average (expectation) of a Hermitian operator A acting on it and its uncertainty are defined by

$$\langle A\rangle = \int \overline{\Psi} A\Psi \qquad \text{and} \qquad \Delta A = \Big(\int \overline{\Psi}(A - \langle A\rangle \mathrm{Id})^2\Psi\Big)^{1/2}. \tag{1.89}$$

Here the square means the square of the operator. At least formally, ΔA is a kind of standard deviation and the name *uncertainty* is not unjustified. The vertical lines in (1.88) are the absolute value (modulus) of the complex number. Physicists usually are not very worried in this context about conditions assuring the existence of $\langle A\rangle$, ΔA, etc. and we stick to this philosophy in the next result.

Proposition 1.2.10. *Taking as granted the existence of the quantities appearing in the proof, the relation* (1.88) *holds for A and B Hermitian operators with* $[A, B] = iC$.

Proof. Changing A and B by $A - \langle A\rangle\mathrm{Id}$ and $B - \langle B\rangle\mathrm{Id}$, we can assume $\langle A\rangle = \langle B\rangle = 0$. By Cauchy-Schwarz inequality

$$\Delta A\, \Delta B = \Big(\int |A\Psi|^2 \int |B\Psi|^2\Big)^{1/2} \ge \int \overline{A\Psi} B\Psi = \int \overline{\Psi} AB\Psi. \tag{1.90}$$

The first and the last equalities follow because A is Hermitian. Clearly

$$\int \overline{\Psi} AB\Psi = \frac{1}{2}\int \overline{\Psi}(AB + BA)\Psi + \frac{i}{2}\int \overline{\Psi} C\Psi. \tag{1.91}$$

As $AB + BA$ and C are Hermitian, the integrals are real and taking absolute values the result follows. □

Very easy, right? It is nevertheless unclear the relation of (1.88) with "our" Heisenberg inequality in Theorem 1.2.9. Take $A = \mathbf{x}$ and $B = \mathbf{p}$ as in (1.86) and write f instead of Ψ. For the sake of simplicity we assume $\langle A\rangle = \langle B\rangle = 0$. Then we have

$$(\Delta A)^2 = \int x^2 |f(x)|^2\, dx, \qquad \langle C\rangle = \|f\|_2^2 \tag{1.92}$$

and by Parseval identity and the properties of the Fourier transform

$$(\Delta B)^2 = -\int \overline{f} f'' = -\int \overline{\widehat{f}(\xi)}(2\pi i\xi^2)\widehat{f}(\xi)\, d\xi = 4\pi^2 \int \xi^2 |\widehat{f}(\xi)|^2\, d\xi. \tag{1.93}$$

Then (1.88) gives Theorem 1.2.9 for $a = b = 0$. As shown in its proof, a change in the function of the form $f \mapsto f(x - c_1)e(c_2 x)$ allows to drop the condition $\langle A\rangle = \langle B\rangle = 0$ and put general constants.

Suggested Readings. The idea of that Fourier series truncated to frequencies less than δ^{-1} skip details of size δ very seldom appears in mathematical books because it is difficult to state it as a theorem. It is anyway something important to keep in mind in signal processing. In [DS89] there are some mathematical statements that capture the practical idea. For the quantum mechanics part, there are a lot of basic books. My advice for a mathematical reader is to escape from those that barely contain formulas and be aware that quantum mechanics outreach literature is very often misleading. I find a notes written by B. Zwiebach [Zwi13] very interesting (and free). An old and interesting book is [LL58]. Perhaps the most pleasant for a hard-core mathematician is [GP90].

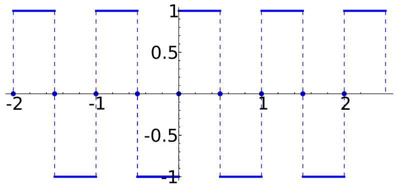
1.2.4 Gibbs phenomenon

After the ideas and graphics above, we infer that for regular functions a limitation of the frequencies to size less than δ^{-1} causes that we miss the function by something like δ and not better in general. Clearly we cannot perform a uniform approximation of a discontinuous function by continuous functions so the regularity plays a role. We can hope anyway that if we have a single jump discontinuity, except for the δ suggested by the uncertainty principle, the truncated Fourier series remains in the gap between the left and the right branches with good approximations when we are δ far apart from the discontinuity. We can hope or conjecture whatever we want and reality can be a slap in the face.

Before entering into details perhaps you are wondering "Why do we care now about discontinuous functions? Is it a mathematicians thing?" Not exactly, if you press a button you have a discontinuous signal, for instance switching on a fluorescent lamp is like considering a single discontinuity in the voltage source in (1.21). On the other hand a digital signal is plenty of jumps. One may say "OK, this is just an idealization because everything in Mother Nature is continuous, nothing can go from 0 to 1 in a short period of time without taking the middle values". Probably many physicists would not agree and even if we admit this claim, we work with mathematical models and a discontinuous function is a good model for something that changes so quickly from a value to another very different value that we cannot detect it. Another natural question is why we bother to know whether the Fourier series approximately stays in the gap when it faces a jump discontinuity. If not, it could introduce serious artifacts. We expect that if we pass from light green to dark green a scarlet tone should not appear in the middle whatever method we use to represent the transition.

The really bad news for applications is that Fourier series and integrals are strongly non local. It means that a bad point "contaminates" the global behavior. Let us focus on the 1-periodic extension f of the sign function $\operatorname{sgn}(x)$ in the interval $[-1/2, 1/2]$, which is just the double of the square wave (1.58).

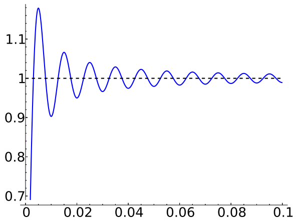
(1.94) $f =$



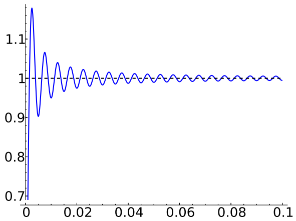
$$f(x) = \begin{cases} 1 & \text{if } \lfloor 2x \rfloor \text{ is even} \\ 0 & \text{if } 2x \in \mathbb{Z} \\ -1 & \text{if } \lfloor 2x \rfloor \text{ is odd} \end{cases}$$

where $\lfloor x \rfloor = \max\{n \le x \, : \, n \in \mathbb{Z}\}$. The Fourier series of f has coefficients that decay as $1/n$ because it is not possible to get rid of the boundary terms when integrating by parts. So we have a non absolutely convergent Fourier series in any compact subset of $(-1/2, 0)$ and $(0, 1/2)$ although f is regular, even constant, in these intervals.

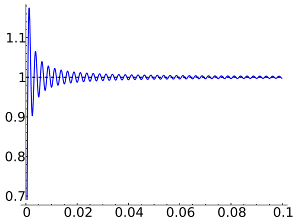
In principle, this does not contradicts our claim, a conditionally convergent series can converge very quickly and uniformly in some regions but the following details of the graph of the truncated Fourier series for x in the left part of $(0,1/2)$ terminates our hope. The convergence is lame, the series overshoots wildly the function and it is not approximately confined to the gap between the branches at both sides of the singularity. The Fourier series and the function are odd, so the behavior to the left is exactly symmetric.



frequencies< 100



frequencies< 200



frequencies< 400

When we take frequencies less than δ^{-1} it seems that the interval $[0,\delta]$ contains a lump of size like 0.2 independently of how small is δ. Let us write and prove it as a mathematical result.

Proposition 1.2.11. *For f as before, we have*

$$\lim_{N\to\infty}\sup_{|x|<1/(2N+1)}\big|f(x)-S_Nf(x)\big| = \int_{-1}^{1}\frac{\sin(\pi x)}{\pi x}\,dx - 1 = 0.17897974\dots$$

There is nothing special about $\operatorname{sgn}(x)$. Think for instance that for $g:\mathbb{R}\longrightarrow\mathbb{C}$ regular, $g(x)+\sum_{j=1}^{J}\lambda_j\operatorname{sgn}(x-\mu_j)$ is the generic form of a regular real variable function except for a finite set of jump discontinuities. In the case of $\operatorname{sgn}(x)$ we have a lump of size $0.17\dots$ for a jump of size 2. Scaling these values, in general the lumps are always like a 9% of the jump, this is called *Gibbs phenomenon*.

Proof. By the symmetry, we only consider the case $x\ge 0$. For any f we know that $S_Nf = D_N * f$ and in our case we have

$$S_Nf(x) = -\int_{x}^{x+1/2} D_N(t)\,dt + \int_{x-1/2}^{x} D_N(t)\,dt.$$

Substituting the explicit formula for D_N and using its parity, this is

$$S_Nf(x) = \int_{-x}^{x}\frac{\sin((2N+1)\pi t)}{\sin(\pi t)}\,dt - \int_{-x-1/2}^{x-1/2}\frac{\sin((2N+1)\pi t)}{\sin(\pi t)}\,dt.$$

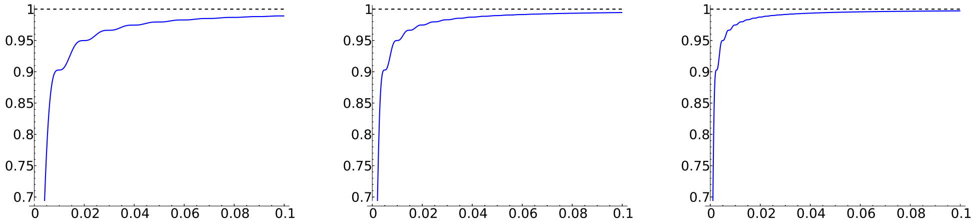
Integrating by parts (note that x is small), the second integral is $O\big(N^{-1}\big)$.

Clearly the supremum of the first integral is reached for $x=1/(2N+1)$ because the function under the integral is positive. Finally,

$$\int_{-1/(2N+1)}^{1/(2N+1)}\frac{\sin((2N+1)\pi t)}{\sin(\pi t)}\,dt = \int_{-1}^{1}\frac{\sin(\pi x)}{(2N+1)\sin(\pi x/(2N+1))}\,dx$$

is just a change of variables and the denominator tends to πx. □

It seems that Gibbs phenomenon is a serious drawback in the applicability of Fourier analysis for discontinuous signals but it is not the end of the world because there are localized versions of Fourier analysis (for instance the windows or the wavelets that we shall treat in §3.1) and because we can reduce the effect of the singularities with a regularization (this is related to the filters that we shall consider in the digital setting in §2.2.3). For instance, compare the following figures with the previous ones, the only difference is replacing $S_N f$ by $\widetilde{S}_N f$ as in Theorem 1.2.5.



frequencies< 100 frequencies< 200 frequencies< 400

Although we still have large errors near zero (as it must be because we are approximating a discontinuous function), we do not see lumps, the general aspect is less shaky. This is important, think for instance in medical imaging. The big wobbles in the first case would transform in stains between sharp transitions, different tissues, that could be misinterpreted as a tumor while the second approach would give a blur smoother change between tissues.

Suggested Readings. Again, Gibbs phenomenon belongs to basic Fourier analysis and it is nicely explained in the monographs mentioned before, for instance [Kör88] and [DM72].

1.2.5 More flavors of harmonic analysis

Why cosine and sine? Because, as we have seen, basic electronic circuits lead to these functions. Another not technologically based reason is that for instance, the physiology of hearing introduces some filtering on the frequencies, then if we want to understand or to simulate hearing (yes, with technology!) it seems handy to read the content on each frequency through Fourier analysis[6].

In other situations, it might be more convenient to analyze in terms of other non trigonometrical functions having special properties. The bundle of techniques and problems derived from the analytical decomposition of a function into something that we would call "pure tones" or even better "harmonics", it is called *harmonic analysis*. What is really a harmonic? The honest answer is whatever you find convenient. Here we consider harmonic analysis somewhat related to symmetries to illustrate that there is something beyond the classic Fourier series and integrals.

Symmetry sounds to group theory, then we are going to consider mainly groups. Let us start with a finite abelian group G. The functions $f : G \longrightarrow \mathbb{C}$ that we want to analyze

[6] In [MI11, §4.1] we read "*Fourier analysis is like a glass prism, which splits a beam of light into frequency components corresponding to colors*".

can be considered as ordered finite lists of real numbers. Why do we want to analyze them? Aren't they already very simple? The case treated in (1.32) corresponds to $\mathbb{Z}_N$, the classes of integers modulo N, because $f(n) = f(n + kN)$ and it is important in digital signal processing. One may imagine that the digital world approximates the real (analog) world when N grows and suspect that it is the only interesting case but the fact is that even small groups are relevant in practice. Believe or not, when you store a photo in your cellular or your computer, harmonic analysis in $\mathbb{Z}_8 \times \mathbb{Z}_8$ is applied quite a number of times (if you are curious about it, continue reading these notes beyond the first chapter).

For G finite and abelian the analogue of cosine and sine unified in the complex exponential are the *characters*. These are maps

$$\chi : G \longrightarrow \mathbb{C}^* \qquad \text{with} \quad \chi(g_1 g_2) = \chi(g_1)\chi(g_2) \quad \forall g_1, g_2 \in G, \tag{1.95}$$

where $\mathbb{C}^* = \mathbb{C} - \{0\}$, which is a multiplicative group and then we are just saying with formulas that χ is a *homomorphism*, a map preserving the group structure. It is easy to see that $\chi(g)$ is always a $|G|$-th root of unity $\chi(g) = e(k_{\chi,g}/|G|)$. Let $\widehat{G}$ denote the set of characters. It inherits a group structure from $\mathbb{C}^*$. For $G = \mathbb{Z}_N$ we have

$$\widehat{G} = \{\chi_0, \chi_1, \dots, \chi_{N-1}\} \qquad \text{with} \quad \chi_k(\overline{n}) = e(kn/N). \tag{1.96}$$

The usual notation in $\mathbb{Z}_N$ is additive then we have $\chi(\overline{n} + \overline{m}) = \chi(\overline{n})\chi(\overline{m})$. Note that $|\widehat{G}| = |G|$. This holds in general [Ter99], you can believe it promptly for instance appealing to the classification of finite abelian groups that allows to write G as a direct product of $\mathbb{Z}_N$'s and the characters are constructed as products of the characters of the direct factors.

The important point in this more general context, is that we have something like an analogue of (1.33) that was the precursor of (1.37).

Lemma 1.2.12. *If G is a finite abelian group the following* orthogonality relations *hold true for every $\chi_1, \chi_2 \in \widehat{G}$*

$$\frac{1}{|G|} \sum_{g \in G} \overline{\chi}_1(g)\chi_2(g) = \begin{cases} 1 & \text{if } \chi_1 = \chi_2, \\ 0 & \text{if } \chi_1 \neq \chi_2. \end{cases} \tag{1.97}$$

One could proof this with an explicit construction of the characters as indicated above but there is a shorter, more elegant and generalizable way to proceed.

Proof. As $\chi_1(g)$ is a root of unity $\overline{\chi}_1(g)\chi_1(g) = 1$ and the case $\chi_1 = \chi_2$ becomes trivial. If $\chi_1 \neq \chi_2$ let $h \in G$ such that $\chi_1(h) \neq \chi_2(h)$. We have

$$\sum_{g \in G} \overline{\chi}_1(g)\chi_2(g) = \sum_{g \in G} \overline{\chi}_1(hg)\chi_2(hg) = \overline{\chi}_1(h)\chi_2(h) \sum_{g \in G} \overline{\chi}_1(g)\chi_2(g). \tag{1.98}$$

The first equality follows because $g \mapsto hg$ permutes the elements of G. As $\overline{\chi}_1(h)\chi_2(h) = \chi_2(h)/\chi_1(h) \neq 1$, the sum must vanish. □

With an elementary calculation, from (1.97) we get a generalization of (1.32).

Theorem 1.2.13. *If G is a finite abelian group, any $f : G \longrightarrow \mathbb{C}$ can be written as*

$$f(g) = \sum_{\chi \in \widehat{G}} a_\chi \chi(g) \qquad \textit{with} \quad a_\chi = \frac{1}{|G|} \sum_{g \in G} f(g) \overline{\chi}(g). \tag{1.99}$$

If we think about a topological infinite abelian group as a kind of limit of finite groups, $|G|^{-1} \sum_{g \in G}$ may be interpreted as an equidistributed measure. It can be proved that for G locally compact there exists a formalization of this concept, it is the *Haar measure* μ that verifies $\mu(gS) = \mu(S)$ for every $g \in G$ and $S \subset G$ a Borel set. It is unique except for scaling it multiplying by a constant. For G abelian and locally compact the *characters*, in this more general context are defined as continuous bounded homomorphisms $\chi : G \longrightarrow \mathbb{C}^*$ (in the finite case the continuity was obvious because everything is continuous with the discrete topology). If G is compact we can state the orthogonality relations as

$$\int_G \overline{\chi}_1 \chi_2 \, d\mu = \begin{cases} 1 & \text{if } \chi_1 = \chi_2, \\ 0 & \text{if } \chi_1 \neq \chi_2, \end{cases} \tag{1.100}$$

choosing the scaling in such a way that $\mu(G) = 1$. In this situation, (1.99) holds true with $a_\chi = \int_G f \overline{\chi} \, d\mu$ under conditions assuring the convergence. If G is not compact it may happen $\mu(G) = \infty$ and a kind of suspicious "infinite scaling" appears to be needed here. Actually it is possible to scale at the same time the sums over G and $\widehat{G}$ to keep the result. To sum up, if G is abelian and locally compact, we have [Kat68]

$$f(g) = \int_{\widehat{G}} \widehat{f}(\chi) \chi(g) \, d\nu \qquad \text{with} \quad \widehat{f}(\chi) = \int_G f(g) \overline{\chi}(g) \, d\mu \tag{1.101}$$

where μ and ν are the Haar measures of G and $\widehat{G}$ respectively, with a convenient normalization, and f is assumed to be continuous with $f \in L^1(G)$ and $\widehat{f} \in L^1(\widehat{G})$ to assure the convergence.

The formula (1.101) embodies the totality of the flavors of harmonic analysis typically seen in mathematical degrees: If G is finite then μ and ν are scaled counting measures and we have (1.99); if $G = \mathbb{T}$ then $\widehat{G} = \{e(nx)\}_{n \in \mathbb{Z}} \cong \mathbb{Z}$, μ is the Lebesgue measure and ν the counting measure and we have (1.34); finally if $G = \mathbb{R}$ then $\widehat{G} = \{e(\xi x)\}_{\xi \in \mathbb{R}} \cong \mathbb{R}$, μ and ν are the Lebesgue measure and we have (1.36).

Is it still possible to generalize (1.99) or (1.101) to nonabelian groups? The answer is yes but it requires to deal with more difficult concepts. We restrict ourselves to finite nonabelian groups G with an ending comment about the compact case.

The previous approach completely fails because in a nonabelian group we have in general very few characters defined as before. The surrogate concept is that of unitary representation. With a little narrow sighted definition adapted to our context, a *representation* is a map π that associated to every $g \in G$ a complex non singular square matrix $\pi(g)$ preserving multiplication, $\pi(g_1)\pi(g_2) = \pi(g_1 g_2)$ and it is called *unitary* if $\pi(g)$ is a unitary matrix. The dimension d_π of a representation indicates the size of the matrix. A very important and not so simple concept is that of *irreducible representation*. It means that it is not possible to find a proper subspace of $\mathbb{C}^{d_\pi}$ that remains invariant by $\pi(g)$ simultaneously for every $g \in G$. Given a representation π we can construct another given by a change of basis $C^{-1}\pi(g)C$ where C is a constant matrix. We are only interested in *non-equivalent representations*, those not related by a change of basis. Just to mention a nontrivial example of representation, the following table defines a unitary irreducible representation of the permutation group S_3:

(1.102)

g	Id	$(1,2,3)$	$(1,3,2)$	$(1,2)$	$(2,3)$	$(1,3)$
$\pi(g)$	$\left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$	$\frac{1}{2}\left(\begin{smallmatrix} -1 & -\sqrt{3} \\ \sqrt{3} & -1 \end{smallmatrix}\right)$	$\frac{1}{2}\left(\begin{smallmatrix} -1 & \sqrt{3} \\ -\sqrt{3} & -1 \end{smallmatrix}\right)$	$\frac{1}{2}\left(\begin{smallmatrix} -1 & \sqrt{3} \\ \sqrt{3} & 1 \end{smallmatrix}\right)$	$\left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$	$\frac{1}{2}\left(\begin{smallmatrix} -1 & -\sqrt{3} \\ -\sqrt{3} & 1 \end{smallmatrix}\right)$

In the context of finite nonabelian groups, one defines $\widehat{G}$ to be a maximal set of non-equivalent unitary irreducible representations. It can be proved that it is finite, in fact one has the curious relation [Ter99]

$$\sum_{\pi \in \widehat{G}} d_\pi^2 = |G|. \tag{1.103}$$

For instance, for S_3 we have the previous representation with $d_\pi = 2$, the trivial one $\pi(g) = (1)$ and $\pi(g) = (\operatorname{sgn}(g))$. The equality $2^2 + 1^2 + 1^2 = 3!$ assures that any other irreducible representation is one of these after a change of basis.

After all of these definitions, the generalization of (1.99) is

$$f(g) = \sum_{\pi \in \widehat{G}} d_\pi \operatorname{Tr}\big(\widehat{f}(\pi)\pi(g)\big) \qquad \text{with} \quad \widehat{f}(\pi) = \frac{1}{|G|}\sum_{g \in G} f(g)\pi(g)^\dagger. \tag{1.104}$$

Here $\pi(g)^\dagger$ means the transpose conjugate of $\pi(g)$ and note that the "Fourier coefficients" $\widehat{f}(\pi)$ are now matrices. This formula actually generalizes (1.99) because for an abelian group the irreducible representations have $d_\pi = 1$ and then $\widehat{G}$ becomes the group of characters. If G is not finite but it is compact, (1.104) still holds changing the formula for $\widehat{f}(\pi)$ by the integral $\int_G f\pi^\dagger$ where one employs a conveniently normalized Haar measure to integrate on G.

In a compact Riemannian manifold M, there is a natural second order operator, the *Laplace-Beltrami operator* and the general theory assures that it possesses and orthonormal discrete system of eigenfunctions $\{\phi_j\}_{j=1}^\infty$ and any function $f \in L^2(M)$ can be expanded as a spectral Fourier series

$$f = \sum a_j \phi_j \qquad \text{with Fourier coefficients} \quad a_j = \int_M \overline{\phi}_j f. \tag{1.105}$$

For $M = \mathbb{T}$ the Laplace-Beltrami operator is $-d^2/dx^2$ and the eigenfunctions are the complex exponentials $e(nx)$. For a (semisimple) compact *Lie group*, a (Riemannian) manifold having a smooth group structure, this description and (1.104) coincide, meaning that the entries of the matrices of $\pi(g)$ are actually orthogonal eigenfunctions of the Laplace-Beltrami operator and so one can rearrange the illusory limit of (1.104) as $|G| \to \infty$ to get (1.105).

Suggested Readings. The finite abelian and nonabelian case is treated in [Ter99], a masterpiece in exposition that includes also some applications linked to more advanced topics. In [DM72] there is a detailed analysis of the compact Lie group SO(3) from scratch. Although [Zee16] does not address topics on harmonic analysis it is a good source to learn about representations, Lie groups and why physicists care about them. A classic for basic spectral theory and generalized Fourier expansions with applications is [CH53].

1.3 Problem set and challenges

Note: These exercises were created for the assessment during the master course at UAM *Wavelets and signal processing* 2017/2018.

PROBLEM SET 1

Problems

1) Consider the driven harmonic oscillator ruled by the equation $x'' + \omega^2 x = \sin(\lambda t)$ with $\omega, \lambda > 0$ under the initial conditions $x(0) = 0$, $x'(0) = 1$. Compute the explicit solution $x_\omega(t)$ for $\omega \neq \lambda$ and find $\lim_{\omega\to\lambda} x_\omega(t)$. Check that the function resulting from this limit fulfills the equation with $\omega = \lambda$.

2) Prove the first formula of (1.51) in §1.2.1.

3) Solve the mathematical part of the experimental challenge *The strange beat*. You can take as a guidance the sample file `interference.mp3` in the complementary material web page if you do not do the experiment by yourself.

4) Check the formula (1.104) for the function $f : S_3 \longrightarrow \mathbb{C}$ defined by $f(\mathrm{Id}) = f\big((2,3)\big) = 1$ and $f(g) = 0$ for $g \neq \mathrm{Id}, (2,3)$.

5) Reproduce the consequence of the uncertainty principle for f and f_δ as in §1.2.3 with f_δ at least C^1. Namely, for $\delta = 0.1$ find an explicit even nonnegative function $f_\delta \in C^1$ such that $f(x) = f_\delta(x)$ for $|x| \notin [1-\delta/2, 1+\delta/2]$ and $\int f_\delta = 1$. For this function, find $x_\epsilon > 0$ for $\epsilon = 1/2$, $1/4$ and $1/8$ such that

$$\frac{|\widehat{f}(x) - \widehat{f_\delta}(x)|}{\sup_{t>x} |\widehat{f}(t)|} \le \epsilon \qquad \text{for every } 0 \le x \le x_\epsilon,$$

trying to get x_ϵ as large as possible.

Notes and hints

1) "Explicit" means involving only elementary functions. If your favorite method to solve the ODE involves integrals, you must compute them. The limit must also be computed, it is not valid to guess the result by some kind of continuous dependance on the parameters.

Note that x_ω is bounded for $\omega \neq \lambda$ but the limiting solution is not because of the resonance.

2) The second formula of (1.51) is easy integrating by parts, right? A possible approach is to use a discrete analogue in the first formula. There are much shorter trickier methods that probably you can improvise. In any approach that comes to my mind one uses in one way or another a relation with the Dirichlet kernel.

3) If this exercise seems almost trivial to you, do not worry. Actually, it is. It only involves high schools mathematics but the experimental meaning is somewhat unexpected and something to keep in mind by anybody interested in signal processing.

4) You can grab from §1.2.5 that the (non-equivalent) unitary irreducible unitary representations of S_3 are given by (1.102) and by two easy one-dimensional representations mentioned there. I assume you remember the usual notation for permutations, if not, you should look it up in any book of basic algebra. This exercise is very short and you can solve it in few lines once you know what is talking about. Its main purpose is to understand the formula (1.104). If you already do, it will take less than three minutes (but surely you spent something more when you studied it).

5) It goes without saying that f is the characteristic function of the interval $[-1,1]$. If you construct f_δ without taking into account the properties of the Fourier transform, you can enter in a very big mess with the calculations. Does *convolution* ring a bell? I do not want to be very picky with “as large as possible”, it is just to avoid trivial solutions.

Experimental challenge: **The strange beat**

Experimental part

You have to generate simultaneously two “pure tones” (say sine waves) in the audible range having very close frequencies but not exactly equal (Can you guess how their interference sounds before doing the experiment?). Try different frequencies and spacing and observe the effect on the result.

Now some practical help. You need a pure tone generator that allows to adjust the frequency. How to get it? Several possibilities that you can explore (and I did not check) in case you want to be creative are: learn how to use the commands for audio generation of `octave` or `matlab`; download an app for your cell phone for simple tone generation; look for online applets on the internet... If you prefer to observe the phenomenon with no machine assistance (this is probably advanced), use two identical tuning forks and modify the frequency of one of them putting small impurities like duct tape or modeling clay.

In case you prefer step-by-step instructions, here it is a checked possibility: install `audacity` in your computer (it is very light, easy to install and multi-platform). Run it and close the pop-up help window. In the menu `Generate>Tone` choose `sine`, some audible frequency, amplitude 1, whatever duration you like and press OK. Now add a new track with `Tracks>Add_new>audio_track` (or `mono_track`). Repeat again `Generate>Tone` but this time with a close frequency. If you want to save the result in an audio file, use `File>Export_audio` and choose your favorite format.

Mathematical part

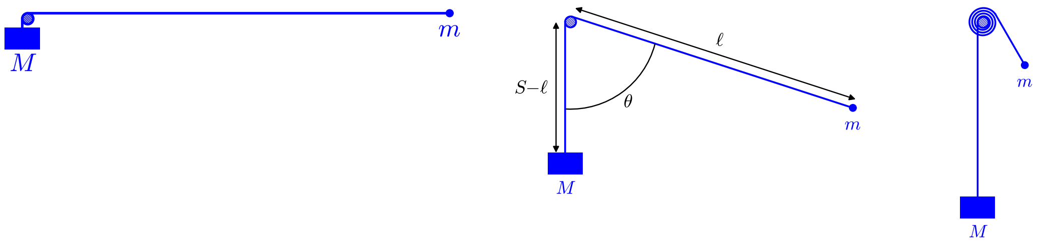
Get a formula expressing the interference of the waves in a way that explains the result of the experiment. From this formula, predict the time lapse in seconds between two consecutive pulses (beats) when the frequencies are in hertz.

The file `interference.mp3` in the complementary material web page corresponds to waves of frequencies $700\,Hz$ and $703\,Hz$. Do you think this is coherent with your last formula?

Experimental challenge: **A not so free fall**

Experimental part

You have to construct a pulley with a string of length S and a not very thick stick. In the right extreme hang a light mass m and in the left extreme a much heavier mass M. Start with M in the upper position and m with the the string in horizontal position (see the figures below). Release the system to let m oscillate as a pendulum and M fall.



Starting position

Releasing the system

Final situation

The surprise is that with reasonable magnitudes, M stops abruptly because m gives many $360°$ turns and wraps up the string around the stick producing a big friction.

In tests I chose a key as light object and a ball bearing as heavy object. I used a string of something less than 2 m. I tried heavier objects and they never reached the floor but I was reluctant to try very heavy objects because my string was very thin.

Mathematical part

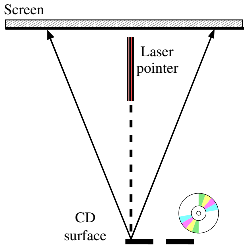
Use Euler-Lagrange equations with the coordinates ℓ and θ displayed in the second figure to find the differential equations for the evolution of the system with this coordinates assuming no friction and the radius of the pulley 0. Recall that the Lagrangian is $L = \frac{1}{2}Mv_M^2 + \frac{1}{2}mv_m^2 - Mgh_M - mgh_m$ with v velocity and h height.

Write the equations in the form $\ell'' = \dots$, $\theta'' = \dots$ and take formally the limit of them when $m/M \to 0$ (this is physically admissible because m is light and M is heavy) to get a simplified system of equations. From it, find explicitly $\ell(t)$ and get a differential equation for θ not involving ℓ. Try to proof, at least numerically with an example, that $\theta \to -\infty$ when $\ell \to 0^+$. This explains mathematically the experiment because it is known that the friction increases exponentially with the number of turns (Capstan equation) and we have seen that this number goes to ∞, hence the system must stop before $\ell = 0$.

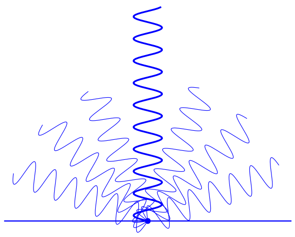
Experimental challenge: **Grooving on the waves**

Experimental part

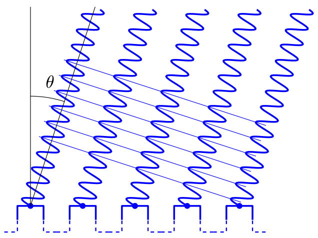
You have to point a laser (a laser pen for presentations does the job) perpendicularly to the reflecting surface of a CD (with a DVD or Blue-ray the result is not the same). In principle it seems that the reflected ray approximately comes back to the laser pen but putting a screen (a white paper paper on a box) we observe bright spots apparently corresponding to rays reflected in weird directions.



Scheme of the experiment



Huygens principle



Bright direction

Huygens principle asserts that when a wave reach a point, it becomes a source of waves in every direction (second figure). What we see in reflection is an interference of these waves. The groove of the CD separates in some way the influence of the plateaus and pits giving unexpected reflection directions.

Mathematical part

Assume a naive form of the Huygens principle consisting in that when the sine waves from the laser pointer reach the mean point of each plateau then new sine waves come out along each direction. In some directions, like the angle θ in the third figure, the interference is constructive, the waves are perfectly parallel with maxima on the same line, and we see a bright spot.

Under this model, find an equation for the bright directions involving the frequency of the laser light and the separation d between the plateaus. The frequency is normally indicated on the laser pointers. Use the experiment and the equation to derive numerically the value of d. If you see several bright points on your screen, check the formula and the calculated value of d with the next one.

Theoretical challenge: **Breaking up a sawtooth**

Simple waves

Motivation

The plots shown in §1.2.4 to illustrate Gibbs phenomenon for the doubled square wave suggests that the approximation is not so bad when we are not very close to the singularity. Our purpose is to analyze the situation for the sawtooth wave

$$s(t) = t - \lfloor t \rfloor - \frac{1}{2} \qquad \text{with } \lfloor t \rfloor \text{ the integral part.}$$

The coefficients of its Fourier series decay as $1/n$. Let us consider some cheap heuristics. If we think about alternating series like $1/1 - 1/2 + 1/3 - \dots$ as a model, we may guess that when we are far apart from the discontinuities the error term when approximating $s(t)$ by the Nth partial sum should be comparable to N^{-1} (the first disregarded term). On the other hand, Gibbs phenomenon seems to appear in its glory at distances comparable to N^{-1} giving an error like a constant. One could combine both claims guessing an error $O\big((Nt)^{-1}\big)$ for $1/N < t \le 1/2$. In fact it is not hard to believe in a t^{-1} decay looking the plots in §1.2.4. The interval $0 < t \le 1/N$ is still under the influence of Gibbs phenomenon, then a kind of periodic version of $(1+Nt)^{-1}$ could give the order of magnitude of the error term.

The challenge

Let $\|x\|$ be the distance of x to the nearest integer (e.g. $\|3.1\| = \|0.9\| = 0.1$). If $s(t)$ is the sawtooth wave as before, prove that

$$s(t) = -\sum_{n=1}^{N} \frac{\sin(2\pi nt)}{\pi n} + O\Big(\frac{1}{1+N\|t\|}\Big).$$

Chapter 2

Introduction to digital signals

2.1 Sampling and A/D, D/A conversion

2.1.1 Shannon sampling theorem

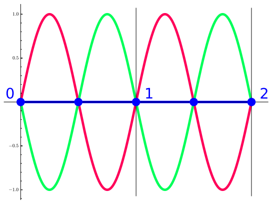
Suppose that we have a continuous time depending signal $f = f(t)$, for a mathematician a nice function $f : \mathbb{R} \longrightarrow \mathbb{R}$, and we want to treat it with magical digital processing tools. We need an A/D converter (analog-to-digital) giving a bunch of bits blocks out of the input signal. Typically it involves firstly to *sample* the signal, a discretization in time that produces values $f(t_n)$, and secondly a discretization of the sampled values, called *quantization*. In this way, the input analog signal is transformed into an ordered list of discrete values, say integers, that can be encoded with bits and stored in a file or promptly processed digitally. If the required output is analog, for instance sound through loudspeakers, we will have to reverse the process with a D/A converter (digital-to-analog). In a scheme:

$$(2.1)\qquad \begin{array}{ccccccccc} \xrightarrow[\text{input}]{\text{analog}} & \overbrace{\boxed{\text{sampling}} \to \boxed{\text{quantization}}}^{\text{A/D}} & \longrightarrow & \boxed{\begin{array}{c}\text{Digital}\\ \text{processing}\end{array}} & \longrightarrow & \boxed{\text{D/A}} & \xrightarrow[\text{output}]{\text{analog}} \\ & f(t_n) \to \quad y_n & \longrightarrow & F(y_n) & \longrightarrow & g(t) & \end{array}$$

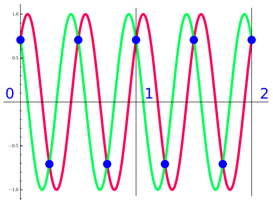
This simple scheme may present variations. For instance, an image in principle requires a function of two variables, the horizontal and vertical coordinates x and y. The images stored in our computer usually provide three channels of color (and perhaps an extra α-channel) then f is better modeled in this case as a two variables function with target in a 3-dimensional space.

A natural situation is the *uniform sampling*. In this case, if the time spacing is T the *sampling frequency* is $\nu_s = 1/T$. Again, for a mathematician uniform sampling corresponds to the sequence $\{f(n/\nu_s)\}_{n=-\infty}^{\infty}$ and it goes without saying that the sequence does not characterize the signal, even for pure tones. To convince yourself, look the graphs below. With the indicated sampling we cannot distinguish between the signals 0, $\sin(2\pi t)$

and $-\sin(2\pi t)$ in the first case, and between $\sin(4\pi t+\frac{\pi}{4})$ and $\cos(4\pi t+\frac{\pi}{4})$ in the second case. These signals become "aliases" with our methods. In general the indistinguishability of two signals under sampling is called *aliasing* and it is something that one wants to avoid imposing some conditions.



$\mathsf{T}=1/2,\quad \nu_{\mathsf{s}}=2$
$0,\quad \sin(2\pi\mathsf{t}),\quad -\sin(2\pi\mathsf{t})$



$\mathsf{T}=1/4,\quad \nu_{\mathsf{s}}=4$
$\sin(4\pi\mathsf{t}+\frac{\pi}{4}),\quad \cos(4\pi\mathsf{t}+\frac{\pi}{4})$

We are going to assume that $\widehat{f}$ is integrable and compactly supported, namely

$$\widehat{f}(\xi)=0 \qquad \text{when } |\xi|\ge B. \tag{2.2}$$

Especially in the signal processing literature, f is said to be a *band limited signal*. It is not an unnatural hypothesis. Firstly because the common channels have a limit to transmit large frequencies (for instance, even modern fiber-optic cannot transmit at a rate of 1 exabyte per second) and secondly because very often we are not interested in large frequencies and they are already filtered.

For instance, you cannot hear anything beyond 20000 Hz and then for audio applications one can assume $B=20000$ in (2.2) without losing anything, in fact for adults the upper of audible frequencies is in general smaller and probably few readers over 30 years old could hear[1] 16000 Hz and we all have problems to distinguish nearby frequencies in a much smaller range. On the other hand, according to some authors, the frequency of voice in male adults is around $85-180$ Hz and $165-255$ Hz in female adults but it does not mean that we could take safely $B=300$ because voice is a complicated signal and these values only indicate the fundamental frequency (something like the one with the largest Fourier coefficient). Bigger frequencies are needed to distinguish the different sounds. They actually "form" the voice and the most important of them (let us say, those giving peaks of the Fourier coefficients or Fourier transform) are called *formants* in phonetics and acoustic. To give a figure, $B=5000$ Hz suits for voice without noticeable loss of quality (for me).

Coming back to the topic of this subsection, Shannon sampling theorem asserts that a band limited signal can be recovered with uniform sampling if the sampling frequency is large enough. It is rather a lemma or a clever observation than a theorem unquestionably deserving this name and probably the name has been popularized by engineers. Judge by yourself. We do not pursue the best regularity hypotheses. In connection with this,

[1] If you want to try, there is an applet in `https://www.echalk.co.uk/Science/biology/hearing/HowOldIsYourHearing/resource.html` that guesses your age in terms of the result. Let me add that I am not very confident with these online tests because your computer and your loudspeakers or headphones play a role here.

note that if f is continuous, and we implicitly assume so, using the inversion formula we conclude from (2.2) that $f \in C^\infty$. In fact, it has a entire extension that is characterized by *Paley-Wiener theorem* as any real-complex analysis lover knows.

Theorem 2.1.1 (Shannon sampling theorem)**.** *Let f be a function with $\widehat{f} \in C^2$ satisfying* (2.2) *with $2B \le \nu_s$. Then*

$$f(t) = \sum_{n=-\infty}^{\infty} f(n/\nu_s)\,\mathrm{sinc}(\nu_s t - n) \tag{2.3}$$

where sinc *is as in* (1.60)*.*

In other words, given a sampling frequency ν_s, the maximal frequency that can be contained in our signal to be fully determined is $\nu_s/2$. It is called the *Nyquist frequency*. Actually this result was stated by E.T. Whittaker as an interpolation result in [Whi15] many years before C.E. Shannon in [Sha49] (and it can be traced even earlier). If you recall the discussion after (1.60), you will find natural the name *cardinal series* for (2.3).

Proof. Let g be the ν_s-periodic extension of $\widehat{f}$ restricted to $I = [-\nu_s/2, \nu_s/2]$. The Fourier expansion of g is

$$g(\xi) = \nu_s^{-1} \sum_{n=-\infty}^{\infty} e(n\xi/\nu_s) \int_I \widehat{f}(x) e(-nx/\nu_s)\, dx. \tag{2.4}$$

Note that I includes the support of $\widehat{f}$ and consequently the integral can be extended to $\mathbb{R}$ and evaluated as $f(-n/\nu_s)$ by the inversion formula. Substituting this expression for g into $f(t) = \int_I g(\xi) e(t\xi)\, d\xi$ we have

$$f(t) = \sum_{n=-\infty}^{\infty} f(-n/\nu_s)\nu_s^{-1} \int_I e\big((t + n/\nu_s)\xi\big)\, d\xi = \sum_{n=-\infty}^{\infty} f(-n/\nu_s)\,\mathrm{sinc}(\nu_s t + n) \tag{2.5}$$

and only remains to rename $n \mapsto -n$. □

There are several generalizations of this theorem [Mar91], [Wal96]. For instance, Papoulis generalized sampling theorem says that we can relax the condition $2B \le \nu_s$ to $2B \le N\nu_s$ if we can sample a vector of N filtered instances of the original signal (see the details in [Mar91, §4.2]). Essentially, if we multiply our knowledge of the signal by N, we can divide the sample rate by N.

In practice there is some uncertainty when measuring $f(n/\nu_s)$ plus a quantization error if we store it by digital means. Even if we reduce this source of errors to negligible levels, the application of Theorem 2.1.1 requires infinitely many samples to recover f and it does not seem very practical. Perhaps we can increase a lot ν_s but still the number of samples must stop at some point.

Let us say that we only sample in a certain finite interval $[-T, T]$ and we do not assume T to be very large (imagine that we have time limited access to the signal). If our technology allows us to take many samples, interpolating them, we can assume that we know quite well the signal in this interval. The mathematical question that arises is if $f|_{[-T,T]}$ determines f satisfying (2.2). The answer is "yes" because f is in fact an analytic function but the real question is how to construct such f.

The *Papoulis-Gerchberg algorithm* [Pap75] is an iterative scheme that addresses this problem. Let us denote χ_A the characteristic function of $[-A, A]$. The starting point is the known function $f_0 = f\chi_T$. Of course $\widehat{f_0}$ cannot be compactly supported then we force the condition (2.2) considering $\widehat{f_0}\chi_B$. The inverse Fourier transform of this function is band limited but fails to coincide with the known values of f, then we modify it by hand in $[-T, T]$ to obtain f_1 and we repeat the process. In one line

$$f_{n+1} = f_0 + (1 - \chi_T)(\widehat{f_n}\chi_B)^{\vee}. \tag{2.6}$$

We know that the convolution turns into a product under the Fourier transform hence we can write this as

$$f_{n+1} = f_0 + (1 - \chi_T)(f_n * h) \qquad \text{with} \quad h(t) = 2B\,\mathrm{sinc}(2Bt). \tag{2.7}$$

It is known that the algorithm converges but it is slow for some practical applications. There are several techniques to speed it up that in some way parallel those employed for ill-conditioned linear systems (cf. [Byr04]). The discretization of the algorithm allows to apply it into the setting of genuine finite samples [Byr15], [Xia93, (9)].

In connection with these ideas, note that if $\nu_s = 2B$ then

$$F(x) = \sum_{n=1}^{N} f(n/\nu_s)\,\mathrm{sinc}(\nu_s x - n) \tag{2.8}$$

satisfies $F(n/\nu_s) = f(n/\nu_s)$ for $n = 1, 2, \dots, N$ and it is always a band limited function with $\widehat{F}$ supported in $[-B, B]$ but $\widehat{F}$ is discontinuous in general.

Shannon sampling theorem is closely related to the *Poisson summation formula* which is obtained integrating the formula (1.37) for δ_P against a function f defined on $\mathbb{R}$. The Dirac comb $\sum \delta(x - n)$ evaluates f at the integers and we get

$$\sum_{n=-\infty}^{\infty} f(n) = \sum_{n=-\infty}^{\infty} \widehat{f}(n). \tag{2.9}$$

The actual rigorous proof requires to consider the 1-periodic function $F(x) = \sum_{k\in\mathbb{Z}} f(k+x)$ and expand it into Fourier series as

$$F(x) = \sum_{n=-\infty}^{\infty} \int_0^1 F(t)e^{-2\pi int}dt\; e^{2\pi inx} = \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} f(t)e^{-2\pi int}dt\; e^{2\pi inx}. \tag{2.10}$$

Taking $x = 0$ we get (2.9). To dub this proof as rigorous we must include some properties assuring that F is well defined and that it can be Fourier expanded. Clearly this holds if f is in the Schwartz class. Of course this is overkilling, a less restrictive condition is for instance $f \in C^2$ with $f(x), f'(x), f''(x) = O(|x|^{-2})$ as $x \to \infty$. In [Zyg88, II.13], the regularity conditions are relaxed a lot as stated in the following result that we do not prove here.

Theorem 2.1.2. *If $f : \mathbb{R} \longrightarrow \mathbb{R}$ is an integrable function of bounded variation such that $f(x+h)+f(x-h) \to 2f(x)$ as $h \to 0$ for every $x \in \mathbb{R}$, then* (2.9) *holds true.*

Indeed, with some modifications in (2.9) it is possible to cover not integrable cases like $f(x) = x^{-\alpha}$ [Gui41]. Poisson summation formula has striking consequences in a broad range of topics [CR17], for instance, believe or not the best known results on sphere packing are based on it [Coh17].

If we apply (2.9) to $f(x) = g(qx+a)$, we get the following generalization

$$\sum_{n=-\infty}^{\infty} g(qn+a) = \frac{1}{q}\sum_{n=-\infty}^{\infty} e(an/q)\widehat{g}(n/q). \tag{2.11}$$

For $q = 1$, $a = 0$ we recover (2.9). In principle one could derive Shannon sampling theorem from here taking $g(x) = \operatorname{sinc}(\nu_s x) f(t+x)$ with $q = \nu_s^{-1}$ and $a = -t$. In this way the left hand side of (2.11) is the right hand side of (2.3).

Note also that for $q = 1$, $a = x$ we get the Fourier expansion of F in (2.10). In general, Poisson summation formula can be used instead of Fourier expansion when the coefficients can be easily interpolated to a smooth function.

One of the most famous and interesting applications of the Poisson summation formula is the θ modular relation

$$\sum_{n=-\infty}^{\infty} e^{-2\pi\alpha n^2} = \frac{1}{\sqrt{2\alpha}}\sum_{n=-\infty}^{\infty} e^{-\pi n^2/2\alpha} \qquad \text{for} \quad \alpha > 0. \tag{2.12}$$

If α is very small the first sum is expensive from the computational point of view while the second gives readily the approximation $1/\sqrt{2\alpha}$ with exponential gain.

Suggested Readings. The whole book [Mar91] is devoted to topics around Shannon sampling theorem. In [Byr15] there are several discussions about sampling and reconstruction scattered along several chapters. See [CR17] for more about the Poisson summation formula.

2.1.2 Basic quantization

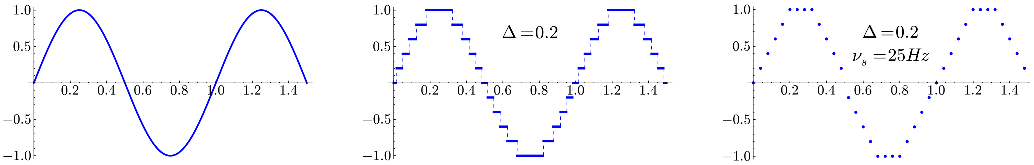
Recall the scheme (2.1). The sampled values $f(t_n)$ are real numbers and we want to approximate them by discrete quantities. This process is called *quantization*.

The function mapping each real number into the nearest integer, sometimes called "round", is given by the formula $x \mapsto \lfloor x + 1/2 \rfloor$ where $\lfloor x \rfloor$ is the integral part defined in (1.56). If instead of the integers $\mathbb{Z}$ we want the output to be Δ multiples of integers to get more (or less precision), we re-scale the argument and the value of this function to find the so called *uniform quantizer*

$$Q(x) = \Delta\lfloor \frac{x}{\Delta} + \frac{1}{2} \rfloor \qquad \text{with } \Delta > 0. \tag{2.13}$$

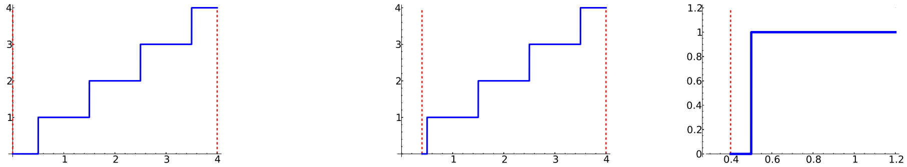
This is the simplest and more common quantization. The geometric idea is that $Q(f)$ is the approximation of f by a step function such that the spacing between steps is always a multiple of Δ. With Δ small the approximation will be good.

Do not forget that in our case we apply Q to a discrete set of values after sampling not to a function or signal. In computer science very often the application of the uniform quantizer is called *pulse code modulation* and abbreviated as PCM (sometimes preceded by the word “linear”). For instance the (uncompressed) `wav` audio file format uses PCM with a sampling frequency of $44100\,Hz$.



f(x) = sin(2πx) f(x) = Q(sin(2πx)) sampled & quantized

In practice, we often manage signals confined to an interval $[\alpha, \beta)$. If we use the uniform quantifier (2.13) as it is, the edges induce in general undesirable results. For instance, when applied with $\Delta = 1$ in the interval $[0, 4)$ and $[0.4, 4)$ we get



Interval [0, 4) Interval [0.4, 4) (detail)

In both cases we are using five levels and the results is not uniform. In the first example we use one level for $[0, 1/2)$ and another for $[1/2, 3/2)$ although the former interval has half length. The second example is even worse because we spend a whole level for the tiny interval $[0.4, 0.5)$.

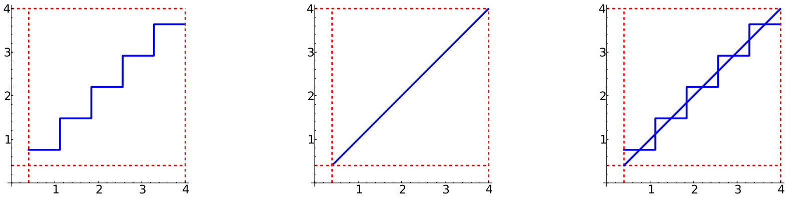
This problem does not appear if $\Delta^{-1}\alpha$ and $\Delta^{-1}\beta$ are half-integers (numbers of the form $k + 1/2$, $k \in \mathbb{Z}$) because in this situation Q jumps at α and β. We can force it with a convenient translation. Say that we want to employ $M \in \mathbb{Z}^+$ uniform levels and define for $I = [\alpha, \beta)$

$$\Delta = \frac{|I|}{M} \qquad \text{where} \quad |I| = \beta - \alpha. \tag{2.14}$$

The translation $T(x) = x - \alpha - \Delta/2$ moves $[\alpha, \beta)$ to $[\alpha', \beta')$ with $\Delta^{-1}\alpha'$ and $\Delta^{-1}\beta'$ half-integers, then the *uniform quantizer* for $[\alpha, \beta)$ is the function

$$Q_{[\alpha,\beta)}(x) = (T^{-1} \circ Q \circ T)(x) = \Delta\big\lfloor \Delta^{-1}(x - \alpha)\big\rfloor + \frac{\Delta}{2} + \alpha. \tag{2.15}$$

For instance, for $[0.4, 4)$ and $M = 5$ we have



uniform quantizer identity both

In this way the image of $Q_{[\alpha,\beta)}$ are the multiples of Δ displaced by $\Delta/2+\alpha$ because the multiples of Δ do not fit $[\alpha,\beta)$ in a uniform way. Note that $Q_{[\alpha,\beta)}$ fixes $\alpha+\delta/2$, $\beta-\Delta/2$ and in general $\alpha+\Delta(k+1/2)$ which are the centers of the intervals where $Q_{[\alpha,\beta)}$ is constant.

The *quantization error* for a certain x is $Q(x)-x$. If the sampled values are uniformly distributed in $[0,1]$ then the mean square error when we use the formula (2.15) is

$$\int_0^1 \big(Q_{[0,1)}(x)-x\big)^2\,dx=\frac{\Delta^2}{12}+O(\Delta^3). \tag{2.16}$$

In fact the term $O(\Delta^3)$ can be omitted, getting an exact formula, if $\Delta^{-1}\in\mathbb{Z}^+$ as required in (2.14). Hence typically the quantization error is like $\Delta/\sqrt{12}$.

Sometimes the processing of the digital data imposes some special ranges and scales and, in some sense, part of the quantizer (2.13) is applied in the A/D conversion and part in the the D/A conversion of (2.1). For example, imagine that we have a rectangular B/W image (this means shades of gray) that is sampled, let us say with a photometer, to get the gray tone of the pixel (i,j). This gives a collection of real numbers a_{ij}, a matrix, and we assume the normalization $a_{ij}\in[0,1)$ with $a_{ij}=0$ pure black and $a_{ij}=1$ pure white (as done in some applications). In the digital side we have a finite palette of c gray tones, commonly $c=256$ in our computer. Then $M=c$ in (2.14) gives $\Delta=c^{-1}$ and the effect of the uniform quantizer (2.15) is

$$\begin{array}{rcl} Q_{[0,1)}\colon [0,1) & \longrightarrow & \big\{\frac{1}{2c},\frac{3}{2c},\frac{5}{2c},\dots,\frac{2c-1}{2c}\big\} \\ a_{ij} & \longmapsto & c^{-1}\big(\lfloor ca_{ij}\rfloor+1/2\big) \end{array} \tag{2.17}$$

But our computer internally works with integers then our digital processor prefers to receive only this part of the quantization:

$$\begin{array}{rcl} Q^*\colon [0,1) & \longrightarrow & \{0,1,2,\dots,c-1\} \\ a_{ij} & \longmapsto & \lfloor ca_{ij}\rfloor \end{array} \tag{2.18}$$

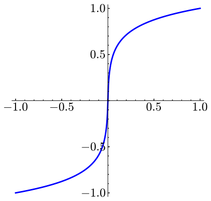
Once the information has been processed digitally, we sum $1/2$ and multiply by c^{-1} in the D/A converter of (2.1) and present it as a fake analog output or we smooth it to get a *bona fide* analog signal. There is a funny terminology around these simple concepts but probably you would find it verbosity. If you are eager to learn it, read [You14, Ch.2].

Uniform quantization is sometimes wasteful. If your signal is a sound wave corresponding to a conversation and you reserve an interval to cover a generous range of amplitudes (volume), typically the upper part of the range is seldom used because people is not shouting all the time (unfortunately there are counterexamples). Also a property of perception, sometimes called *Weber's law* enters in the game. It turns out that, especially beyond certain thresholds, one only feels proportional changes in a stimulus. It is like saying that our senses are logarithmic. If after quantization we get 2^{16} possible values (as in `wav` format) that we represent with 2 bytes but most of the time these numbers are small and when they are big we can allow large errors without noticing anything, it is clear that we are wasting bytes[2]. Solving this problem is closely related to coding, a topic with a pretty mathematical background that we shall treat later in these notes. Here we mention an idea at the analog level and an algorithm to devise optimal quantizers.

The idea is to apply a bijection f to the signal immediately after or before sampling in such a way that the sampled values become uniformly distributed or close to it. In this situation (2.13) is optimal. Correspondingly, the inverse function f^{-1} has to be applied in the last step of (2.1). In the previous example, if the upper part of the range is reached less frequently, then f must compress this part of the range.

A practical example is the *μ-law*, a standard in speech processing in telecommunications in U.S.A. and Japan. Once the signal is normalized to fit in $[-1, 1]$, it is applied

$$f(x) = \operatorname{sgn}(x)\frac{\log(1+\mu|x|)}{\log(1+\mu)} \tag{2.19}$$



with μ a certain fixed constant. Note that $f : [-1, 1] \longrightarrow [-1, 1]$ is a homeomorphism that can be inverted with a simple explicit formula. The upper and lower parts of the interval $[-1, 1]$ are compressed because the changes in these zones are less audible. On the other hand, for x around 0, the function f is well approximated by Cx with C a constant i.e., it is just a linear scaling.

The μ-law (2.19) produces a quasi-uniform distribution when applied to typical voice signals but in other applications similar functions can be expensive to get and invert. There is an algorithm than allows to get a similar effect acting on quantization knowing the density function g of the sampled real signal. To design a quantizer Q_g adapted to it, we consider that it is characterized by a partition of $\mathbb{R}$ into a finite number of intervals and a way to assigns to each interval a fixed element in it. In the jargon, the intervals are called *quantization regions* and the elements the *representation points*. Let us write

$$\mathbb{R} = \bigcup_{j=1}^{M} I_j, \quad I_1 = (-\infty, b_1), \quad I_M = [b_{M-1}, \infty), \quad I_j = [b_{j-1}, b_j) \text{ for } 1 < j < M, \tag{2.20}$$

with $a_j = Q_g(I_j) \in I_j$ the representation points. Clearly if g has a lot of mass around a point the a_j should cluster there. One could consider optimal a quantizer minimizing the mean square quantization error

$$\int_{-\infty}^{\infty} \big(Q_g(x) - x\big)^2 g(x)\, dx = \sum_{j=1}^{H} \int_{I_j} (a_j - x)^2 g(x)\, dx. \tag{2.21}$$

[2] I bet very rarely you store songs in your portable music player in `wav` format. Ripping CDs to another lighter formats (and even buying CDs) was a must not so long time ago.

This is a function F in $2M-1$ variables $b_1,\dots,b_{M-1},a_1,\dots,a_M$ and their extrema are reached at values with $\nabla F=\vec{0}$. By the fundamental theorem of calculus, we have

$$\frac{\partial F}{\partial b_j}=(a_j-b_j)^2g(b_j)-(a_{j+1}-b_j)^2g(b_j)\qquad\text{for } j=1,2,\dots,M-1,\tag{2.22}$$

and differentiating under the integral

$$\frac{\partial F}{\partial a_j}=2\int_{I_j}(a_j-x)g(x)\,dx\qquad\text{for } j=1,2,\dots,M.\tag{2.23}$$

If $g(b_j)\neq 0$, the vanishing of (2.22) is equivalent to $2b_j=a_j+a_{j+1}$. The *Lloyd-Max algorithm* [Llo82] [Max60] takes this information to solve $\nabla F=\vec{0}$ by successive approximations. Namely, the algorithm starts with any educated guess for the representation points and applies successively the following formulas (in the indicated order)

$$b_j=\frac{a_j+a_{j+1}}{2}\quad\text{for } j=1,2,\dots,M-1\quad\text{and}\quad a_j=\frac{\int_{I_j}xg(x)\,dx}{\int_{I_j}g(x)\,dx}\quad\text{for } j=1,2,\dots,M.\tag{2.24}$$

Note that the second set of equations solves (2.23) equated to 0. The algorithm ends when the values of the b_j's and a_j's stabilize with certain precision. A drawback of the algorithm is that it could converge to a critical point different from one in which the absolute minimum of F is attained. One can cook some counterexamples of this kind [Gal06] choosing a distribution function g with some valleys but anyway one trusts the educated initial guess should avoid this problem.

Suggested Readings. As you see, from the mathematical point of view, quantization is not a big deal. Probably you do not need extra bibliography. I only dare to repeat that in [You14] you can find the odd terminology used by engineers.

2.1.3 Data approximation

In practice, Theorem 2.1.1 allows to reconstruct a band limited function from their sampled values but as we mentioned in connection with Papoulis-Gerchberg algorithm and (2.8), its practical efficiency is objectionable in several situations. On the other hand, after quantization and digital processing very rarely it is judicious to assume that the resulting data are exact values of a band limited function.

Here we approach D/A conversion in a simple general mathematical form: We have discrete (digital) data and we want to approximate them with a somewhat smooth function. We restrict ourselves to 1D samples, as before, then have in mind sound signals rather than images.

You know how to do it, finite samples, simple functions like polynomials… connect the dots, literally. It is interpolation from basic courses of numerical analysis. You have some *interpolation nodes* $x_0<x_1<\dots<x_n$ and you want to find a polynomial P such that $P(x_j)=y_j$ for some given y_j. There is a nice theorem, with a short beautiful proof, providing the error with respect to the purported smooth function connecting the dots.

Theorem 2.1.3. *Given $x_0 < x_1 < \cdots < x_n$ and $f \in C^{n+1}([x_0, x_n])$ there exists a unique polynomial P of degree at most n such that $P(x_j) = f(x_j)$. For each $x \in [x_0, x_n]$ there exists $\xi \in (x_0, x_n)$ such that*

$$f(x) - P(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{j=0}^{n}(x - x_j). \tag{2.25}$$

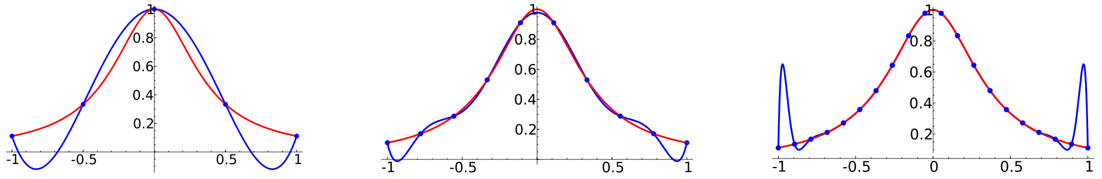
Proof. The existence of P follows taking $P = \sum f(x_j)L_k$ where L_k is the *Lagrange polynomial* $L_k(x) = \prod_{j\neq k}(x - x_j)/(x_k - x_j)$ that verifies $L_k(x_j) = 0$ for $j \neq k$ and $L_k(x_k) = 1$. The uniqueness follows because a polynomial of degree at most n with $n + 1$ zeros is identically zero.

For $x = x_j$, (2.25) is trivial. In the rest of the cases, for $x \in [x_0, x_n]$ fixed different from the interpolation points, define $g : [x_0, x_n] \longrightarrow \mathbb{R}$ by

$$g(t) = f(t) - P(t) - \big(f(x) - P(x)\big)\prod_{j=0}^{n}\frac{t - x_j}{x - x_j}. \tag{2.26}$$

It satisfies $g(x_0) = \cdots = g(x_n) = g(x) = 0$ and applying $n + 1$ times Rolle's theorem we conclude $g^{(n+1)}(\xi) = 0$ (isn't it beautiful?) and it gives the result because $P^{(n+1)} = 0$ and $(n + 1)!$ is the $(n + 1)$-derivative of $\prod(t - x_j)$. □

The $(n + 1)!$ grows to the speed of light (even faster because somebody said that the latter is constant). Then probably you expect that for "smooth" data the more (degree) the merrier. Let us check what happens with $f(x) = (1 + 8x^2)^{-1}$ in $[-1, 1]$ when we take N equally spaced nodes with $N = 5$, 10 and 15. Behold the results!



N = 5 N = 10 N = 20

For $N = 50$ the error has peaks of order 10^5. You are seeing the so-called *Runge's phenomenon* [KC96]. the rough idea under this weird situation is that for an analytic function with bounded convergence radius the Taylor coefficients behave like a power (by the root test) then you should not expect too much from the fraction in (2.25) and especially near the extremes the product involves a factorial in the numerator.

Definitively, it is not a good idea to use large degree polynomials to approximate functions[3] even if they are C^∞. What about using low degree polynomials?

A first approach is joining the dots as kids, using straight lines (the same method to define the length of rectifiable curve, if you prefer a more scientific precedent). Not bad but we get corners. It is up to us to decide what kind of regularity we want but

[3] There is still a chance if we move the nodes to the zeros of *Chebyshev polynomials* but we do not enter into it [Dav75, §4.1] [Atk89, §4.7].

some people would say that in graphical representations one can see at first glance first derivatives (growth, monotonicity) and second derivatives (curvature). Let us focus then on $C^2([x_0, x_n])$ interpolation. With something piecewise linear we get a continuous result, with quadratic we get C^1 and we need piecewise cubic to reach C^2. Given the interpolation nodes $x_0 < x_1 < \cdots < x_n$ and their images y_j, we define the space

$$C_n^2 = \{f \in C^2([x_0, x_n]) \;:\; f(x_j) = y_j,\ 0 \le j \le n \text{ and } \ f''(x_0) = f''(x_n) = 0\}. \tag{2.27}$$

The functions in this space that coincide with a polynomial of degree at least 3 in each interval $[x_j, x_{j+1}]$ are called *cubic splines*, the adjective *natural* is added to indicate the boundary condition $f''(x_0) = f''(x_n) = 0$. There are other possibilities not discussed here.

How do we know that we can find a (natural) cubic spline for any interpolation nodes and corresponding values? It is a theorem but a simple count suggests that it is true: Each polynomial of degree at most three has 4 coefficients and we have one of them for each interval $[x_j, x_{j+1}]$ then we have in total $4n$ unknowns. On the other hand, in the inner nodes $s \in C^2$ imposes $s(x_j^-) = s(x_j^+) = y_j$, $s'(x_j^-) = s'(x_j^+)$ and $s''(x_j^-) = s''(x_j^+)$ that make $4(n-1)$ linear equations. At the boundary nodes $s(x_0) = y_0$, $s(x_n) = y_n$, $s''(x_0) = s''(x_n) = 0$ give 4 more linear equations. A linear system with as many equations as unknowns is very likely to have a unique solution. This is the case for cubic splines and indeed the resulting linear system suits special known algorithms of numerical linear algebra which allow to deal with a large number of nodes. In the following proof we exemplify the situation in the case of equally spaced nodes. With little changes, it extends to the general case [SB02, §2.4.2].

Theorem 2.1.4. *Given $x_0 < x_1 < \cdots < x_n$ and $\{y_j\}_{j=0}^n$, there exists a unique cubic spline $s \in C_n^2$.*

Proof (for equally spaced nodes). After a linear change it suffices to consider $x_j = j$. Let us say that the restriction of s to the interval $[j, j+1]$ is $S_j(t+j)$ where $S_j : [0,1] \longrightarrow \mathbb{R}$ is the polynomial

$$S_j(t) = s_1^j t^3 + s_2^j t^2 + s_3^j t + s_4^j \qquad \text{with} \quad 0 \le j < n. \tag{2.28}$$

The upper indexes, of course, do not indicate powers here.

Let us write the conditions mentioned above. Firstly, s must join the interpolation points (x_j, y_j) then for $0 \le j < n$

$$s_4^j = y_j, \tag{2.29}$$

$$s_1^j + s_2^j + s_3^j + s_4^j = y_{j+1}. \tag{2.30}$$

On the other hand, $s \in C^2$ imposes $S_j'(0) = S_{j-1}'(1)$ and $S_j''(0) = S_{j-1}''(1)$. Equivalently, for $0 < j < n$

$$s_3^j = 3s_1^{j-1} + 2s_2^{j-1} + s_3^{j-1}, \tag{2.31}$$

$$2s_2^j = 6s_1^{j-1} + 2s_2^{j-1}. \tag{2.32}$$

Finally, the boundary values of the second derivative give

$$2s_2^0 = 0, \tag{2.33}$$

$$6s_1^{n-1} + 2s_2^{n-1} = 0. \tag{2.34}$$

In principle the linear system (2.29)–(2.34) seems messy but it becomes quite simple if we employ as unknowns the values $D_j = s'(j)$ that coincide with both sides of (2.31). Subtracting (2.30) and (2.29), it follows $s_1^j + s_2^j + s_3^j = y_{j+1} - y_j$ and on the other hand $D_{j+1} + D_j = (3s_1^j + 2s_2^j + s_3^j) + s_3^j$. Therefore $s_1^j = D_{j+1} + D_j - 2(y_{j+1} - y_j)$. In the same way we can get s_2^j. Summing up, if we find D_j we shall have proved that the linear system (2.29)–(2.34) has the solution

$$\begin{cases} s_1^j = D_{j+1} + D_j - 2(y_{j+1} - y_j), & s_3^j = D_j \\ s_2^j = -D_{j+1} - 2D_j + 3(y_{j+1} - y_j), & s_4^j = y_j. \end{cases} \tag{2.35}$$

To get an equation for D_j, we employ (2.32). Substituting (2.35) we obtain

$$D_{j-1} + 4D_j + D_{j+1} = 3(y_{j+1} - y_{j-1}), \tag{2.36}$$

for $0 < j < n$. For $j = 0$ and $j = n$ it has to be modified according (2.33) and (2.34) to

$$2D_0 + D_1 = 3(y_1 - y_0) \qquad \text{and} \qquad D_{n-1} + 2D_n = 3(y_n - y_{n-1}). \tag{2.37}$$

The linear system (2.36)–(2.37) has a *tridiagonal matrix*, one having nonzero elements only on the main diagonal and on the neighboring diagonals above and below. It is easy to see that it is nonsingular (for instance applying Gaussian elimination or computing the determinant with an inductive argument [Mui60]). There are very efficient methods to solve linear systems with this kind of matrices [SB02]. Once we know that D_j are uniquely determined, the same apply for the coefficients s_k^j thanks to (2.35). □

We now have a theorem that states that cubic splines exist and digging in the proof they are easy to compute numerically. Are they useful? Have we avoided with them the large wobbles of Runge's phenomenon? The answer is absolutely yes. It turns out that cubic splines are the "less curved" function connecting the interpolation points. In mathematical terms

Theorem 2.1.5. *Let s be the unique spline in C_n^2. Then for any other function $f \in C_n^2$ we have $\|s''\|_2 \le \|f''\|_2$ with equality only if $s = f$.*

Proof. We have

$$\|f''\|_2^2 = \|f'' - s''\|_2^2 + \|s''\|_2^2 + 2\int_{x_0}^{x_n} s''(f'' - s''). \tag{2.38}$$

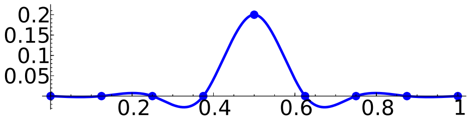
If the integral is zero, we get the first part of the result. Let us check this point. On each interval $[x_j, x_{j+1}]$ the spline s is a polynomial s_j, integrating by parts on each of these intervals

$$\int_{x_0}^{x_n} s''(f'' - s'') = \sum_{j=0}^{n-1} \left(f'(x_{j+1})s''(x_{j+1}) - f'(x_j)s''(x_j)\right) - \sum_{j=0}^{n-1} \int_{x_j}^{x_{j+1}} s_j'''(f' - s'). \tag{2.39}$$

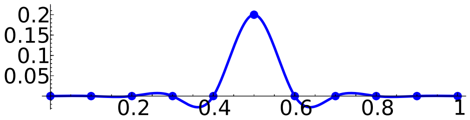
The first sum telescopes to 0. The last integral is also zero because s_j''' is constant and $f-s$ vanishes at x_j and x_{j+1}.

If $\|s''\|_2 = \|f''\|_2$ then (2.38) gives $\|f''-s''\|_2 = 0$ and f and s may differ in a quadratic function but $f''(x_0) = s''(x_0) = 0$, $f(x_0) = s(x_0)$, $f(x_1) = s(x_1)$ imply that they actually coincide. □

Let us try a couple of numerical examples. Consider $2N+1$ nodes in $[0,1]$ and impose $s(x_j) = 0$ except for the central point $s(x_N) = 0.2$. For $N = 4$ and $N = 5$ (9 and 11 nodes), we obtain



Nine nodes



Eleven nodes

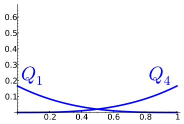
In both cases we see that the first and the last pieces are almost flat. For $N = 4$ we have $s'(0) \approx -6 \cdot 10^{-3}$ and for $N = 5$ we have $s'(0) \approx 1.6 \cdot 10^{-3}$.

After this little long distance influence, one may wonder whether there exists a nontrivial compactly supported cubic spline b, meaning that it is identically zero outside an interval determined by two nodes $x_{j_1} < x_{j_2}$, hence it must hold $b''(x_{j_k}) = b'(x_{j_k}) = b(x_{j_k}) = 0$. As before, we restrict ourselves to the case of equally spaced nodes. Some calculations show that there is no solution for $j_2 - j_1 < 4$. On the other hand, for $x_j = j-2$, $0 \le j \le n = 4$ we have

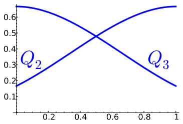
(2.40)

$$b(t) = Q_j(t-x_j) \quad \text{if } x_j \le t \le x_{j+1} \text{ with } \begin{cases} Q_1(t) = \frac{1}{6}t^3, & Q_3(t) = Q_2(1-t), \\ Q_2(t) = \frac{1}{6}(t+1)^3 - \frac{2}{3}t^3, & Q_4(t) = Q_1(1-t), \end{cases}$$

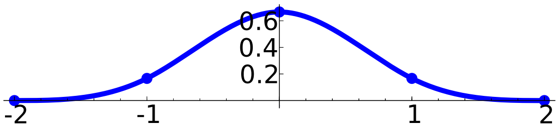
that vanishes to order 3 at the end nodes and can be safely extended as zero outside its support. The graphics of the pieces Q_j composing this spline and of the spline itself are



Q$_1$ and Q$_4$



Q$_2$ and Q$_3$



The spline b

This cubic spline is, except for scaling, the only one supported in three equally spaced consecutive nodes. The splines like this, having minimal support for a given regularity are called *B-splines*. What is the big deal about them? It is related to applications. Let us say that we have to interpolate f at a huge number of consecutive integer nodes $x_0 < x_1 < \cdots < x_n$. It requires solving the linear system indicated in the proof of Theorem 2.1.4. Moving a little a value y_j, as we have seen, has little influence except for few nearby nodes, then one may consider a cheap alternative that does not require any numerical linear algebra

$$s_f(x) = \sum_j f(x_j)b(t-x_j) \tag{2.41}$$

where we use integer translated copies of b as a basis (the "B" in "B-splines" stands for "basis"). As b only overlaps with four copies, we have the formula

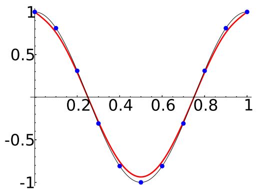
$$(2.42)\qquad s_f(x) = \sum_{k=1}^{4} Q_k(t-x_j)f(x_{j+3-k}) \qquad \text{for} \quad x_j \le t \le x_{j+1} \quad \text{with} \quad 0<j<n-1.$$

We have to take a decision about what happens on the boundary $j=0$ and $j=n-1$. A natural one is to keep (2.42) in these cases introducing two artificial nodes x_{-1} and x_{n+1} at which we assume f takes the linear extrapolated values i.e., $f(x_{-1}) = 2f(x_0) - f(x_1)$ and $f(x_{n+1}) = 2f(x_n) - f(x_{n-1})$.

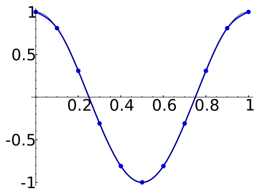
Note that each evaluation requires only 4 multiplications by the pieces of the B-spline, instead of using $n+1$ coefficients coming from a previously solved linear system. The drawback is that (2.42) does not perform actual interpolation. At the nodes we have

$$(2.43)\qquad s_f(x_j) = \sum_{k=1}^{4} Q_k(0)f(x_{j+3-k}) = \frac{f(x_{j-1}) + 4f(x_j) + f(x_{j+1})}{6}.$$

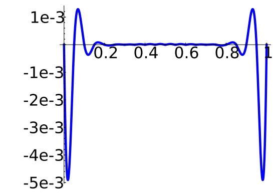
What is the point of having a hyper-speed non-interpolating formula like (2.42)? A possible answer is that after quantization we have already introduced errors everywhere and imposing error free interpolation of error affected data is in some occasions to use a sledgehammer to crack a nut. Moreover for smooth data, B-splines give a good approximation to spline exact approximation. For instance, if we interpolate $f(x) = \cos(2\pi x)$ in $[0,1]$ with $n=10$ the maximal error is like $2\cdot 10^{-2}$ and with B-splines $6\cdot 10^{-2}$. Increasing n to 20 the figures are $5\cdot 10^{-3}$ and $1.6\cdot 10^{-2}$ and for $n=40$, 10^{-3} and $4\cdot 10^{-3}$. It is fair to mention that the error in the cubic spline interpolation accumulates near the endpoints and in the inner point the approximation is by far better. This is because $f''(0), f''(1) \neq 0$ while natural splines satisfy $s''(0) = s''(1) = 0$. Anyway, the error using B-splines is small taking into account the simplicity of the method.



n = 10, B-splines



n = 10, interpolation



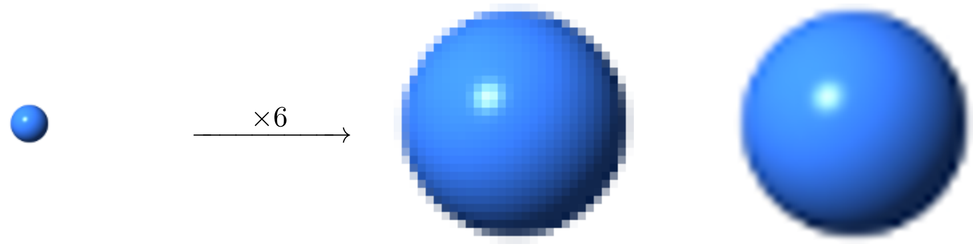
n = 20, interpolation error

To emphasize that this is practical, let us consider an example related to GIMP (the free, cross-platform and open-source competitor of Adobe Photoshop®). If you try to scale an image with it, under the name "Quality" there is a little menu to choose None, Linear, Cubic and Sinc. By default it is selected Cubic that, according to the documentation, "produces the best results". Why on earth are there several possibilities for this simple operation? In principle to enlarge a photo making the sides six times longer is just passing the color of the point (x,y) to the point $(6x,6y)$. Yes, it is but $f(\vec{x}) = 6\vec{x}$ is a bijection when applied on $\mathbb{R}^2$ and not when applied on $\mathbb{Z}^2$ and the pixels are labeled by integers. The result would be an image plenty of holes. OK, let us take the units six times bigger, in this way we apply the pixel (m,n) into the square of pixels $(6m+k, 6n+l)$ with $k,l \in \{0,1,\dots,5\}$. This corresponds to None in GIMP. It seems the only natural solution but the results are in general visually poor because we see the enlarged edges of the squares of the pixels. A better solution is to assign the color to the pixels $(6m, 6n)$, use them as nodes and interpolate the rest of the pixels. Exact cubic interpolation would be expensive (an image 640×400 requires 256000 nodes) while B-spline approximation is affordable. How can we manage B-splines in this $2D$ setting? If $f = f(x,y)$ is the function giving the color, (2.42) generalizes promptly to approximate it by

(2.44)

$$\sum_{k=1}^{4}\sum_{l=1}^{4} Q_k(x-x_j)Q_l(y-y_m)f(x_{j+3-k}, y_{m+3-l}) \qquad \text{for} \quad x_j \le x \le x_{j+1}, \quad y_m \le y \le y_{m+1}.$$

This is the Cubic method. As you suspect, Linear means piecewise linear approximation. Finally Sinc is an interpolation method related to the sinc function, a kind of smoothing of (2.8). Here you can see an example of enlarging an image without and with B-splines. Which one do you prefer?



Original 30 × 30 None 180 × 180 Cubic 180 × 180

For the avid reader, a last brief comment about non-interpolating cubic curves. Probably you have heard the name *Bézier curves*. They are based in the following fact: Given $P_0, P_1, P_2, P_3 \in \mathbb{R}^2$, the curve $\sigma : [0,1] \longrightarrow \mathbb{R}$

$$\sigma(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3. \tag{2.45}$$

joins P_0 and P_1 and pass somewhat close to P_1 and P_2. The segments P_0P_1 and P_2P_3 are tangent to the curve. The Bézier curves are composed by curves like this. Many interactive applications use them because it is quite intuitive to use P_1 and P_2 as *control points* that when moved change locally the aspect of the curve.

Suggested Readings. Lagrange interpolation, splines and B-splines are discussed in the second chapter of [SB02]. Look up this book for a mathematically spotless treatment of several numerical analysis methods. There is a lot of information about Bézier curves and their relatives on the internet and on texts oriented to the applications. I have found very clear the exposition in [Bus03].

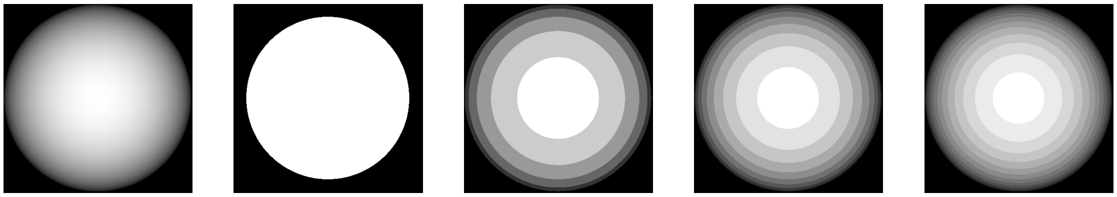
2.1.4 Dithering

If I say that *dithering* consists of improving A/D conversion adding noise to the analog signal you will think I am crazy. We are not talking about me but truly this seems idiotic until one sees an example with images or sound. The possibilities to include sounds here are not very informative (they reduce to boo, miaow, ha, ...) then we focus only on the case of images.

Consider the semi-spherical shaped function

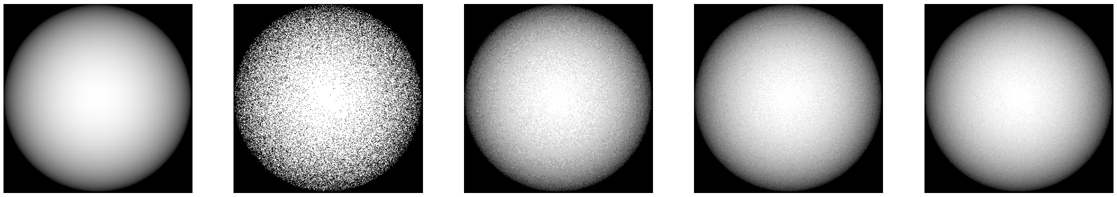
$$f(x,y) = \sqrt{\max(1-x^2-y^2,0)} \qquad \text{for} \quad x,y \in [-1,1]. \tag{2.46}$$

Let A be the map performing the spatial quantization with a grid 400×400. In this way A is a 400×400 matrix storing orderly the values of f. Let us assume that the value of $f(0,0)$ is clamped[4] to $0.999\dots$ and hence the elements of A are real numbers $a_{ij} \in [0,1)$. If we have a large palette of gray tones (theoretically an infinite one), we can represent faithfully the values in $[0,1)$ and plot A as a spherical gradient. In some situations we have a reduced palette, for instance in newspapers or GIF images. Let us say that we have to our disposal c gray tones labeled as $0,\ 1,\ 2,\dots,\ c-1$ varying from black to white. Quantizing under this restriction, our digital image assigns the color $\lfloor ca_{ij} \rfloor$ to the pixel (i,j). This is the natural option and it seems the only sound option. On the other hand, the following examples show that the result is utterly disappointing if c is small.


original c = 2 c = 6 c = 10 c = 14

When $c = 2$, as the graph of f is very steep near the unit circle, the values with $f > 1/2$ determine a circle close to it, so more or less we have a white circle where we had a gradient. For greater, but still small, values of c, we see clearly rings corresponding to the quantized levels that do not recall the original image.

Now it goes the crazy idea. Let ξ_{ij} be independent random variables under a uniform distribution in $[0,1)$. The noisy discrete signal (matrix) $\widetilde{A} = A + (c-1)^{-1}\Xi$ with $\Xi = (\xi_{ij})$ has elements $\widetilde{a}_{ij} \in \big[0, c/(c-1)\big)$. Now, the digital image with color $\lfloor (c-1)\widetilde{a}_{ij} \rfloor$ at pixel (i,j) looks, no doubt, more satisfactory

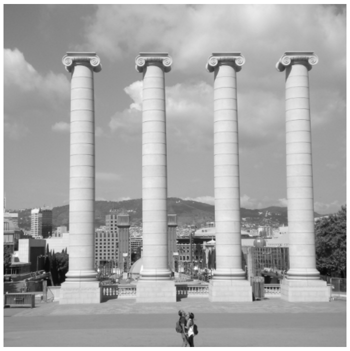

original c = 2 c = 6 c = 10 c = 14

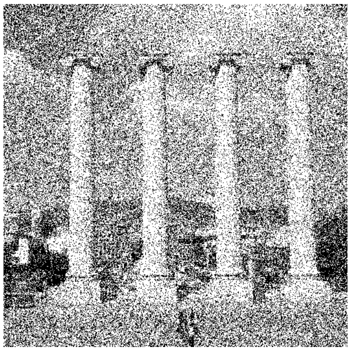
[4] The term *clamping* or *clipping* is used in signal processing to mean that the signal f is constrained to an interval $[a,b]$ changing f by $\max(\min(f,b),a)$ usually to avoid overflow errors.

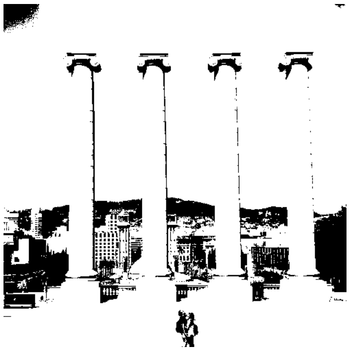
The result for $c = 14$ is not bad and for $c = 2$, although it is clearly defective, gives a better idea about the gradient than a perfect white circle.

Usual photographs are plenty of gradients but often what calls our attention are the sharp edges, then the brute force dithering we have just introduced, called *random dithering*, could be less effective than in the previous example.

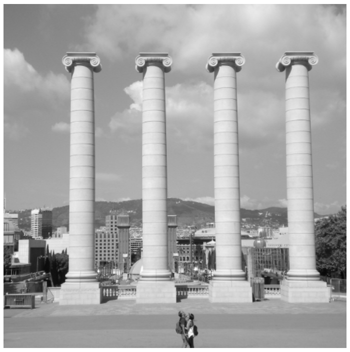
In these images (of original size 500×500) for $c = 2$ without dithering we lose completely the clouds and the sky, nevertheless the result may be more pleasant than the noisy image with dithering.

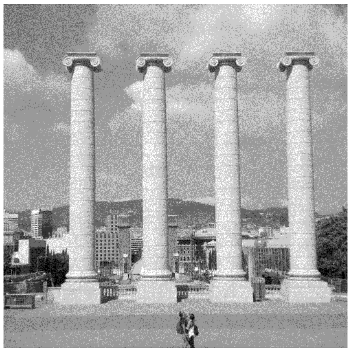

original

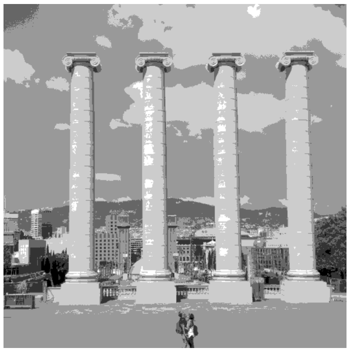

c = 2
dithering


c = 2
no dithering

For $c = 6$, without dithering the sky is artificial and with artifacts and the columns show an unrealistic aspect but somebody with a not so twisted taste could consider it preferable to the version with dithering that is more informative with respect to the details but contains visible grains of noise.


original


c = 6
dithering


c = 6
no dithering

There are several methods to avoid the noisy aspect caused by random dithering. Essentially all of them substitute the randomness by something deterministic. We discuss here two of these methods. Both are quite popular, being a noticeable asset the ease to program the corresponding algorithms even in a low-level environment.

Firstly we present the most common form of *ordered dithering*. It uses a real matrix M_k of size $2^k \times 2^k$, called *Bayer matrix*, defined according to the recurrence

$$M_k = \begin{pmatrix} M_{k-1} & M_{k-1} + 2 \cdot 2^{-2k} \\ M_{k-1} + 3 \cdot 2^{-2k} & M_{k-1} + 1 \cdot 2^{-2k} \end{pmatrix} \qquad \text{with} \quad M_0 = (0). \tag{2.47}$$

Here, adding a number to a matrix means adding it to every entry. The entries of M_k are a rearrangement of $\{j2^{-2k}\}_{j=0}^{2^{2k}-1}$ that, following the literature, maximizes the average distance between consecutive entries where the last row/column is consecutive to the first one (I have not been able to find a proof of this in the bibliography). The first two Bayer matrices are

$$M_1 = \frac{1}{4}\begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix} \qquad \text{and} \qquad M_2 = \frac{1}{16}\begin{pmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{pmatrix}. \tag{2.48}$$

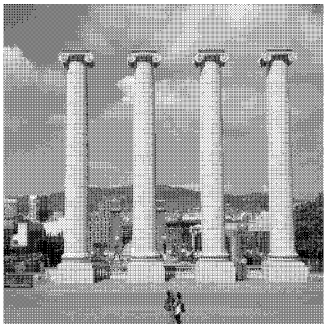
In principle a greater matrix is better but on the other hand, it has to be small in comparison to the size of the image. Once we have selected the matrix, we proceed as before but with the noise

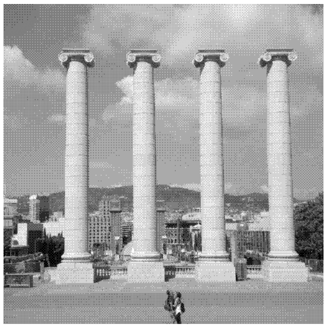
$$\xi_{ij} = M_{i'j'} \qquad \text{with} \qquad i' \equiv i \pmod{2^k}, \quad j' \equiv j \pmod{2^k} \tag{2.49}$$

where $M_k = (m_{ij})$. In other words, the noise matrix is obtained now tiling the image size with M_k.

Let us see the results for 4×4 matrices ($k = 2$):

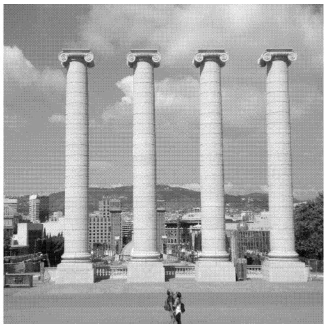
original

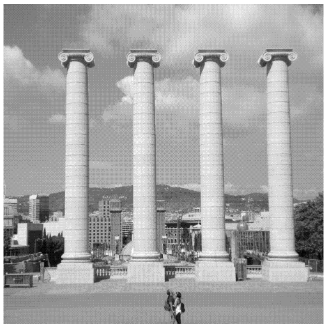

c = 2


c = 4

and for higher values of c

original

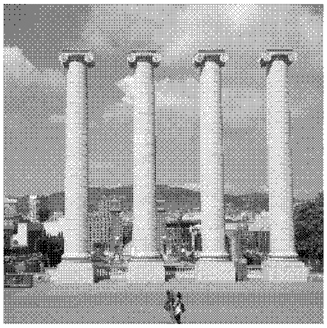

c = 6


c = 8

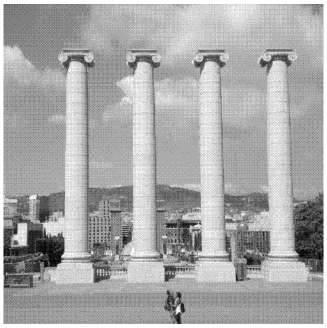
The result is clearly much better than the one obtained with random dithering. With 16×16 matrices ($k = 4$, often employed in practice) the improvements are not very noticeable.

For instance, for the smaller values of c we get:

original



c = 2



c = 4

Although the results are rather good, cross-hatch pattern artifacts appear typically with this method. The result looks very often like fabric and, obviously, images with big variations between adjacent pixels can give weird results. One alternative is to combine both techniques adding a small random noise to reduce the visual impact of the pattern due to M_k. Another possibility is to use tiles with more involved forms instead of square matrices. We do not explore these possibilities but a completely different idea.

There is a collection of techniques generically known as *error diffusion dithering* that try to compensate the quantization error in each pixel distributing it among the values of neighboring pixels. By far the most common version is the *Floyd-Steinberg dithering* [FS76]. In it, the pixels are scanned from left to right and from top to bottom and in each pixel the quantization error is distributed according to the proportion indicated in the following scheme

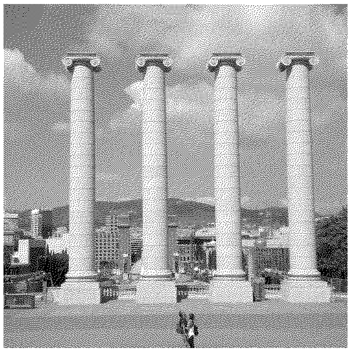
	■	7/16
3/16	5/16	1/16

The pixel marked in black is the one that has been quantized and the rest of the pixels in the scheme receive a part of the quantization error, of course before suffering themselves quantization. In formulas, the pass from a matrix A as before to its quantized version B with elements (colors) in $\{0, 1, 2, \dots, c-1\}$ is

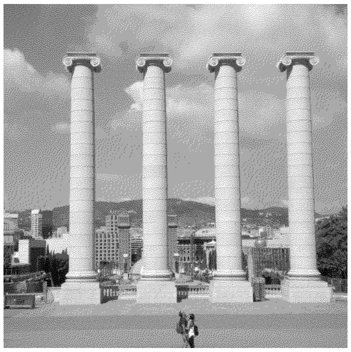
$$\begin{cases} b_{ij} = \lfloor ca_{ij} \rfloor, & e_{ij} = b_{ij} - ca_{ij} + \dfrac{1}{2}, \\ \widetilde{a}_{i+1\,j} = a_{i+1\,j} + \dfrac{7}{16} e_{ij}, & \widetilde{a}_{i+1\,j+1} = a_{i+1\,j+1} + \dfrac{1}{16} e_{ij}, \\ \widetilde{a}_{i-1\,j} = a_{i-1\,j} + \dfrac{3}{16} e_{ij}, & \widetilde{a}_{i\,j+1} = a_{i\,j+1} + \dfrac{4}{16} e_{ij} \end{cases} \tag{2.50}$$

where $\widetilde{a}_{k\,l}$ are the updated values of A. For pixels on the boundary one has to define in some way a_{ij} with $i, j \in \{0, N\}$. This is less important and simply skipping these values does not cause a problem.

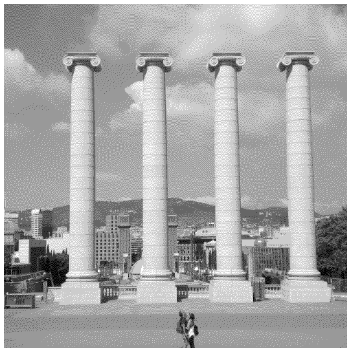
The results are quite impressive given the simplicity of the method:



c = 2

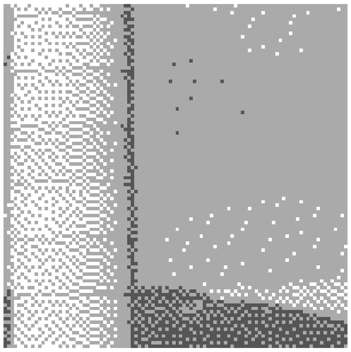


c = 4

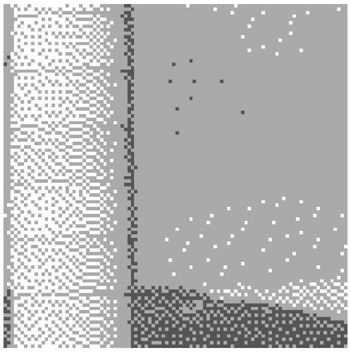


c = 6

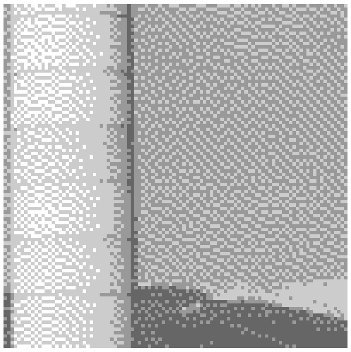
To better appreciate the quality of the result, here it is a zoom of the central 100×100 square of the images:



c = 2



c = 4



c = 6

Probably you are wondering what is interesting in this choice of the coefficients in the scheme (other diffusion methods employ different neighboring pixels and coefficients). It turns out that for $c = 2$, when having a middle gray $a_{ij} = 0.5$, the result is a checkerboard pattern. For a greater number of colors, the middle tones are represented in this optimal way replacing in the checkerboard black and white by the upper and lower adjacent colors.

For people that like whining, Floyd-Steinberg dithering has a serious drawback, the diffusion of the quantization error tends to form worm-like lines in the dithered image. An idea to minimize this effect is to raster the pixels in a funny (fractal) way instead following horizontal lines, another idea is to explore other diffusion schemes.

With a little of experimentation, surely a lot of tricks will come to your mind. You can find hundreds of ideas in the literature (even books like [Uli87]). Unfortunately some of them, although simple, are under patent laws that also apply if you rediscover the idea on your own.

Suggested Readings. There is a lot of information about dithering in the internet. One of the most informative sites that I have found is `http://caca.zoy.org/study/`. Do not get wrong with the software under Spanish lavatory names hosted in the main page, it is done by serious programmers.

2.2 Discrete Fourier analysis

2.2.1 Some discrete transforms

We had already seen in (1.32) the discrete analogue of the Fourier series and integrals. In fact it motivated them. We also saw it later in the more general setting of finite abelian groups (1.99) where (1.32) corresponds to $G = \mathbb{Z}_N$, the cyclic group of N elements. We recover here the definition with some minor notational changes to match part of the literature.

Given a vector $\vec{x} = (x_0, x_1, \dots, x_{N-1}) \in \mathbb{C}^N$, we define its *discrete Fourier transform*, abbreviated DFT, as the vector $(\widehat{x}_0, \widehat{x}_1, \dots, \widehat{x}_{N-1})$ where $\widehat{x}_n$ are the "Fourier coefficients"

$$\widehat{x}_n = \sum_{m=0}^{N-1} x_m e(-nm/N). \tag{2.51}$$

Given these coefficients, one can recover the original vector through the *Fourier inversion formula*

$$x_n = \frac{1}{N}\sum_{m=0}^{N-1} \widehat{x}_m e(nm/N). \tag{2.52}$$

The relation with (1.99) is clear: The vector $\vec{x}$ corresponds to the function $f : \mathbb{Z}_N \longrightarrow \mathbb{C}$ with $f(\overline{n}) = x_n$ for $0 \le n < N$.

An elegant way of introducing the orthogonality relations (1.33) is defining

$$U = (u_{kl})_{k,l=1}^N \in \mathrm{U}(N) \qquad \text{with} \quad u_{kl} = \frac{1}{\sqrt{N}} e\Big(-\frac{(k-1)(l-1)}{N}\Big). \tag{2.53}$$

As usual, $\mathrm{U}(N)$ means the $N \times N$ *unitary matrices*, verifying $A^\dagger A = \mathrm{Id}$ where $A^\dagger$ is the conjugate transpose. With this notation, (2.51) and (2.52) are respectively

$$\vec{f} = \sqrt{N} U \vec{x} \qquad \text{and} \qquad \vec{x} = \frac{1}{\sqrt{N}} U^\dagger \vec{f} \tag{2.54}$$

with $\vec{f} = (\widehat{x}_0, \widehat{x}_1, \dots, \widehat{x}_{N-1})$. In this way the inversion formula is a direct consequence of $U \in \mathrm{U}(N)$. The unitary matrices $\mathrm{U}(N)$ preserve the standard scalar product, indeed this property is the reason to introduce them. One deduces at once the *Parseval identities*

$$\sum_{n=0}^{N-1} |x_n|^2 = \frac{1}{N}\sum_{n=0}^{N-1} |\widehat{x}_n|^2 \qquad \text{and} \qquad \sum_{n=0}^{N-1} \overline{x}_n y_n = \frac{1}{N}\sum_{n=0}^{N-1} \overline{\widehat{x}}_n \widehat{y}_n. \tag{2.55}$$

In the context of locally compact abelian groups there exists a concept of convolution that in our simple setting is the obvious discretization of (1.76). Given $\vec{x}, \vec{y} \in \mathbb{C}^N$, its *convolution* $\vec{x} * \vec{y}$ is

$$\vec{z} = \vec{x} * \vec{y} = (z_0, z_1, \dots, z_{N-1}) \qquad \text{with} \quad z_n = \sum_{\substack{k=0 \\ l \equiv n-k}}^{N-1} \sum_{\substack{l=0 \\ (\mathrm{mod}\ N)}}^{N-1} x_k y_l. \tag{2.56}$$

If one defines $y_{l+mN} = y_l$ for $m \in \mathbb{Z}$, the convolution can be defined with the lighter formula $z_n = \sum_{k=0}^{N-1} x_k y_{n-k}$. The analogue of (1.77) is

$$\widehat{z}_n = \widehat{x}_n \widehat{y}_n. \tag{2.57}$$

The underlying idea in the previous definition is to mimic the formulas in the theory of Fourier series substituting functions by vectors formed by sampled values. As we have seen, for regular 1-periodic functions few terms of the Fourier series give a good approximation. We would like to capture a similar concept in the sampled values. A qualitative way of understanding the situation is through the following combinatorial result that is close to the first formula in (1.69)

Lemma 2.2.1. *For $n \neq 0$ we have*

$$\widehat{x}_n = x_{N-1} - x_0 + \sum_{k=0}^{N-2} \frac{x_{k+1} - x_k}{1 - e(-n/N)} e(-n(k+1)/N). \tag{2.58}$$

Proof (sketch). Use the elementary identity

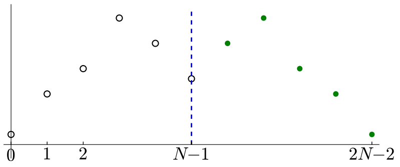
$$\sum_{k=0}^{N-1} x_k y_k = \sum_{k=0}^{N-2} (x_k - x_{k+1}) \sum_{l=0}^{k} y_l + x_{N-1} \sum_{l=0}^{N-1} y_l \tag{2.59}$$

with $y_k = e(-kn/N)$. □

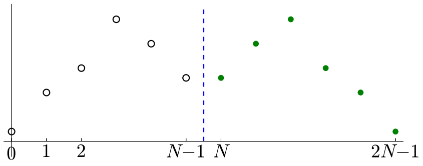
If $\vec{x}$ corresponds to sampled values of a smooth varying periodic function, then $x_{k+1} - x_k$ and $x_{N-1} - x_0$ are small. If n is not close to 0 or N the denominator $1 - e(-n/N)$ is far apart from zero and the corresponding Fourier coefficients $\widehat{x}_n$ are less important to recover the signal with the inversion formula.

In many applications the condition $x_0 \approx x_{N-1}$ resembling periodicity is not assured out of the box and Lemma 2.2.1 suggests that under $x_k \approx x_{k+1}$ the middle Fourier coefficients are equally important for the reconstruction: We cannot compress the signal forgetting some of them. To avoid this problem, it is introduced the *discrete cosine transform*, abbreviated DCT, which has the extra technical advantage of using only real values when the input signal is real. The idea is very simple: If we symmetrize the values of a smooth discrete signal, we force the first and the last value to be close. In the continuous setting it would correspond to the even periodic extension.

There are two typical situations called DCT-I and DCT-II which correspond to perform the symmetry around $N-1$ or around $N-1/2$. The second one is more natural and pleasant from the numerical point of view because it doubles the number of points. The difference is illustrated with the following pictures in which the original signal is marked with hollow points.



DCT-I



DCT-II

Hereafter we only focus on DCT-II and we call it simply DCT as usual.

The duplication of the signal sampled points and the symmetry through $N-1/2$ imply that in the analysis with the DFT we get terms $e\big(n(m+1/2)/2N\big)+e\big(-n(m+1/2)/2N\big)$ that give rise to cosines (see the proof below). The inversion formula loses a part of its symmetry but, as pointed before, we avoid problems with the lack of periodicity and the calculations are with real numbers for real signals. The concrete definition of the DCT and its inversion is contained in the following result

Proposition 2.2.2. *Given* $\vec{x}=(x_n)_{n=0}^{N-1}\in\mathbb{C}^N$ *we define its* discrete cosine transform DCT *as*

$$\widehat{x}^c_n=\sum_{m=0}^{N-1}x_m\cos\big(\frac{\pi n}{N}(m+\frac{1}{2})\big). \tag{2.60}$$

Then we have the Fourier inversion formula

$$x_m=\frac{\widehat{x}^c_0}{N}+\frac{2}{N}\sum_{n=1}^{N-1}\widehat{x}^c_n\cos\big(\frac{\pi n}{N}(m+\frac{1}{2})\big). \tag{2.61}$$

As suggested before, a way of proving this result is to apply the inversion formula (2.52) for the DFT to the symmetrized signal. For illustration we follow this approach although it is not the simplest. An ultra-quick proof motivated by the discretization of certain ODE is included in [Str99]. In this interesting paper it is claimed that the DCT was not discovered until 1974 [ANR74].

Proof. Let y_n the symmetrized signal. For convenience we consider the indexes of y modulo $2N$. In this way, $y_n=x_n$ if $0\le n<N$ and $y_n=x_{-1-n}$ for $-N\le n<0$. The definition (2.51) applied to y reads

$$\widehat{y}_n=\sum_{m=0}^{N-1}x_m e\Big(-\frac{nm}{2N}\Big)+\sum_{m=-N}^{-1}x_{-1-m}e\Big(-\frac{nm}{2N}\Big)\qquad\text{for}\quad |n|\le N. \tag{2.62}$$

Changing m into $-m-1$ in the last sum, we have

$$\widehat{y}_n=\sum_{m=0}^{N-1}x_m\bigg(e\Big(-\frac{nm}{2N}\Big)+e\Big(\frac{nm}{2N}\Big)e\Big(\frac{n}{2N}\Big)\bigg). \tag{2.63}$$

The big parenthesis is $2e(n/4N)\cos\big(\pi n(m+1/2)/N\big)$ that vanishes for $n=N$. Then

$$\widehat{y}_n=2e\Big(\frac{n}{4N}\Big)\widehat{x}^c_{|n|}\quad\text{for}\quad |n|<N\qquad\text{and}\qquad \widehat{y}_N=0. \tag{2.64}$$

The inversion formula (2.52) gives for $0\le n<N$

$$2Nx_n=2Ny_n=\sum_{m=-N+1}^{N-1}2e\Big(\frac{m}{4N}\Big)\widehat{x}^c_{|m|}e\Big(\frac{mn}{2N}\Big) \tag{2.65}$$

and grouping the terms m and $-m$ we have the formula in the statement. □

Let us finish introducing two forms of another transform related to the discretization of classic Fourier analysis.

If theoretically we have to our disposal all the sampled values in past and future of a signal $(x_n)_{n=-\infty}^{\infty}$, we may think that

$$\sum_{n=-\infty}^{\infty} x_n e(-n\xi) \tag{2.66}$$

might be an approximation to its Fourier transform. This is called the *discrete time Fourier transform* because it only involves sampled values of the signal at discrete times. The series (2.66), when converges, defines a 1-periodic function then it does not approximate the Fourier transform which must decay by Riemann-Lebesgue lemma. Indeed, if $x_n = f(n)$ with a rapidly decaying f, by Poisson summation formula (2.11)

$$\sum_{n=-\infty}^{\infty} \widehat{f}(\xi + n) = \sum_{n=-\infty}^{\infty} x_n e(-n\xi). \tag{2.67}$$

The *Z-transform* of $(x_n)_{n=-\infty}^{\infty}$ is (2.66) after the complex change of variables $z = e(\xi)$,

$$X(z) = \sum_{n=-\infty}^{\infty} x_n z^{-n}. \tag{2.68}$$

Of course this is just a formal series if we cannot assure the convergence. This apparently unmotivated transform plays an important role for engineers because it fits nicely the theory of filter design. From the mathematical point of view it is a generating function that is used to solve linear difference equations. The most fundamental property of the Z-transform is that

$$z_n = \sum_{l=-\infty}^{\infty} x_l y_{n-l} \qquad \text{implies} \qquad Z(z) = X(z)Y(z) \tag{2.69}$$

where X, Y and Z are, respectively, the Z-transforms of the sequences x_n, y_n and z_n.

Suggested Readings. Any book on digital signals enters in the definition of several discrete transforms and in the basics of discrete Fourier analysis but if you are a theoretician and you want to read a masterpiece from the point of view of the quality of the exposition, your book is [Ter99].

2.2.2 An example: JPEG

Do the math, each pixel of a standard color image requires 3 bytes, then a 3648×2736 photo (this is the size provided by my low quality old digital camera) requires something like 30 megas. Why in your computer or in your cell phone do they need much less space? The answer is that they are compressed, but not as something like you get when you apply `gzip` to a text file containing your assignment or your essay for a course that (fortunately) can be recovered without changing a comma. The kind of compression commonly applied to photos is lossy compression, you lose some information in an irreversible way. The camera and other devices know what information to lose keeping the visual aspect thanks to an algorithm heavily based on discrete Fourier analysis.

To say the whole truth, professional cameras allow to use the RAW image format that, as the name indicates, contains the signal almost as it comes out from the sensor. For mortals and even for arguably the most of the professional applications, the king format is JPEG also known as JPG. The name, different from the official original one, is actually metonymical because these letters stands for *Joint Photographic Experts Group*, the team that introduced the specifications of the format.

The treatment of color does not add too much to the essence of the method and involves some extra technicalities, then we worry firstly about B/W images (meaning colored with shades of gray). Our camera or scanner have already performed sampling and quantization and the image (signal) in a digital device is an integer matrix $C = (c_{jk})_{j=1,\,k=1}^{H\quad W}$ where $c_{jk} \in \{0, 1, 2, \dots, 255\}$ specifies the gray tone of the pixel (j, k), being 0 black and 255 white. Here H is the height and W is the width, in pixels.

A first step is the subdivision of the image surrogate C into blocks of size 8×8. In some sense JPEG process individually these tiny blocks taking each few millimeters on your screen but it is important for the final compression that many of these blocks give alike outputs after signal processing. A problem appears at the edges if $8 \nmid H$ or $8 \nmid L$. In this case one invents artificially new pixels at the border to force H and L to be multiples of 8. We disregard this point here.

Each block can be considered as a function

$$F : \mathbb{Z}_8 \times \mathbb{Z}_8 \longrightarrow \{0, 1, 2, \dots, 255\} \subset \mathbb{R} \tag{2.70}$$

where the arguments of F indicate the position in the block matrix B i.e., $F(\overline{j}, \overline{k}) = b_{jk}$ for $j, k \in \{0, 1, \dots, 7\}$ where here and hereafter in this subsection we consider matrix indexes starting from 0.

We can write the inversion formula of Proposition 2.2.2 as

$$x_m = \frac{1}{N} \sum_{n=0}^{N-1} \alpha_n \widehat{x}_n^c \varphi_n(m) \quad \text{with} \quad \varphi_n(m) = \cos\big(\frac{\pi n}{N}(m + \frac{1}{2})\big) \quad \text{and} \quad \alpha_n = \begin{cases} 2 & \text{si } n \ne 0, \\ 1 & \text{si } n = 0. \end{cases} \tag{2.71}$$

If we apply it with $N = 8$ to $F(\cdot, l)$ and to $F(k, \cdot)$ i.e., taking discrete cosine transforms on each variable, we have the discrete Fourier expansion

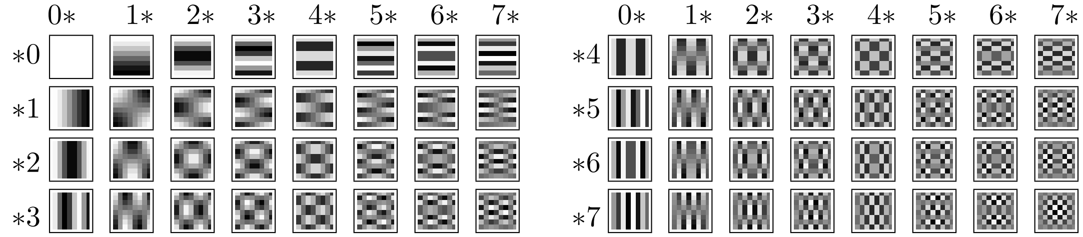
$$F = \sum_{n=0}^{7} \sum_{m=0}^{7} a_{nm} \phi_{nm} \quad \text{with} \quad \phi_{nm}(k, l) = \varphi_n(k) \varphi_m(l) \tag{2.72}$$

and

$$a_{nm} = \frac{\alpha_n \alpha_m}{64} \sum_{k=0}^{7} \sum_{l=0}^{7} F(k, l) \phi_{nm}(k, l). \tag{2.73}$$

Hence each coefficient requires a sum of 64 values and the computation by brute force of $\{a_{nm}\}_{n,m=0}^{7}$ requires roughly $64^2 = 4096$ operations per block. We have $W/8 \times H/8$ blocks and the resulting number of operations for a photo is affordable (see the comments at the end of this subsection).

We can read (2.72) saying that any 8×8 image represented by F can be written as a superposition of harmonics ϕ_{nm}. If we assign conventionally -1 to black and 1 to white, these harmonics can be seen as the following fundamental images:



For smooth blocks, Lemma 2.2.1 (recall that DCT is DFT forcing periodicity) implies that a_{nm} is small for the higher values of n and m. A usual photo contains many gradients, then for most blocks we could have a good reconstruction using only the a_{nm}'s with small index. On the other hand, it is clear that from a distance it is easier to distinguish an internal structure in the stripes of ϕ_{04} than in the checkerboard of ϕ_{77}. There are also effects related to the neurology of vision. This suggests to use a nonuniform quantizer giving few levels in general for higher values of n and m. In practice, one fixes a *quantization matrix* $Q \in \mathcal{M}_{8\times 8}(\mathbb{Z})$ and applies the uniform quantizer (2.13) with $\Delta = q_{nm}$. In other words, the quantization is

$$(2.74) \qquad a_{nm} \longmapsto q_{nm}\widetilde{e}_{nm} \qquad \text{with} \quad \widetilde{e}_{nm} = \lfloor q_{nm}^{-1} a_{nm} + 1/2 \rfloor.$$

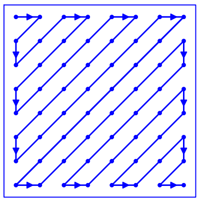
In this way, smaller values of q_{nm} give less spaced levels and hence a finer approximation to a_{nm}. The quantization matrix is not forced by the standards of JPEG format. In principle one could change it manipulating the header of the file but in reality everybody (every software) uses blindly the recommended quantization matrix

$$(2.75) \qquad Q = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}.$$

According to the official documentation [CCI91, Annex K] this and another quantization matrix that we shall see later "are based on psychovisual thresholding and are derived empirically [. . .]. These tables are provided as examples only and are not necessarily suitable for any particular application" and "If these quantization values are divided by 2, the resulting reconstructed image is usually nearly indistinguishable from the source image". As expected, the northwest triangle contains smaller numbers than the southeast triangle because the former corresponds to more visible frequencies. A curious property is that Q is only approximately symmetric. Probably it reflects a minor asymmetry in our vision.

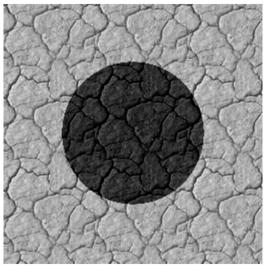
The resulting integers $\widetilde{e}_{nm}$ are stored following a zigzag path

$$\widetilde{e}_{00},\ \widetilde{e}_{10},\ \widetilde{e}_{01},\ \widetilde{e}_{11},\ \widetilde{e}_{20},\ \widetilde{e}_{30},\ \widetilde{e}_{12},\dots,\ \widetilde{e}_{77}. \tag{2.76}$$

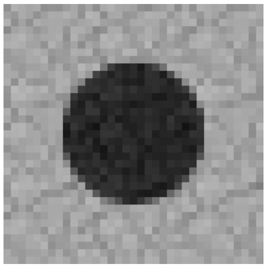


This finite sequence is very likely to be few numbers followed by a majority of zeros. It is easy to understand that a file with a lot of zeros admits an efficient lossless compression. We do not enter into the details of the compression here. We only mention that even a primitive method like RLE (Run Length Encoding) applied to the zeros i.e., substituting strings of zeros by their lengths, gives a noticeable result. In the case of JPEG, RLE is combined with other method (in practice *Huffman coding*).

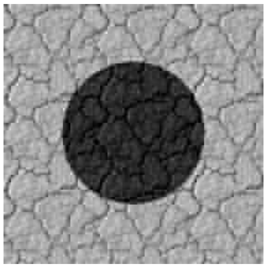
Perhaps you think that assuming that turning most of the a_{nm}'s into zeros without losing a lot in quality is wishful thinking. If you do some experiments with no quantization, just cropping the discrete Fourier expansion (2.72), it will amaze you how much information can be removed with little visual impact. If you do not want to do any experiment you can still look at the following images with original size 280 × 280.



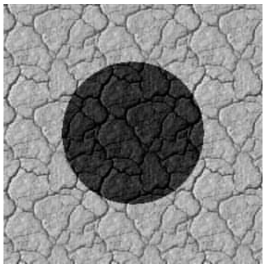
original



$\mathsf{a}_{\mathsf{nm}} = 0,\ \mathsf{n}+\mathsf{m} \geq 1$



$\mathsf{a}_{\mathsf{nm}} = 0,\ \mathsf{n}+\mathsf{m} \geq 3$



$\mathsf{a}_{\mathsf{nm}} = 0,\ \mathsf{n}+\mathsf{m} \geq 6$

In the second image we only keep a_{00} and then we see the blocks as thick pixels. In the third figure we keep 6 Fourier coefficients, it is less than 10% of the information, and we have a faithful idea about the original. Finally, in the last image with about 33% of the information we get something difficult to distinguish from the original. If you think that this example is made up, it is just the opposite because the furrows cause a bad performance of Fourier analysis. If you try a typical photo instead of this artificial small sized composition you will be more amazed.

Let us now consider the treatment of colors. In Computer Science, colors are commonly expressed by their coordinates in the *RGB color space*. These coordinates take values in $\{0, 1, 2, \dots, 255\}$ and indicate how much of *R*ed, *G*reen and *B*lue contains the color. For instance, $(255, 0, 0)$ is pure red, $(0, 0, 0)$ is black, $(255, 255, 255)$ is white and in general (n, n, n) are all possible gray tones. In total there are $256^3 = 16777216$ colors. When the coordinates are nearby the colors are indistinguishable (at least by me) at naked eye. The choice of red, green and blue as primary colors is related to human perception and probably to the available technologies. As an aside, the difficulties to make cheap blue LEDs (light-emitting diodes) stopped during decades this promising technology for full color LED displays that we see everyday[5].

A problem with the RGB color space is that it is not uniform with respect to our perception. When the *brightness*, the arithmetic mean of the coordinates, is smaller we have more difficulties to distinguish colors and it also depend on the color, for instance, we are less sensitive to changes in blue. In different applications one change the RGB color space by other color spaces (arguably the most known after RGB is CMYK used in professional printing). In JPEG and in many applications related to video, the color space is $\mathrm{YC_bC_r}$, where Y is the *luminance* and the pair $(\mathrm{C_b}, \mathrm{C_r})$ is the *chrominance* (or both are the *chroma components*). The relation between both color spaces is given by [Sal02, p.145]

$$\begin{pmatrix} \mathrm{Y} \\ \mathrm{C_b} \\ \mathrm{C_r} \end{pmatrix} = \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix} + \frac{1}{256} \begin{pmatrix} 77 & 150 & 29 \\ -44 & -87 & 131 \\ 131 & -110 & -21 \end{pmatrix} \begin{pmatrix} \mathrm{R} \\ \mathrm{G} \\ \mathrm{B} \end{pmatrix}. \tag{2.77}$$

In practice this relation is not completely exact because we adjust the result to fit in a byte.

Roughly Y represent the achromatic brightness of an image for our perception and $\mathrm{C_b}$ and $\mathrm{C_r}$ the color information. It turns out that we are much more sensitive to Y than to chrominance (in fact for chrominance usually only a part of the pixels are considered, see the comments below).

For B/W images the RGB colors are (n, n, n) and (2.77) implies $\mathrm{Y} = n$, $\mathrm{C_b} = 128$ and $\mathrm{C_r} = 128$, showing that Y carries the achromatic information. After a normalization subtracting 128, the chroma components become zero. The Y component is processed as described above. For the other components, one repeats the procedure but in a careless way, meaning that we take instead of (2.75) a quantization matrix with in general larger values. The recommendation in [CCI91, Annex K] is

$$Q = \begin{pmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{pmatrix}. \tag{2.78}$$

The outcome is that the corresponding values $\widetilde{e}_{nm}$ to be stored are more likely to be zero than the corresponding one for Y.

The lines above show the main aspects of the algorithm but the implementation involves many other technical points that probably will be of interest to you only if you have an itch for engineering. We mention here three of them. Firstly, not too much precision is need

[5] The scientists that created the first blue LEDs were awarded with the Nobel prize in 2014.

when computing a_{nm} because we are going to quantize the results, then one can speed up the $64W \times H$ operations using different formats of numbers. When exporting an image as JPEG with GIMP, you have some control on it in `Advanced options>DCT method`. Secondly, a_{00} is typically very large in comparison with the rest of the values because it gives the sum of the colors of the block. To avoid to store large numbers after quantization, it is normalized subtracting certain quantity. This coefficient is called the *DC component* and the rest the *AC component*, following the electrical terminology AC/DC (Alternating Current/Direct Current). A third technical point is that as $\mathrm{C_b}$ and $\mathrm{C_r}$ are less important than Y, when treating the former components it is usual to take into account only one out of each two or each four pixels. In GIMP you can choose among four possibilities in `Advanced options>subsampling`.

If we examine critically the format, the subdivision into 8×8 blocks to apply Fourier analysis on them is somewhat arbitrary (and in part related to the capabilities of 1990s computers). It is a kind of balance, because taking very small blocks, 1×1 in the limit, is like doing nothing and choosing larger blocks we take the risk of including sharp changes of the color that avoid the well behavior of Fourier analysis. Summing up, we would like to enlarge these blocks in zones with gradients and perhaps to reduce them in zones with edges.

From the theoretical point of view, it would be convenient to have a so to speak adaptable Fourier analysis able to work at different scales. This theoretical aim is achieved with wavelets as we shall see in the next chapter. The idea led to create in 2000 the JPEG2000 format that employs wavelet transforms. Although it is superior to JPEG it has not been successful and very seldom cameras, scanners, browsers and widespread software support it. One can cook many explanations for it, a sound one is that it appeared when JPEG was widely used (JPEG2000 is not backward compatible) and with the features of modern computers more compression or more quality is no longer a primary issue (see the final comments in [Aus08]).

Suggested Readings. You can find information about JPEG in the short note [Aus08] and in the books [Sal02] and [GWE03, §8.5.1]. The website of the Joint Photographic Experts Group `https://jpeg.org/` contains references and information. If you like to go to the source without reading long documentation files, you will enjoy the 1991 paper [Wal91].

2.2.3 Linear filters: from analog to digital

Given a signal $f : \mathbb{R} \longrightarrow \mathbb{R}$, say smooth and compactly supported or nicely decaying, in many applications one wants to attenuate the contribution of certain ranges of frequencies while keeping or amplifying the contribution of others. Mathematically we can write this as $\widehat{f}(\xi) \longmapsto a(\xi)\widehat{f}(\xi)$, which is called a *multiplier* in harmonic analysis by obvious reasons and a (linear) *filter* in engineering by obvious reasons too. If $a = \widehat{h}$ then by (1.77) we have that filtering is convolving:

$$f \longmapsto h * f \qquad \text{with} \quad a = \widehat{h}. \tag{2.79}$$

In the digital setting the things go in the same way and alike linear filters applied to a discrete signal $\{x_n\}_{n\in\mathbb{Z}}$ act as

$$x_n \longmapsto y_n \qquad \text{with} \quad y_n = \sum_{k\in\mathbb{Z}} h_k x_{n-k}. \tag{2.80}$$

We assume that we have a lot of bits to represent the values of x_n then quantization is not an issue here and we consider $x_n \in \mathbb{R}$. In the actual numerical computations we only manage a finite number of terms in the summation, hence we are forced to consider $h_k = 0$ except for a finite number of indexes. The size of this number is important for the performance. In engineering a filter of this kind is called a *FIR filter* where FIR stands for *finite impulse response*. Very often it is imposed $h_k = 0$ for $k < 0$ meaning that the filter is *causal*: If n represents time, the output values do not depend on future input values.

Other popular filters in practice for discrete signals as before, are the *IIR filters*, which are also mathematically linear. As you are guessing, IIR stands for *infinite impulse response*. The difference is that we admit a dependence on previous values of the output in the following form:

$$x_n \longmapsto y_n \qquad \text{with} \quad y_n = \sum_{k\in\mathbb{Z}} h_k x_{n-k} + \sum_{k\in\mathbb{Z}^+} b_k y_{n-k}, \tag{2.81}$$

where $\{b_k\}_{k\in\mathbb{Z}^+}$ is nontrivial and, as $\{h_k\}_{k\in\mathbb{Z}}$, is zero except for a finite number of indexes. For instance if $x_n = f(n)$, the discrete derivative $y_n = x_n - x_{n-1}$ is a FIR filter while the discrete integral given by the trapezoid rule $y_n = y_{n-1} + \frac{1}{2}\big(x_n + x_{n-1}\big)$ is an IIR filter.

In mathematical circles the names *nonrecursive filters* and *recursive filters* are more informative. They also appear in the literature, probably under of the influence of [Ham89], although they are less common.

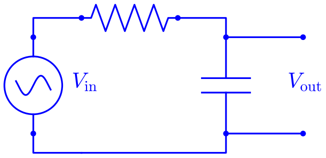
One can consider other filters with nonlinear formulas or involving multidimensional signals. We shall see examples in the context of image processing (in fact the JPEG format in the previous subsection involves a kind of nonlinear filtering by the quantization). In this subsection we keep in mind audio signals and in the digital case we think that $\{x_n\}_{n\in\mathbb{Z}}$ is the result of sampling one of these signals with sampling frequency $\nu_s = 1$.

A primary goal for both, the analog and the digital setting, is to trim the frequencies beyond certain *cutoff frequency* ν_c. This is called a *low-pass filter*. Its theoretical construction in the analog case is very simple, just taking in (2.79) $a = \chi_F$ the characteristic function of $F = [-\nu_c, \nu_c]$. In this way, a low-pass filter is the operator

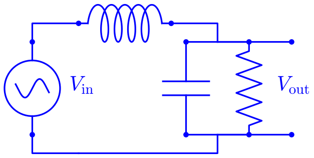
$$Lf(t) = \int_{-\infty}^{\infty} \operatorname{sinc}(2\nu_c u) f(t-u)\, du. \tag{2.82}$$

This is called an idealized low-pass filter and as the name suggests it has not so many realizations beyond notebooks. In practice, as in the digital setting, we can only manage the signal for a limited interval of time T then in the integral f is replaced by $f\chi_T$ (recall the Papoulis-Gerchberg algorithm). An interesting theoretical problem with practical applications [Sle83] is to study the eigenfunctions of this modified operator. They are called the *prolate spheroidal wave functions*. The one with the smallest eigenvalue ("energy") gives the best way of fighting uncertainty principle if we put limits simultaneously to the frequencies and the time.

Our final aim here is to study the design of digital low-pass FIR filters but let us imagine that we are at the early times of analog electronics to introduce some basic concepts and motivation. From (1.20) we extract the idea that capacitors attenuate the current for low frequencies of the voltage and inductors do the same with large frequencies because oscillatory functions have large derivatives and small integrals. We consider the following circuits to devise an approximation of a low-pass filter.



First order low-pass filter



Second order low-pass filter

For the first circuit the relevant Kirchhoff's law is $V_R + V_C = V$ where V is V_{in}, the input voltage. For the second one we have $V_L + V_R = V$ with $V_R = V_C$ and $I = I_1 + I_2$ where I_1 and I_2 are respectively the currents through the capacitor and the inductor. Then (1.20) gives respectively the equations

$$RI' + C^{-1}I = V' \qquad \text{and} \qquad RCLI_2'' + LI_2' + RI_2 = V, \tag{2.83}$$

and the output voltage V_{out} is $V_C = C^{-1}\int I$ in the first case and $V_R = RI_2$ in the second case.

We know by the theory of linear ODE's with constant coefficients that if $V_{\text{in}} = A_{\text{in}}e(\nu t)$ is the complex version of the input voltage, to treat simultaneously sine and cosine, then the output voltage has the same frequency $V_{\text{out}} = A_{\text{out}}e(\nu t - \delta)$. Here $A_{\text{in}}, A_{\text{out}} > 0$ are amplitudes. For the first circuit $V_{\text{out}}' = C^{-1}I$ and then the equation can be rewritten as

$$RCV_{\text{out}}' + V_{\text{out}} = V_{\text{in}} \quad \text{that implies} \quad A_{\text{out}}\big(RC(2\pi i\nu) + 1\big)e(-\delta) = A_{\text{in}} \tag{2.84}$$

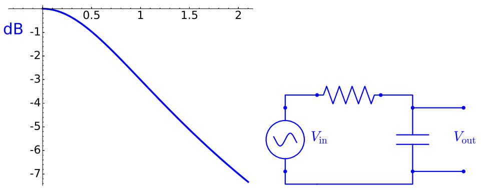
If $A_{\text{in}} = 1$, taking absolute values $A_{\text{out}} = |2\pi\nu RCi + 1|^{-1}$. A similar argument works for the second circuit. In both cases we can write A_{out}, respectively, in the form

$$\Big|1 + i\frac{\nu}{\nu_c}\Big|^{-1} \text{ with } \nu_c = \frac{1}{2\pi RC} \qquad \text{and} \qquad \Big|1 - \frac{\nu^2}{\nu_c^2} + \frac{2\pi\nu L}{R}i\Big|^{-1} \text{ with } \nu_c = \frac{1}{2\pi\sqrt{LC}}. \tag{2.85}$$

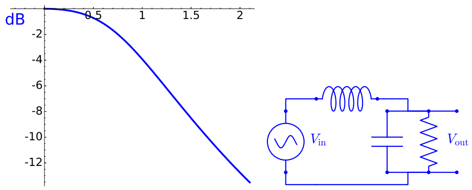
Then A_{out} is approximately $A_{\text{in}} = 1$ for ν small and it decays to 0 for ν much larger than ν_c. Namely, like ν^{-1} in the first case and like ν^{-2} in the second case. Then these circuits constitute a primitive form of low-pass filters. The usual measure of the performance in amplitude is trough the *decibels*, abbreviated dB, defined by the formula

$$\text{number of dB} = 20\log_{10}\frac{A}{A_{\text{ref}}} \tag{2.86}$$

where A_{ref} is an amplitude used as reference. An explanation of this funny definition is that for waves like the sound the energy or the intensity depend on the square of the amplitude and our senses respond logarithmically (Weber's law) then $\log_{10}(A/A_{\text{ref}})^2$ is a natural "unit" called *bel* but it is too coarse and decibels are preferred. For $\nu_c = 1$ (and $R/L = 0.25$ in the second case) using $A_{\text{ref}} = A_{\text{in}} = 1$ as reference and $A = A_{\text{out}}$ we get the following plots for the performance in decibels:



First circuit



Second circuit

A perfect low-pass filter would take the value 0 dB for $0 \le \nu \le \nu_c$ and $-\infty$ for $\nu > \nu_c$ (negative values are symmetric). The performance is not admissible with the common standards and could be improved with more complicated circuits [Win02, Ch.2]. Any circuit using resistors, capacitors and inductors gives a linear ODE as before and A/A_{ref} will be the absolute value of a rational function. So the mathematical problem is to approximate the characteristic function of an interval by such a function.

After this training let us move to the digital setting. If we have a signal $x_n = e(\nu n)$, a pure tone of frequency ν, then when we substitute in (2.81) the output signal will be of the form $y_n = A_{\text{out}} e(\nu n - \delta)$ where

$$A_{\text{out}} = \big|H(e(\nu))\big| \qquad \text{and} \qquad H(z) = \frac{\sum_k h_k z^{-k}}{1 - \sum_k b_k z^{-k}}. \tag{2.87}$$

Here there are not differential operators linked to electronic components but the result is alike. Note that H is a rational function because h_k and b_k vanish except in a finite number of cases. This rational function is called the *transfer function*. An FIR filter is formally the same as an IIR filter with $b_k = 0$ then $H(z^{-1})$, the transfer function evaluated at z^{-1}, is a polynomial for causal FIR filters. On the other hand $H(e(\nu))$ is called the *frequency response*[6] because it really indicate how the amplitude and the phase change when a pure tone of fixed frequency passes trough the filter.

Let us focus on FIR filters. The mathematical problem to construct a digital low-pass filter is to find coefficients h_k such that $\big|H(e(\nu))\big|$ is an approximation of the characteristic function of $|\nu| < \nu_c$. As we assumed that the sampling frequency was $\nu_s = 1$ then ν_c is less than the Nyquist frequency 1/2. The constraint for this approximation is that for practical reasons we want as less nonzero coefficients h_k as possible. Let us sacrifice the causal nature of our filter in favor of the symmetry and let us put $h_k = h_{-k}$. A technical justification is that very often we can store digitally a part of the signal to move it to the future (anyway, the subsequent analysis can be easily adapted to the causal case). In this

[6] In the literature it is commonly employed the angular frequency $\omega = 2\pi\nu$ and with a clash of notation difficult to admit for a mathematician $H(z)$ means the transfer function and $H(\omega)$ the frequency response.

situation our target is

$$\Big|h_0 + 2\sum_{k=1}^{N} h_k \cos(2\pi k\nu)\Big| \approx \chi_F(\nu) \qquad \text{with} \quad F = [-\nu_c, \nu_c]. \tag{2.88}$$

The analogue of (2.82) is that we would have an exact approximation taking

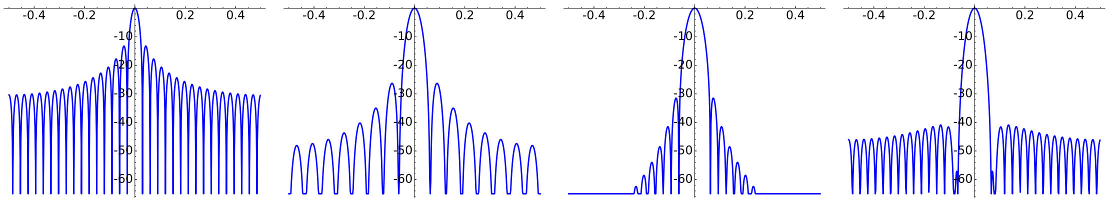
$$h_k = 2\nu_c \operatorname{sinc}(2\nu_c k) \qquad \text{and} \qquad N = \infty. \tag{2.89}$$

We cannot take $N = \infty$ in the same way that we said after (2.82) that we should replace f by $f\chi_T$, but this procedure there and also here has bad consequences on the regularity and then in the decay. A natural solution is introducing *windows*. We will study more closely the role of the windows in the continuous setting that pioneered the multiresolution analysis. Now we focus on the digital setting. A window is a choice of some weights w_k such that $\sum_{|k|\le N} w_k e(kx)$ approximates a Dirac delta at the origin. We assume $w_k = w_{-k}$. By (1.77), given an "ideal filter" $\{h_k\}_{k\in\mathbb{Z}}$ the convolution of the signal with $\{h_k w_k\}_{k=-N}^{N}$ gives almost the same result and is a realizable digital filter.

Actually a problem of this kind already appeared when studying the Fejér kernel in Theorem 1.2.5. Fejér kernel corresponds to take $w_k = \big(1 - |k|/N\big)_+$ and it is sometimes called the *triangular window* while doing nothing, the sharp cut $w_k = 1$ for $|k| \le N$, is called the *rectangular window*. These windows are very seldom used because for applications are better alternatives. Two of them with a quite similar aspect are the *Hann window* and the *Hamming window*, given respectively by the formulas (for $|k| \le N$)

$$w_k = \frac{1}{2} + \frac{1}{2}\cos\frac{\pi k}{N} \qquad \text{and} \qquad w_k = \frac{25}{46} + \frac{21}{46}\cos\frac{\pi k}{N}. \tag{2.90}$$

Let us see for $N = 16$ how much $f(x) = \sum_{k=-N}^{N} w_k e(kx)$ differs from a Dirac delta plotting it in decibels, $20\log_{10}|f(x)/f(0)|$, for the windows that we have mentioned.



Rectangular Triangular Hann Hamming

In general terms the windows f shows an oscillatory behavior and one wants to have a main lobe as thin as possible and a peak side lobe as small as possible. By the uncertainty principle the width of the main lobe goes like N^{-1} but the constant of proportionality varies. The rectangular window has the smallest main lobe width but the size of the other lobes is too large. In a moment we will see the relevance of these values in the design of a digital low-pass filter. In (2.90) the coefficients for the Hamming window are a little perturbation of those of the Hann window and are chosen to cancel the first side lobe. This idea can be extended using trigonometrical polynomials of higher degree. For instance, the *Blackman window* is

$$w_k = \frac{3969}{9304} + \frac{1155}{2326} \cos\frac{\pi k}{N} + \frac{715}{9304} \cos\frac{2\pi k}{N} \tag{2.91}$$

and the funny coefficients, that commonly are approximated by 21/50, 1/2 and 2/25, are adjusted to introduce zeros canceling the first side lobes.

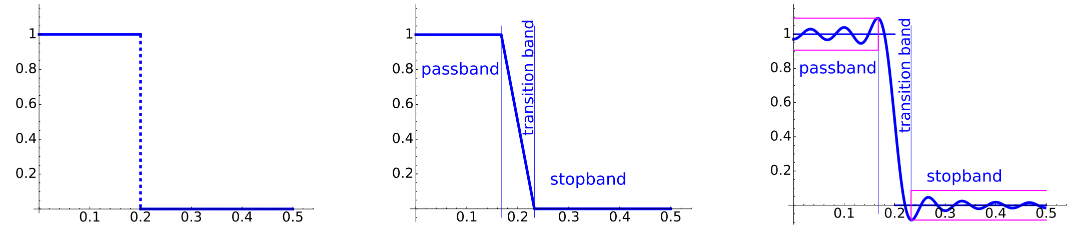
There exists a digital version of the prolate spheroidal wave functions that we have mentioned before and it gives rise to a *prolate spheroidal window* that maximizes the proportion of energy (L^2 norm) in a frequency range. The drawback is that its calculation leads to ill conditioned numerical problems [Sle78] [PB72] [VBM96]. J.F. Kaiser introduced in [Kai74] a family of windows that approximates this, in some way optimal, window. This family, depending on α, is given by

$$w_k = \frac{I_0\big(\alpha\sqrt{1-(k/N)^2}\big)}{I_0(\alpha)} \quad \text{for } |k| \le N \qquad \text{with} \quad I_0(x) = \sum_{k=0}^{\infty} \frac{x^{2k}}{2^{2k}(k!)^2}. \tag{2.92}$$

The function I_0 is an instance of the modified Bessel functions of the first kind[7]. In the degenerate case $\alpha = 0$ this is the rectangular windows, because $I_0(0) = 1$, and for $\alpha = 5.4$ the main lobe is very similar to that of the Hann window but with a smaller side lobe peak.

We address now the design of a digital FIR low-pass filter that epitomizes the issues in digital FIR filter design. With IIR we would have more freedom because the transfer function would be a rational function instead of a polynomial divided by a power, consequently the results can be improved.

As we have seen an ideal low-pass filter involves a sharp cut of the frequency response. the frequencies $0 \le \nu \le \nu_c$ remain intact, this is the *passband*, while the frequencies $\nu_c < \nu \le 1/2$ disappear, this is the *stopband*. Recall that we assume symmetry and we have normalized the sampling according $\nu_s = 1$ then we only consider the interval $[0, 1/2]$. In an oscilloscope we would never see a discontinuous line, in practice there is a *transition band* in the middle in which we have a continuous approximation of the original discontinuous function.



Ideal Smoothed Real

[7] In the original paper [Kai74] he does not give clue about how he got I_0. The prolate spheroidal wave functions and the Bessel functions, in particular I_0, satisfy a differential equation but I do not see any approximation of one by the other. Another possibility is that he considered its Fourier transform, that is a simple function in part mentioned in the title of the paper.

With a window we are cutting the Fourier series of the ideal filter and we know after Gibbs phenomenon that some ripples appear causing the values in the passband and the stopband to be contained in $|y-1|<\delta$ and $|y|<\delta$ where δ is the maximal amplitude of these ripples. Keep in mind the Dirichlet kernel (1.40) that corresponds to the rectangular window. It has a positive main lobe in the interval $|x|<1/(2N+1)$ hence if f is the profile of the ideal filter $f*D_N(x)$ essentially remains monotonic in the interval $|x-\nu_c|<1/(2N+1)$. Then the width of the transition band is essentially that of the main lobe. The calculation of $f*D_N\big(\nu_c-1/(2N+1)\big)$ moves entirely the main lobe to the passband but its area is not exactly 1, as it should for a Dirac delta, and this excess of area causes the lump in Gibbs phenomenon. This was the idea under the proof of Proposition 1.2.11. The excess of area depends on the other lobes because we know $\int_{\mathbb{T}} D_N=1$ and we expect the closest lobes to give the greatest contribution because of the decay (see [Har78] for more information).

With an evil mathematical mind you can contradict the meaning of these not well defined concepts, considering for instance a main lobe with an abrupt change of curvature. The engineers use very often as a measure of the width of the main lobe the *bandwidth*: the range of frequencies, considering also negative values, in units of the inverse of the number of points corresponding to the part in which the main lobe is above -3dB. As a measure of the contribution of the rest lobe it is considered the *attenuation* given by the peak side lobe. The results for the windows that we have considered are summarized in the following table[8]

	Rectangular	Triangular	Hann	Hamming	Blackman
-3 dB Bandwidth	0.88	1.28	1.44	1.29	1.61
Peak side lobe	-13.3 dB	-26.5 dB	-31.5 dB	-42.2 dB	-68.1 dB

Note that the triangular window has almost the same bandwidth as the Hamming window and a much larger peak side lobe. This explains why it is not employed in practice.

After the previous considerations the width of the transition band, usually denoted by ΔF multiplied by $2N+1$, the number of points, gives approximately a constant that depends on the width of the main lobe. We can define ΔF "mathematically" putting $(2N+1)\Delta F=Kc_{-3}$ where c_{-3} is the constant of the -3 dB banwidth corresponding to the window and K is a fixed constant (for instance 2 is more or less natural). On the other hand, δ (the maximal amplitude of the ripples) is controlled by the peak side lobe.

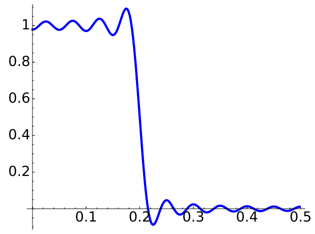
Once we have chosen a window and a value of N that suit our needs, recalling (2.80) and the formula (2.89) for the ideal filter, we have that the digital low-pass filter is given by

$$(2.93)\qquad x_n\longmapsto y_n,\qquad y_n=2\nu_c\sum_{k=-N}^{N}w_k\,\mathrm{sinc}(2\nu_c k)x_{n-k}.$$

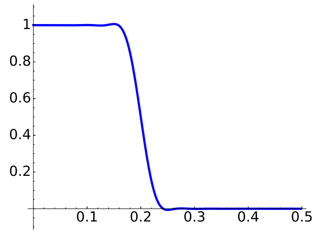
In the following figures it is plotted the frequency response $H\big(e(\nu)\big)$ for the rectangular, the Hann and the Hamming windows with $N=20$ and $\nu_c=0.2$. In the first case the

[8] It corresponds to my experiments. There are slight variations with respect to [Har78], [PM96] and [Win02] but these references do not completely agree among them. One possible source of this disagreement is that the dependence of the bandwidth on N is not completely linear and Blackman and Hamming windows are usually approximated with simpler coefficients.

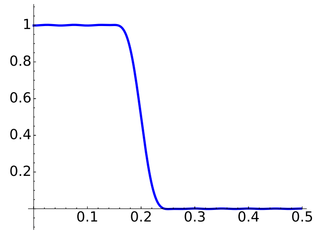
Gibbs phenomenon is apparent but note the narrower transition band width related to its smaller constant. The tiny ripple before the jump in the Hann window is not visible in the Hamming window.



Rectangular



Hann



Hamming

We finish this subsection with two methods that allow to tune the parameters.

If you want to adjust δ and ΔF as you please without checking a long list of available windows, Kaiser already did the work for you. He defined $A = -20\log_{10}\delta$ and he found the following experimental approximate formulas for the parameters in (2.92)

$$\alpha = \begin{cases} 0.1102(A-8.7) & \text{if } A \geq 50, \\ 0.5842(A-21)^{0.4} + 0.07886(A-21) & \text{if } 21 \leq A < 50, \\ 0 & \text{if } A \leq 21, \end{cases} \qquad N \geq \frac{A-7.95}{28.72\Delta F}. \tag{2.94}$$

Of course one wants to take N as small as possible to minimize the computations. The case $A < 21$ has a simple interpretation, it corresponds to $\delta = 0.089$, the 8.9% of the Gibbs phenomenon and in this case the rectangular window ($\alpha = 0$) is enough. If we want to reduce the Gibbs phenomenon to one fourth of its value, then we have to take $A = 33.0057$ and according to the formula in the middle we must choose $\alpha = 2.5255$. If we want $\Delta F \leq 0.02$ then we have to take $N > 43.62$. An experiment shows that for $N = 44$ the maximum of the frequency response is 1.0208 and the minimum -0.0203 then we have the expected δ with accuracy. These values are reached at $\nu = 0.1855$ and $\nu = 0.2148$ that is coherent with $\Delta F \approx 0.02$.

Imagine now that you fix the passband and the stopband, which define a set of the form $\mathcal{F} = [0,\nu_1] \cup [\nu_2, 1/2]$ and for a fixed N you want to determine an optimal filter, meaning one with minimal δ. This can be translated into computing real numbers h_0, h_1, …, h_N such that

$$\max_{\nu\in\mathcal{F}} \Big| h_0 + 2\sum_{k=1}^{N} h_k \cos(2\pi k\nu) - \chi_{[0,\nu_c]}(\nu) \Big| \tag{2.95}$$

is minimum. This recalls to the famous optimization problem for polynomials:

$$\min_{P\in\mathcal{P}_N} \max_{x\in[-1,1]} \big|P(x)\big| = 2^{1-N} \quad \text{where} \quad \mathcal{P}_N = \{P \text{ monic with } \deg P = N\} \tag{2.96}$$

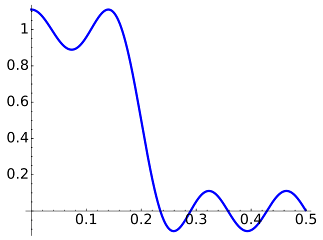
and the minimum is reached by $P = 2^{1-N}T_N$ where T_N is the N-th *Chebyshev polynomial* defined in $[-1,1]$ by

$$T_N(x) = \cos(N \arccos x). \tag{2.97}$$

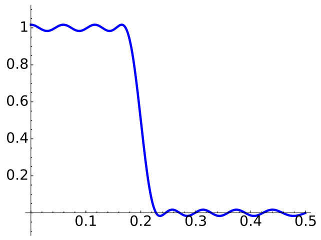
Believe or not, this is a polynomial (show it!) and clearly its maximum and minimum is reached at $N+2$ points alternating the values -1 and 1. Any other polynomial P with the same leading coefficient as T_N (which is 2^{N-1}) and having smaller maximum in absolute value would give at least N zeros in $T_N - P$ by the intermediate value theorem and this contradicts $\deg(T_N - P) < N$ and proves (2.96). In approximation theory there exists an enhanced converse called the *Chebyshev oscillation theorem* [Dav75, Th.7.6.2] or sometimes in a stronger form the *alternation theorem* [Bra86, §3.5] that implies that for an optimal polynomial approximation there are $N+2$ points at which the error alternates a maximum and a minimum with the same absolute value. This applies also to our problem (2.95) because with the change of variables $2\pi\nu = \arccos x$ we have a linear combination of Chebyshev polynomials. This is the basis of the *Parks-McClellan algorithm*.

The idea is very simple (see the original paper [PM72] for the implementation). One chooses $N+2$ frequencies $\nu_0, \dots, \nu_{N+1} \in \mathcal{F}$ and finds coefficients $h_0, \dots, h_N$ such that the function D in the absolute value in (2.95) takes the same value at ν_j with alternating sign. This establishes $N+1$ linear equations, $D(\nu_0) = (-1)^j D(\nu_j)$ $0 < j \le N+1$, with $N+1$ unknowns, the h_j, but it is better to consider it as a problem in polynomial interpolation [PM72]. Once we have solved this problem, we update the frequencies ν_j choosing them as the points at which the interpolation error reaches local extrema and one repeats the procedure.

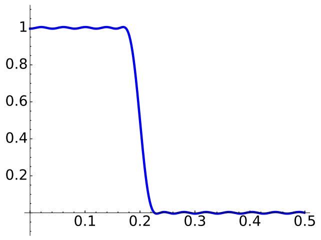
For instance, if we want to design an optimal low-pass filter with $\nu_c = 0.2$ and $\Delta F = 0.05$ then $\mathcal{F} = [0, 0.175] \cup [0.225, 0.5]$ and running the algorithm for $N = 7$, $N = 16$ y $N = 24$ we obtain the optimal filters, with the indicated maximal error δ in $\mathcal{F}$, depicted in the following figures



N = 7, $\delta \approx 0.09$



N = 16, $\delta \approx 0.017$



N = 24, $\delta \approx 0.004$

To count properly the $N+2$ extrema we have to consider also those appearing in the border of the transition band. So in the first figure we must count four local maxima plus one at the beginning of the stopband and three local minima plus one at the end of the passband.

Suggested Readings. The book [Ham89] on digital filters is a classic, one of the firsts on this topic, and even though still today constitutes a very interesting reference. There is a nice story about this text, according to his famous author R.W. Hamming: "I knew very little about digital filters, and, furthermore, I was not really interested in them". In principle the author or coauthor would be his friend Kaiser but finally his contribution was "[to] educate me over lunches in the restaurant". A good point of [Ham89] is that tries to avoid the jargon. Of course, technology has changed a lot even from the last edition, and there are quite a number of modern books and lecture notes incorporating these changes, for instance [Win02] or [Cha11] (not centered in filters).

2.2.4 Basic linear filters for images

Image processing is a huge subject that recently has become very close to our daily experience. Nowadays many people with mild technological knowledge are able to deeply retouch a photo with a raster graphic editor or even a mobile app. The experts who worked hard in the infamous manipulation of the official photos of the Russian Revolution would be baffled if they knew how easy their task is now.

For the sake of simplicity we consider grayscale images where the gray levels vary between 0 (black) and 1 (white). In this way we ignore color and the most common quantization of the levels as integer numbers between 0 and 255, which was treated in a previous subsection. Nevertheless some references to the quantization will appear from time to time.

Summing up, for us an image of weight w pixels and height h pixels is a function

$$F\ :\ \big([0,w)\times[0,h)\big)\cap\mathbb{Z}^2\longrightarrow[0,1]. \tag{2.98}$$

An alternative representation is like a matrix $h\times w$ with elements in $[0,1]$. A *linear filter* is something given by a discrete convolution as in (2.80)

$$F(m,n)\longmapsto\sum_{k,l\in\mathbb{Z}}H(k,l)F(m-k,n-l). \tag{2.99}$$

We restrict ourselves to filters H with a finite support (recall the FIR filters) and it is common to identify H with the matrix of the values taken on a square containing its support, although there is certain ambiguity in it because of the possibility of translations. A first problem is that the result of (2.99) is not exactly well defined because we cannot assure $(m-k,n-l)\in[0,w)\times[0,h)$ for (k,l) in the support of H. In the significant cases the support of H is small in comparison to the dimensions of the image and the problem only affects to the pixels close to the boundary. The practical solution is to invent for calculations a virtual flange of pixels around the image. There is a number of techniques to assign values (gray tones) to these fake pixels. One which is quite natural and that we will use in the examples is to consider mirror reflections of the image trough the edges. In this way if we need $F(-1,2)$ in a calculation, we borrow its value from $F(1,2)$.

A basic example of linear filter is to take the average of the values of a pixel and its eight neighboring pixels: $H(k,l)=1/9$ for $k,l\in\{-1,0,1\}$ and $H(k,l)=0$ otherwise. Its matrix is

$$B=\frac{1}{9}\mathbf{1}_3\qquad\text{with}\quad\mathbf{1}_d=\text{the } d\times d \text{ matrix of ones.} \tag{2.100}$$

The effect of this filter on the image is a short distance blurring. In image processing blurring is a basic operation and there are several ways to do it. Clearly a natural generalization is to change B by $d^{-2}\mathbf{1}_d$. The amount of blurring increases with d. Other well known variants involving different weights in the average correspond to the matrices

$$G=\frac{1}{16}\begin{pmatrix}1&2&1\\2&4&2\\1&2&1\end{pmatrix}\qquad\text{and}\qquad M=\frac{1}{3}\begin{pmatrix}1&1&1\\0&0&0\\0&0&0\end{pmatrix}. \tag{2.101}$$

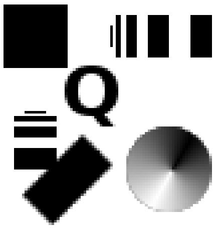
The first filter is the 3×3 instance of the *Gaussian blur* that approximate the result of a continuous convolution with the Gaussian kernel $(2\pi\sigma^2)^{-1}e^{-(x^2+y^2)/(2\sigma^2)}$. When $x^2 + y^2$ is much larger than σ^2 the value is negligible and then the approximation by a compactly supported function is reasonable, so the matrix of the filter has a size proportional to σ^{-1} (see [Rus07, Ch.4] for references about its elements and [PP10, §4.2] for some simple calculations). A Gaussian blur gives more weight to the closer pixels and hence it changes less the image than the flat blur. The second filter in (2.101) is the simplest case of *motion blur* that represents blurring in only a direction, in this case horizontally. This is the kind of defocus we observe when taking photos of high speed objects.

The previous filters B, G and M have a "width" of 3 pixels which is difficult to observe with the typical size of photos we manage. As a matter of fact, if pixels are a too small unit for our needs[9] one can take $d \times d$ groups of pixels as the new unit and substitute the matrix H of a filter by

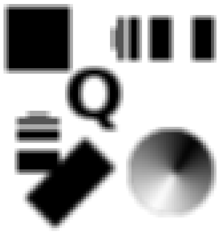
$$H \otimes B_d = \begin{pmatrix} h_{11}B_d & \dots & h_{1n}B_d \\ \vdots & \ddots & \vdots \\ h_{n1}B_d & \dots & h_{nn}B_d \end{pmatrix} \qquad \text{with} \quad B_d = \frac{1}{d^2}\mathbf{1}_d. \tag{2.102}$$

In case it rings a bell, $\otimes$ denotes the *Kronecker product*. Note that $B = B_3$ and the algebraic relation $B_n \otimes B_m = B_{nm}$. Grouping pixels involves blurring them into a cell having a uniform tone.

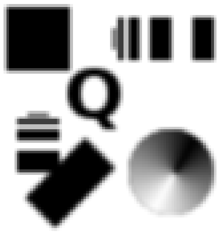
Let us see some examples. The sources of the following images have size 96×96 and the scale when reproduced here is such that if you see this page on a normal computer monitor keeping the 100% scale of an A4 sheet then a pixel here corresponds approximately to a pixel on your screen. You can check this prediction (or not) knowing that the separation from the smallest bar to the following bar is one pixel.



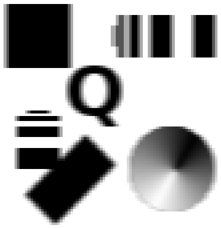
Original



Filter B



Filter G

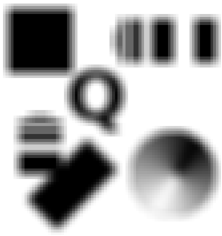


Filter M

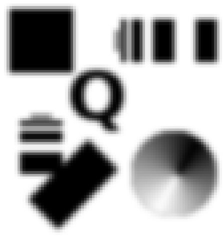
With this small size the filters B and G differ very little, but anyway it is possible to observe in some details that the second is neater, less blur. On the other hand, note that M preserves the horizontal borders of the rectangles adding something to both sides. We lose the symmetry between horizontal and vertical rectangles that we had with B and G.

[9] For instance in new generation cell phones the screen resolution is like 1000×2200, enough to represent quite faithfully in full screen the commonly available wallpapers for computers. The display area is around $85\,\text{cm}^2$, then there are more than 28000 pixels per cm^2 and this makes theoretically the length of a pixel less than a tenth of millimeter, in the limit of the visual acuity of many of us. On a normal computer screen the size of a pixel is around one quarter of millimeter.

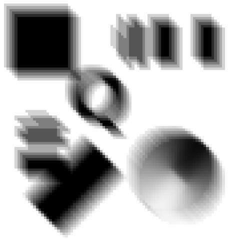
After this challenging practice for your next visit to the ophthalmologist, let us exaggerate the effect. If we change $B = B_3$ by B_5 then the average involves 25 pixels instead of 9 and the blur is more noticeable. If we look up in the literature the 5×5 matrix that is usually considered as a discrete approximation of a Gaussian kernel then we observe again a less blurred image because closer pixels have higher weights.



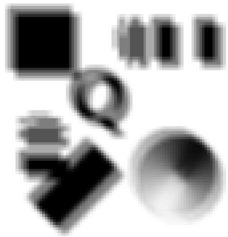
Filter B_5



Gaussian $d = 5$



Filter $\frac{1}{6}I_6$



Filter $\frac{1}{6}I_2 \otimes B_3$

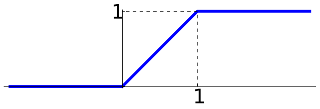
The third image shows the effect of a motion blur filter of width 6 with angle $3\pi/4$, represented by the diagonal of ones in I_6, where I_d denotes the unit matrix $d \times d$. Note that this is different than using I_2 after grouping the pixels in 3×3 blocks, as shown in the last image.

Still within the ambit of the 3×3 filters there are some artistic effects funnier than the humble blur. The following matrices correspond respectively to the *Laplacian filter*, the *sharpen filter* and the *emboss filter*:

$$(2.103) \qquad L = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}, \quad S = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad \text{and} \quad E = \begin{pmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix}.$$

Before explaining their effects note that they can give results outside the interval $[0,1]$ when applied to an image (2.98), this did not happen with the blur filters considered before (do you see why?) and there are two natural ways of managing this issue. The first one is *clamping* the values in such a way that the infra-black and ultra-white levels are considered black and white, respectively. In mathematical terms we apply to the results of the filter the clamping function

$$(2.104) \qquad c(t) = \min\big(\max(t,0),1\big) =$$



Other solution is scaling the minimal interval $[a,b]$ containing the results by way of a linear function

$$(2.105) \qquad T : [a,b] \longrightarrow [0,1], \qquad T(x) = \frac{x-a}{b-a}.$$

This is called *normalization* especially in the context of histograms that we will see later.

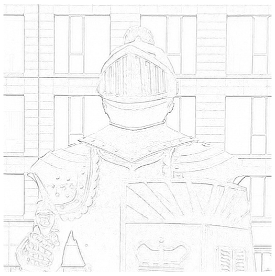
Coming back to the filters (2.103), to motivate L note the Taylor expansion

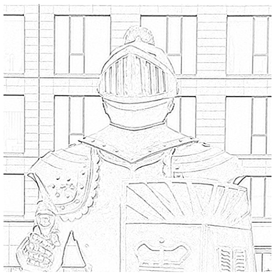
$$4f(0,0) - f(0,h) - f(0,-h) - f(h,0) - f(-h,0) = -h^2\nabla^2 f(0,0) + \dots \tag{2.106}$$

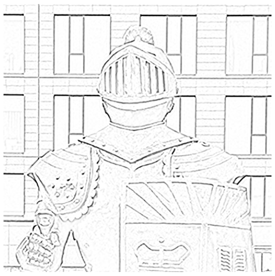
where $\nabla^2 f = f_{xx} + f_{yy}$ is the *Laplace operator* and the dots contain terms of order at least 4. The Laplace operator is invariant by rotations $\nabla^2(f \circ R) = (\nabla^2 f) \circ R$, it applies constant functions into zero and it is very sensitive to changes in the regularity because it involves derivatives. Then it qualifies as an *edge detector*. In plain terms, L subtract 4 times the values of a pixel from those of the 4 surrounding pixels up, down, left and right, the result is quite small if there is not an edge there because these pixels carry similar gray colors.

Let us see it in action on an originally 512×512 image. Here it is the application of $L \otimes B_d$ for $d = 1, 2, 3$ clamped with (2.104). To ease the visualization it is printed the negative of the result, $1 - F$ instead of F, in this way the background becomes white.

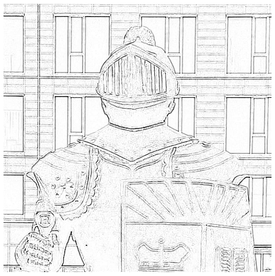
Original

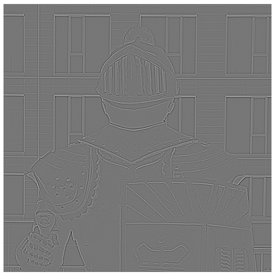

Filter L

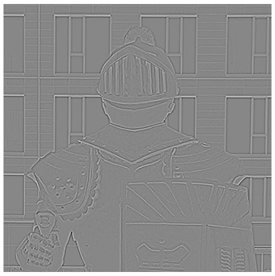

Filter $L \otimes B_2$

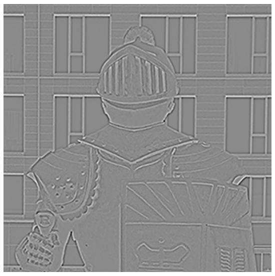

Filter $L \otimes B_3$

It is a simple exercise to check that if we set the zero elements of L in (2.103) to -1 and the central value to 8, the new filter L' corresponds to $-3h^2\nabla^2 f$. This exaggerates the effect of L as shown in the first image. The rest of the images are as before but considering normalization instead of clamping. The background corresponds originally to 0 (they are positive images) but after normalization it becomes close to the mean gray tone 1/2.


Filter L'


L normalized


$L \otimes B_2$ normalized


$L \otimes B_3$ normalized

The effect of S is not difficult to guess even without recalling its name. Note that $S - L$ has the same effect as B_1 that acts as the identity. Hence S gives the images plus the edges, it outlines the picture sharpening the global aspect in some way. On the other hand, it is much more challenging to guess the effect of E on an image before seeing any example. Note that if we omit the central 1, which we can consider as the identity B_1, we have antisymmetry along the secondary diagonal. The southwest neighboring pixels tend to raise the resulting value and the northeast neighboring pixels tend to lower it. In the presence of an edge, an abrupt change, both contributions are not compensated and create a *bump map* recalling to sheet metal embossing.

The filters S and E give the following results in the previous example. They are shown clamped with (2.104).

Filter S

Filter $S \otimes B_2$

Filter E

Filter $E \otimes B_2$

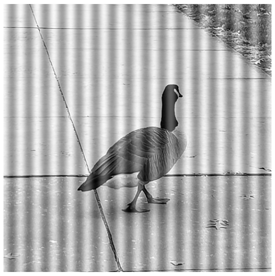
So far we have seen linear filters for images on the space domain. We can interpret them on the frequency domain via (2.57) as a multiplication of the DFT by certain function but they would acquire useless cumbersome formulas lacking any motivation. On the other hand, there are examples of linear filters that become more natural on the Fourier transform side. We examine here two of them related to some practical problems.

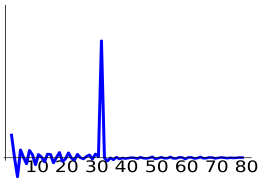
It is not uncommon in analog electronics to observe periodic noise, a contamination of the signal by the interference of signals coming from other devices, for instance a power supply. In frequency domain they appear as peaks and to cancel them it is enough to multiply by a function, the discrete Fourier transform of the filter, that is zero at the peaks and one elsewhere. To keep real numbers and to respect our mirror continuation of the images, we consider the DCT instead of the DFT. By the way, as we have a function of two variables we have to apply the definition of Proposition 2.2.2 on each variable (recall the JPEG format). For an image $F(m,n)$ its DCT will usually have a big peak at the origin because $\widehat{F}^c(0,0) = \sum_m \sum_n F(m,n)$ while we expect a lot of cancellation at the rest of the values. If we observe a set of big peaks at other points $\mathcal{P} = \{(p_j, q_j)\}$ the process described before to clean the image is

$$(2.107) \qquad F \longmapsto \text{inverse DCT of } \begin{cases} \widehat{F}^c(m,n) & \text{if } (m,n) \notin \mathcal{P}, \\ 0 & \text{if } (m,n) \notin \mathcal{P}. \end{cases}$$

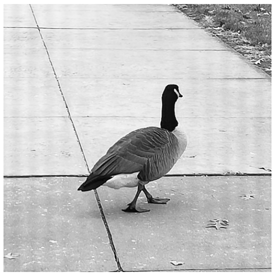
Here we have an example again of original size 512×512:

Original


Periodic noise



DCT at $(m,0)$


Filtered image

If we plot the DCT of the image affected by the periodic noise we will obtain a surface with a big peak at $(0,0)$ of size around $3.2\cdot 10^6$ and another at $(32,0)$ of size $7.4\cdot 10^5$. The relevant section of this surface is depicted in the third figure. If we put this value to zero and take the inverse DCT we get the last image which is quite clean. In this purely academic example (see [Rus07, Ch.6] for more realistic images) the periodic noise was made up by hand adding the pure wave $0.25\cos(\pi m/16)$. The final result is still not perfect because with the zero that we have introduced we have lost completely $\widehat{F}^c(32,0)$, represented by the faint shadow of the noise in the last image.

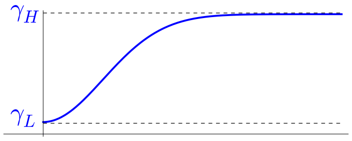
The second example we are going to present of filters on the frequency domain is more realistic (a little old fashioned though). Some photos, especially under artificial sources of light, show a displeasing effect of gradient of shadowed zones. We intend here to face this problem making the illumination more homogeneous preserving the tones and the aspect of the photo (for more extreme procedures see the next subsection). The *illumination-reflectance model* [GW08, §2.3.4] claims that in this situation an image (2.98) can be written as $F = IR$ where I gives the illumination and contains low frequencies and R gives the reflectance and contains mainly high frequencies. To clean the photo we want to suppress the variation of I causing the shadowed zones setting it to a constant. The natural procedure is to take logarithms and apply a linear *high-pass filter*, the opposite of a low-pass filter: it lets only the high frequencies. This is called *homomorphic filtering*. The funny name is explained in one of the original papers [OSS68] and essentially comes from the group homomorphism $(\mathbb{R}^+,\cdot)\longrightarrow(\mathbb{R},+)$ given by the logarithm. So, in the idealized model to solve the problem we have to use

$$\text{(2.108)}\qquad \log F(m,n)\longmapsto \text{inverse DFT of }\begin{cases}0 & \text{if } m^2+n^2 \text{ is small,}\\ \text{DFT}(\log F)(m,n) & \text{otherwise.}\end{cases}$$

This operator produces the enhanced $\log F$, to get the image we have to apply the exponential. Note that rigorously speaking it is not a linear filter on F but on its logarithm.

In practice an ideal high-pass filter taking values 0 and 1 with a sharp jump is not a good idea, as we saw in the previous subsection for low-pass filters (see also [Bov09, §10.3.2]). In our situation is even a worse idea because the model is an approximation. Instead of a step function, it is usually considered

$$\text{(2.109)}\quad G(m,n) = (\gamma_H-\gamma_L)\Big(1-\exp\big(-\frac{m^2+n^2}{2\sigma^2}\big)\Big)+\gamma_L$$

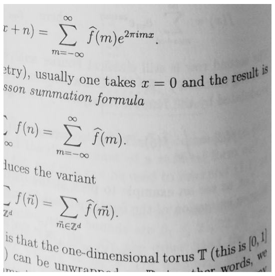


for some parameters γ_L, γ_H and σ. The width of the step, in some sense the width of the transition band, is proportional to σ and provides the smoothing. On the other hand, γ_L controls the attenuation of the illumination, in principle it is a small value, and γ_H ideally should be 1 but one can take greater values to amplify the reflectance. All of these parameters admit a broad tuning depending on the photo.

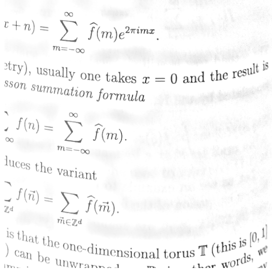
Another practical problem is that F can take the value 0 ruining the definition of $\log F$. A solution is to change $\log F$ by $\log(F+\epsilon)$ with ϵ a small positive constant. We use here, as before, the DCT instead the DFT. Summing up, our application of the homomorphic filtering is

$$F \longmapsto \exp\big(\text{inverse DCT of } G \cdot \text{DCT}\big(\log(F+\epsilon)\big)\big) - \epsilon. \tag{2.110}$$

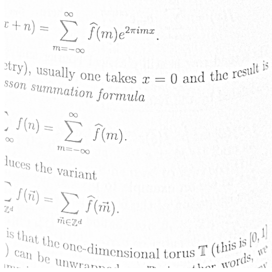
We have got an example of size 512×512 taking a photo of a combed sheet of paper. The curvature induces a gradient of light with a shadowed part to the right as shown in the first image.



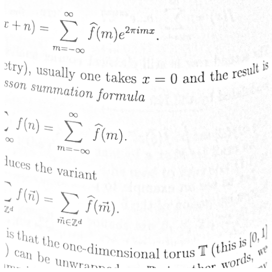
Original



Ideal r=20



$\sigma = 60$, $\gamma_{\mathsf{H}} = 1$



$\sigma = 30$, $\gamma_{\mathsf{H}} = 1.1$

For the rest of the images the small constant ϵ is $1/256$. In the second photo we see the effect of an ideal high-pass filter canceling all the frequencies in a circle of radius 20. In the last two images it is used (2.109) with $\gamma_L = 0$. For the third $\gamma_H = 1$ gives a pure Gaussian high-pass filter and with some experimentation $\sigma = 60$ seems to give one of the best results although it is not very sensitive to similar values of σ. On the other hand there is a noticeable sensitivity to γ_H. In the last image with $\gamma_H = 1.1$ and $\sigma = 30$, when observed to its original size, it seems that there is a tiny improvement in the contrast.

It is fair to say that the images were clipped to $[0,1]$ and for a photo of a text in which we expect only black and white, there is a little bit of cheating doing this. See in [PP10, §4.3] the effect in more standard photos and [GW08, §4.9.6] for some medical images that are somewhat as special as texts.

Suggested Readings. Linear filtering is a small part of image processing and it appears in general books together with more advanced techniques. One of the most popular books on digital image processing is [GW08] with quite clear explanations. Another noteworthy addition to the bibliography is [PP10] containing numerous simple examples. The readers of this kind of books are not usually mathematicians and then for somebody used to advanced mathematics they can seem too verbose or too non rigorous. If it is not a problem for you, consider also [Bov09] and [Rus07]. The latter, as its title advertises, tends to be a reference book. It contains a large number of very illustrative full color images.

2.2.5 More on image processing

There is a big world in image processing beyond the simple linear filters that we have seen before. The purpose of this subsection is to give some isolated examples just for illustration, without trying to do any general theory. We still consider only grayscale images represented mathematically by (2.98).

We focus mainly on *nonlinear filters* but we will start taking into consideration some actions on the histogram of the image [GW08, §3.3]. This is the histogram, in the sense of descriptive statistics, of the gray levels of the image (as a matter of fact due to the quantization the maximum number of bins is 256). If the histogram is supported on a small interval $[a, b] \subset [0, 1]$ then we are wasting levels of gray and it is convenient to do a *normalization* of the levels by the bijective linear function (2.105). With this simple trick the quality of some images improves a lot. In practice many electronic devices already include a normalization step and to get nontrivial results we have to clip firstly the values to a suitable $[a, b]$ containing the bulk of the histogram. This is called *histogram stretching* by obvious reasons.

The *histogram equalization* exaggerates this latter idea. Ideally the histogram represents an approximation of the density function f of a random variable and the histogram equalization consists in finding a change of variables, acting on the gray levels, such that the random variable becomes a uniform distribution $U(0, 1)$. Mathematically, if the domain of f is $[a, b]$ (the range of gray levels appearing in the image) the map $\phi : [a, b] \longrightarrow [0, 1]$ that performs the change of the gray levels is

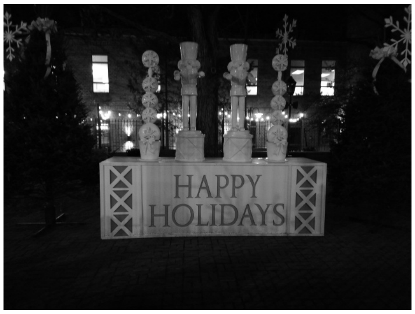
$$\phi(x) = \int_0^{\phi(x)} 1\, dt = \int_a^x f(t)\, dt. \tag{2.111}$$

In reality we have a discrete distribution of levels and the previous mathematical model for equalization must be replaced by a kind of rectangle rule for the last integral:

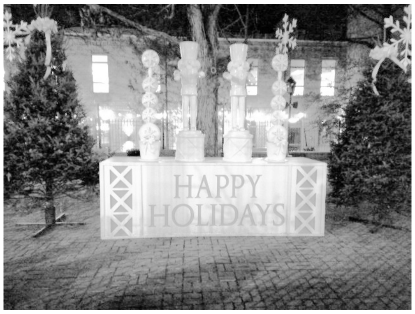
$$\phi(x) = \frac{\#\{\text{pixels with level} < x\}}{\#\{\text{pixels}\}}. \tag{2.112}$$

Note also that in practice x and $\phi(x)$ are limited to a finite number of values by the quantization. This is a big restriction and causes that the histogram of an equalized image largely differs from being flat, it is just more evenly distributed that the original but does not correspond to a uniform distribution (see [PP10, §4.4] for perfectly flat histograms).

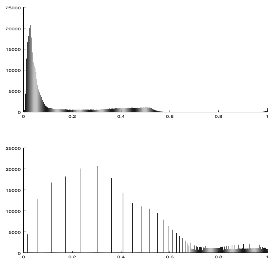
As an example consider the left photo of size 640 × 480 that was taken by night. By the poor illumination the histogram (upper right) is clearly biased to 0 but the lights in the background cause that there is a tiny amount of levels close to 1, then there is no room for normalization. After equalization (central photo) a couple of Christmas trees appears in the side dark zones. Believe or not, this was a surprise even for the photographer!



Original



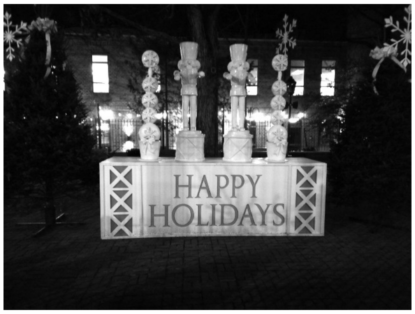
Equalized



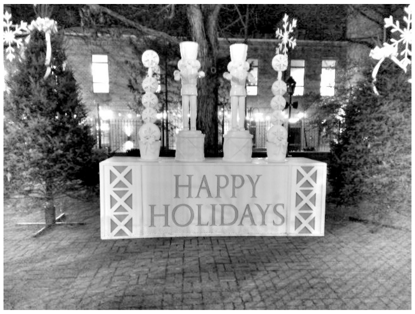
Histograms

Note that the second histogram, corresponding to the equalized image, is flatter but by no means mimics a uniform distribution. The hair comb aspect, plenty of gaps, is caused by the quantization of the original levels.

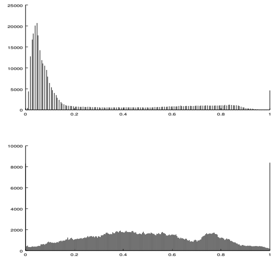
The histogram of the original photo is included in $[a, b] = [0, 0.6]$ except for an insignificant amount of values but histogram stretching does not give a good result to reveal the darker details, as shown below.



Histogram stretching



Log model



Histograms

Normalization and histogram equalization cost just clicking an option in a raster graphic editor (like GIMP, histogram stretching is also possible) and with them some photos or scanned documents experiment a noticeable enhancement. Normalization gives the right contrast while histogram equalization makes all the tones equally important which gives an artificial aspect to artistic photos but it is surprisingly efficient to reveal hidden details in low contrast areas, by this reason it is useful in medical imaging, for instance processing X-ray images. It can also renew your old photocopies.

More sophisticated methods automatically detect the zones with low contrast and apply the stretching only to a range of values depending on the contrast [Rus07, Ch.5] [PP10, §4.3]. A logarithmic model based in Weber's law is to modify the value $f(i, j)$ of the pixel (i, j) of the image according to

$$f(i,j) \longmapsto (T \circ c)\Big(\log \frac{f(i,j)+\epsilon}{m(i,j)}\Big) \tag{2.113}$$

where ϵ is a small constant to escape from $\log 0$, m is the result of averaging f applying a Gaussian blur of certain σ, T is like in (2.105) and c is like (2.104) replacing 0 and 1 by a and b. The second image above was obtained taking $\sigma = 60$, $\epsilon = 1/256$, $a = -1.8$ and $b = 1$. Surely playing a little with the values one can improve the result.

We have already seen that the Laplacian filter detects edges. From the mathematical point of view, the edges are more related to abrupt changes in the first derivatives than to second derivatives and it motivates to consider the gradient. In polar coordinates it is

$$r = \sqrt{f_x^2 + f_y^2}, \qquad \theta = \arctan \frac{f_y}{f_x}. \tag{2.114}$$

Here r is isotropic, meaning that $r(p)$ is invariant when f is replaced by $f \circ R$ with R a rotation fixing p. This is an interesting property and suggests to use r as an edge detector.

Note also the Taylor expansion

$$f(h,h)-f(-h,h)+2f(h,0)-2f(-h,0)+f(h,-h)-f(-h,-h)=8hf_x(0,0)+\dots \tag{2.115}$$

where the dots contain terms are least cubic in h. With it and a similar expansion exchanging the role of x and y we have that f_x and f_y are well represented by the application of the "horizontal" and "vertical" linear filters H and V with matrices

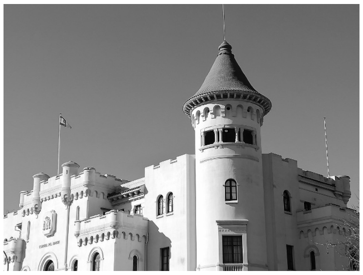
$$H=\frac{1}{8}\begin{pmatrix}-1&0&1\\-2&0&2\\-1&0&1\end{pmatrix}\qquad\text{and}\qquad V=H^t=\frac{1}{8}\begin{pmatrix}-1&-2&-1\\0&0&0\\1&2&1\end{pmatrix} \tag{2.116}$$

If F is the image, the nonlinear filter

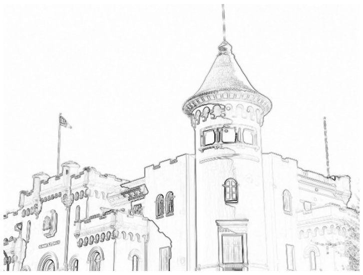
$$F\longrightarrow\sqrt{(H*F)^2+(V*F)^2} \tag{2.117}$$

is called the *Sobel filter* and it takes large values in the potential edge points in which the variation of the values of nearby pixels is large. Recall that if with your scale pixels are too small to see any result one can replace H and V by $H\otimes B_d$ and $V\otimes B_d$, the Kronecker products.

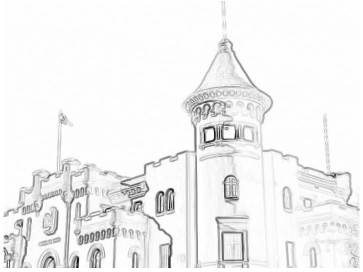
To do the visualization more appealing, it is convenient to represent the negative of the result of applying the filter scaled to $[0,1]$, in this way the edges appears in black and the smooth zones in white. Here you can see a couple of examples with an original size 640×480:



Original

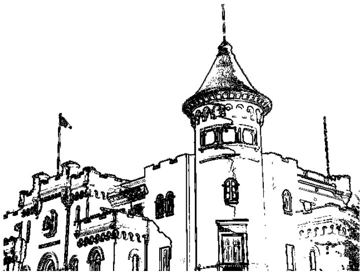


Sobel d = 1

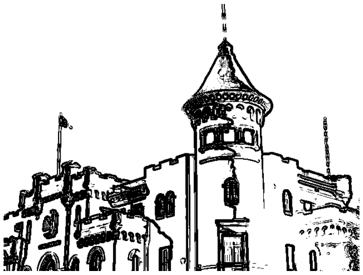


Sobel d = 2

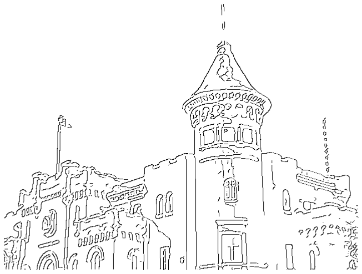
The results remember to something made with a pencil or a charcoal pencil (in fact it is a cheap way of simulate a handmade drawing from a photo). To get more realistic sharp edges, after the normalization to $[0,1]$ of the negative of the result of the filter, one can establish a threshold δ such that every level below δ is considered black and any level above is considered white. The first two pictures were obtained taking $\delta=0.92$ in the previous example.



Sobel d = 1, $\delta = 0.92$



Sobel d = 2, $\delta = 0.92$

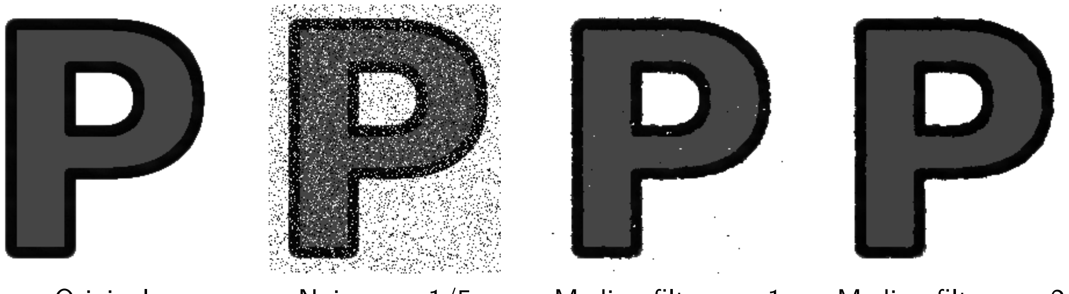


Canny

Edge detection is a vast topic involving many techniques (see [GW08, §10.2] for an overview and [NA08, §4.1] for a short list of techniques and aims). Arguably the most employed is the *Canny edge detector*. Its effect, as implemented in `octave`, is shown in the last picture. A precise description would take too long. The main lines are that the Sobel filter is applied to a Gaussian blurred version of the image to balance the possible noise, the angle in (2.114) is also estimated and as the gradient is orthogonal to the level sets, it is possible to obtain some idea about the direction of the edge. With this information replicas of the edge coming for close points with large gradient are suppressed and finally it is established an algorithm to join them into lines (in [NA08, §4.2.5] there are pictures showing the different steps). For the theory under this algorithm and on edge detection theory in general, see the original paper [Can86].

Now we are going to consider noise reduction in image processing. There are several kinds of *noise* according to their statistical properties [PP10, §4.2]. We consider first the situation in which due to a defective channel, from time to time with probability p we get a completely erroneous value well outside the interval $[0, 1]$. After clamping each of these outliers will become a white pixel or a black pixel, by this reason this kind of noise is called *salt-and-pepper noise*. A first idea would be to average with B_d but after a basic course on statistics we all know that the mean does not behave very well in the presence of outliers. Think that in a course all the students except one can fail an exam and still the mean grade can be enough to pass the exam. People believing that the statistics giving the average income are untrustworthy should have this in mind. A more robust sample statistic is the median. If the median income is x dollars we will be sure than half of the people receive less than this amount. It reflects a kind of democracy of the data, if we obtain in an experiment 0.11, 0.12, 0.11, 0.09 and 103.04, we should suspect that the result is close to 0.11, irrespectively of the mean value, and that the last was an experimental error or an error in communication.

In our case, we fight the salt-and-pepper noise with the *median filter* that consists in taking the median of the values reached on a square $(2r + 1) \times (2r + 1)$, $r \in \mathbb{Z}^+$, around the selected pixel. For $p = 1/5$ in a 200×232 image we get:

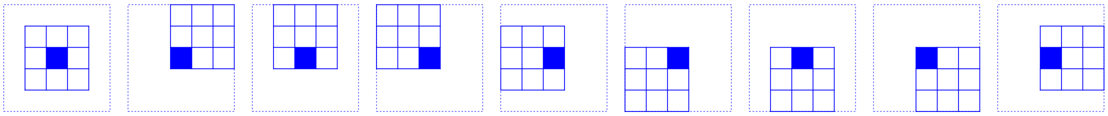


We see that in the third image they only remain some scattered points taking the median in squares 3×3. When we consider squares 5×5 (last image) the result is almost perfect except for a small lack of resolution on the edges. This is quite impressive taking into account that we had lost completely one fifth of the pixels.

The median filter is part of what is known as *rank order filters*. In these filters we change the value of a pixel taking into account the ordering of the values in a neighborhood containing it. So, the median takes the value in the middle when they are ordered and taking the smallest value could be useful to grow a dark background.

Another instance of noise consists of small random independent alterations of each value. Due to the central limit theorem the typical situation is that the errors added to each value follow a normal distribution $N(0,\sigma)$. This is called *Gaussian noise*. In this case the median filter is not very useful: the democracy of the values is not the best idea if all are wrong. In the linear setting a possibility is to use B_d or another kind of blur filter because on average the error is zero. We have to find a balance between how much noise we want to avoid and the amount of blur we admit.

The result would be more appealing if we could keep the hard edges, preserving in some way the shape, blurring separately in the regions bounded by these edges. The *Kuwahara filter* is a nonlinear filter that approaches to this selective blur. Given a radius $r\in\mathbb{Z}^+$, for each pixel we consider all the $(2r+1)\times(2r+1)$ squares of pixels containing it.



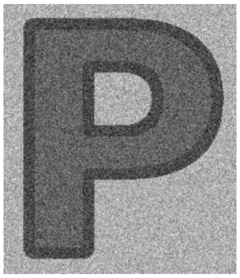
Case r = 1: The nine 3 × 3 squares containing a pixel

Among them we consider the square $\{s(m,n)\}_{m,n=0}^{2r}$ having the smallest variance

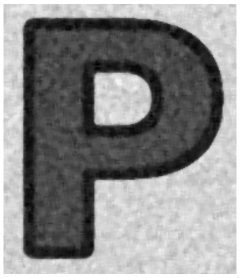
$$\frac{1}{(2r+1)^2}\sum_{m,n=0}^{2r}\big(s(m,n)-\bar{s}\big)^2 \qquad \text{with} \quad \bar{s}=\frac{1}{(2r+1)^2}\sum_{m,n=0}^{2r}s(m,n) \tag{2.118}$$

and we assign to the current pixel the value $\bar{s}$. In this way the pixels tend to be blurred with alike pixels.

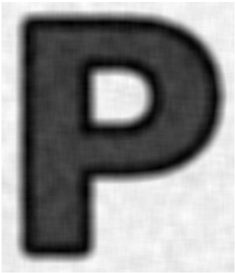
With the previous original image affected with Gaussian noise $\sigma=1/5$, the Kuwahara filter for $r=5$ gives a neater result than the corresponding flat blur with B_{2r+1}.



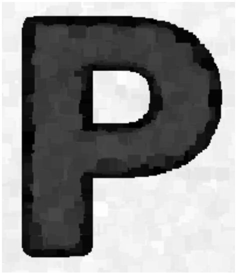
Noisy $\sigma = 1/5$



Median r $=5$



B_d with d $=11$



Kuwahara r $=5$

The hazy aspect of the noisy image reveals that we are displaying it normalized because clamping it to $[0,1]$ would not give a clear idea about the noise.

Kuwahara filter produces a nice artistic effect on photos, it gives them a water painting like aspect. In some way r is the radius of the brushstroke and when it is higher it looks more like a sketch.

This is the result on one of the images that we have managed before.

r = 2

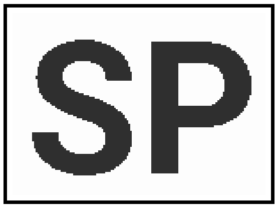
r = 4

r = 5

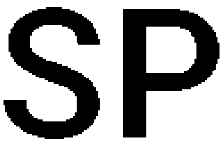
We finish with some comments about a family of filters called *morphological filters* that in principle act on *binary images*, those containing only black and white so Im F in (2.98) is $\{0,1\}$. There are two primary operations called *erosion* and *dilation* that are the basis of these filters.

A binary image can be considered as the set formed by the black pixels[10]. With this interpretation, the erosion of a binary image is the set formed by all pixels such that a 3×3 square centered in them is completely contained in the image. On the other hand, the dilation is formed by the pixels such that the intersection of the 3×3 square with the image is nonempty. A more general definition considers bigger squares or other regions.

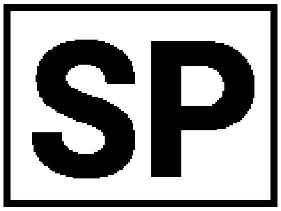
Here we have a 160×120 example. The 2 pixels thick frame disappears after erosion.



Original

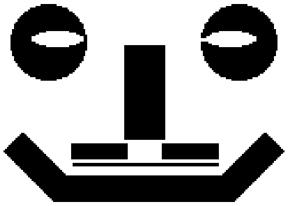


Erosion

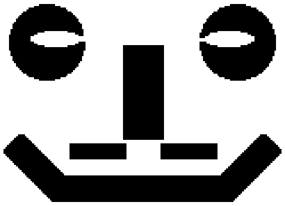


Dilation

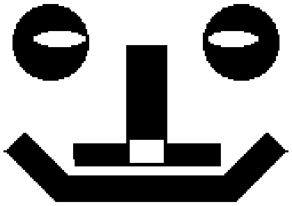
These operations do not commute. The erosion followed by dilation is called *opening* and the dilation followed by erosion is called *closing*. As the names suggest, opening and closing tend to open and close the gaps. This is another 160×120 example:



Original



Opening



Closing

[10] The tradition is to take the white pixels but to ease the visualization I dare to contradict it.

Note the thin horizontal line is omitted by opening and it is glued to the upper rectangles by closing. Note also that the "eyes" are different in the three images.

A part of the theory of morphological filters extends to nonbinary images [Bov09, §13.2.2]. For instance, erosion and dilation can be generalized as the rank order filter corresponding to the maximum and minimum [Soi03, §3.2, 3.3].

Suggested Readings. The reader is referred to the general aforementioned books [GW08], [PP10], [Rus07] and [Bov09] where the topics of this subsection are treated with more detail. In a big part of [NA08] it is also covered general digital image processing with a certain mathematical twist. There is a chapter in [GW08] and [Rus07] devoted to morphological filters and their applications and even the whole book [Soi03]. Its last chapter contains references to real life applications.

2.3 Problem set and challenges

Note: These exercises were created for the assessment during the master course at UAM *Wavelets and signal processing* 2017/2018.

Problem set 2

Introduction to digital signals

Problems

1) Let f be a smooth band limited function with $\widehat{f}(\xi) = 0$ for $|\xi| \geq B$ and w a bounded integrable function such that $w(\xi) = 1$ when $\widehat{f}(\xi) \neq 0$ and $w(\xi) = 0$ for $|\xi| \geq B$. Prove

$$f(x) = \frac{1}{2B} \sum_{n=-\infty}^{\infty} f\big(\frac{n}{2B}\big)\widehat{w}\big(\frac{n}{2B} - x\big).$$

2) A signal takes values in $J = [-2, 2]$ with density of probability proportional to the distance to the closest extreme of J and we want to quantize the signal to get only two possible values. Design a quantizer minimizing the mean square error.

3) For the nodes $x_j = -1 + j/2$, $0 \leq j \leq 3$ and the images $y_0 = 5$, $y_1 = y_3 = 2$, $y_2 = 0$ consider the natural cubic spline s with $s(x_j) = y_j$. Compute explicitly the cubic polynomials $s\big|_{[x_j,x_{j+1}]}$ for $j = 0, 1, 2$.

4) Prove that the entries of the (dyadic) Bayer matrices M_k, as defined in the notes, are a rearrangement of $\{j2^{-2k}\}_{j=0}^{2^{2k}-1}$.

5) Prove the formula employed in the experimental challenge *Moiré, qu'est-ce que c'est?*

$$\sum_{n\in\mathbb{Z}} e^{-\pi(n-\nu t)^2} = \sum_{n\in\mathbb{Z}} e^{-\pi n^2} e(\nu n t).$$

Notes and hints

1) Note that Shannon sampling theorem can be got as a consequence. It corresponds to choose as w the characteristic function of the interval $[-B, B]$. This exercise allows to choose different windows to recover the signal.

2) "Design" means "find a formula". Using the symmetry you will simplify a lot. By sheer curiosity I have tried the calculations not assuming any symmetry and I can assure they are a nightmare.

3) Use whatever you like to help you with the computations (fingers, slide rule, calculator, supercomputer. . .) but saying “I have a wise computer package that putting the nodes and values give me the spline” is not a valid answer. In other words, you have to display the sequence of calculations you (or your calculator or your computer) are doing. The final result involves small numbers and it is possible to proceed completely by hand without any assistance. If you want to think about it, there is a subtle symmetry trick to simplify the calculations with bare hands but I did not assume it in my last claim.

4) Yes, this is simple cheap combinatorics (no offense intended if there is a combinatorist in the room). Consider it an Easter gift.

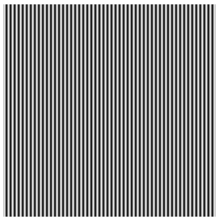
5) The most expeditious way to proceed is to apply the Poisson summation formula. You can also try the direct Fourier expansion that is almost as quick and somewhat includes the proof of the Poisson summation formula.

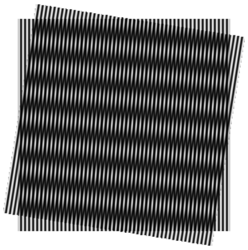
Experimental challenge: **Moiré, qu'est-ce que c'est?**

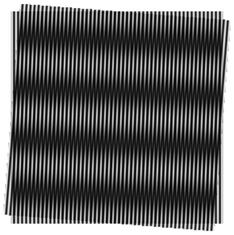
Introduction to digital signals

Experimental part

Print very close black dots, straight lines or curves on a couple of transparent slides and play to overlap them and observe some funny patterns, it is the so called, *moiré effect*. You can also use gray tones. To my taste one of the best results is achieved with thin circular sectors (see the complementary material web page) but it appears even in simple situations.


Single slide


angle $\pi/20$


angle $\pi/40$

Try to reproduce at least the experiment above with vertical lines. For large angles nothing peculiar happens, but for small angles the pattern appears.

Mathematical part

The grid depicted above actually corresponds to the gray tone $C(t) = \big(F(t) - m\big)/(M - m)$ with $F(t) = \sum_{n\in\mathbb{Z}} e^{-\pi(n-\nu t)^2}$ and m and M are the minimum and the maximum of F to normalize it between 0 (black) and 1 (transparent). Explain why the natural model for any color function C depending on one variable is that superposition with angle α corresponds to the color function $G(x, y) = C(x)C(x\cos\alpha + y\sin\alpha)$.

The (2D) Fourier transform formally detect periodicity through Dirac deltas. On the other hand, at a certain distance we cannot distinguish frequencies beyond a certain threshold V. So if you compute $\widehat{G}$ and look for values $\|\vec{\xi}\| \le V$ such that $\widehat{G}(\vec{\xi})$ has the highest peaks (coefficients of the Dirac deltas) then $\vec{\xi}$ indicates the direction in which the pattern repeats and $\|\vec{\xi}\|$ its frequency.

Explain the figures using $F(t) = \sum_{n\in\mathbb{Z}} e^{-\pi n^2} e(\nu n t)$ and this model with V much smaller than ν. Prove that using grids of different frequencies ν_1 and ν_2 the frequency of the pattern is $\sqrt{\nu_1^2 + \nu_2^2 - 2\nu_1\nu_2\cos\alpha}$. It is interesting because it allows to measure tiny distances observing macroscopic effects. You can find much simpler geometric explanations for this example of vertical lines but Fourier analysis applies to more complicate cases.

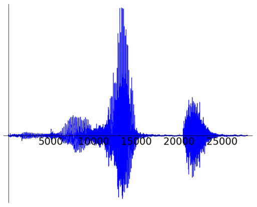
Experimental challenge: **Echo, echo, echo**

Introduction to digital signals

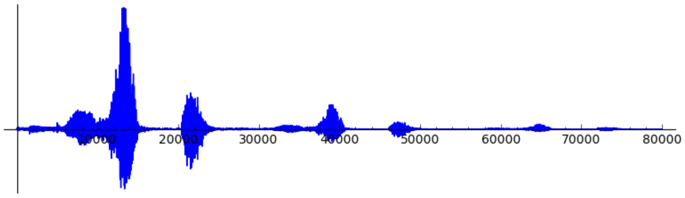
Experimental part

This is a simple experiment to do with Octave or Matlab. You only have to look up in the documentation the commands to read digital audio (`audioread`, `wavread` in older versions) and to play it (`sound` or `soundsc`) or to store it (`audiowrite`, `wavwrite` in older versions).

When sound meets a distant obstacle it is reflected and come back to the source. A simple model for echo is to sum to a signal a displaced version of itself attenuated by a factor A. We assume that there is another obstacle (a wall) just behind the emitter and hence we also add the n times displaced signal with a factor A^n until A^n is negligible. Write a program simulating echo taking as input a sample of voice and the distance d to the object (note that the speed of sound is $343m/s$ and the distance d is covered twice). Try to adjust the attenuation to produce a realistic effect. For d like 100 you should get something like the echo in the mountains and for d like few meters you should get an effect of "empty room".



Voice signal



Distant echo mildly attenuated ($d = 100$, $A = 1/5$)

Technically, in acoustics the name *echo* refers to only one reflection ($n = 1$), instead of many as we consider here. With this jargon, our case is a toy model of *reverberation*. In the most realistic simulations it takes into account the phase change of the reflected signal, the different responses of different frequencies and even the situation of multiple obstacles.

Mathematical part

Find a formula relating the Fourier transform of a signal (as smooth as you wish) and the Fourier transform of the echo. Taking the inverse transform, write the integral operator mapping the signal into its echo. If you have software to compute integrals numerically, an interesting exercise is to check numerically that it works as expected for instance for Gaussian signals.

Experimental challenge: **Unhuman voice**

Introduction to digital signals

Experimental part

The proposal is to reproduce digitally three effects that come from the analog times. In every case use Octave or Matlab to load an audio file as a sample $\{y_k\}_{k=1}^{L}$ and to store or play the result.

The first effect consists in multiplying the sample by a pure tone (sine or cosine) of certain frequency ν. Adjusting ν one gets a fairly good robot voice effect (professional results are obtained with *vocoders*, relying heavily on Fourier analysis). It depends on your taste and your sample but I have got decent results with $\nu = 800\ Hz$ (real time frequency, not counting ν_s).

The second effect is called *distortion* and it appeared as a defect of the first electric guitars that was intentionally exaggerated later. By the low quality of the early amplifiers, loud sounds produced noise. To reproduce the effect one applies to the signal a function that is close to a multiple of the identity near the origin and introduces a severe clipping to large values. I have tried $f_1(x) = \frac{2}{\pi}\arctan(ax)$ and $f_2(x) = \operatorname{sgn}(x)\big(1 - e^{-a|x|}\big)$ with a large.

The third effect is the *vibrato*, well-known in music. As the name suggests, it introduces a kind of vibration in the sound. You can produce it, creating a new sample with $z_{m(k)} = y_k$ where $m(k) = k + H(k)$ with H a discrete harmonic oscillator. I have tried $H(t) = Q\big(W + W\cos(\omega t)\big)$ with Q the uniform quantizer. For $W = 10$ (under $\nu_s = 44100$) and ω corresponding to $\nu = 5000\ Hz$ the effect with music is not very bad. For voice I have found $W = 20$ more convenient. In both cases it would improve using interpolation of nearby values instead of quantization but I have no tried it.

Mathematical part

This challenge is mainly experimental, to play with the many variations and parameters to get good results so the mathematical part is rather meager.

In connection with the first effect, write the relation between the discrete Fourier transform of the signal and that of the robotized signal. For the second effect, be sure that you see that f_2 is close to a multiple of the identity near the origin. If you want to try the third effect a must is to find the relation between ν and ω.

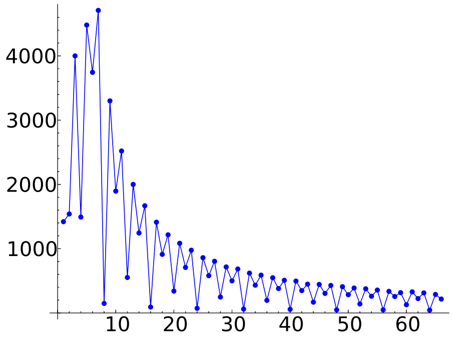
Experimental challenge: **Checkerboard oddity**

Introduction to digital signals

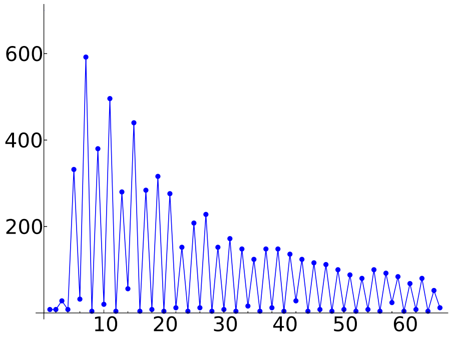
Experimental part

Create a very large image with a checkerboard pattern i.e., alternating black and white squares. If you want to save time and to have more extensive data I recommend to use a little Octave or Matlab code.

The experiment consists in exporting the image to JPEG format and plot the size of the file in kilobytes in terms of the side in pixels of each square, say N. With $1 \leq N \leq 66$ and an image 2520×2520 I got the first figure. Try your own experiments. The results and scales may vary with the software but the general aspect should be similar.



Main experiment



For the optional part

The second plot shows what I got applying to the resulting JPEG files a standard data compression program, in this case `gzip`. Usually this is not a good idea because JPEG already involves compression and we take the risk of getting a bigger file but in this case the ratio is impressive.

Mathematical part

Find formulas matching your data that allow to predict approximately the size of the JPEG file. Try to find a theoretical foundation to these formulas. This foundation cannot be perfect because we have not seen in detail the compression part of JPEG but you have to be able to explain the global aspect of the graph.

Optional: Although we have not studied how `gzip` works, does any idea come to your mind to explain the difference between both figures?

Chapter 3

Wavelets

3.1 Multiresolution analysis

3.1.1 Windowed transforms

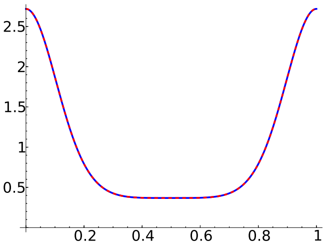
Although along this chapter the underlying Hilbert space will be $L^2(\mathbb{R})$, we start with a completely explicit example with Fourier series to illustrate the situation. We consider the 1-periodic functions

$$f_1(x) = e^{\cos(2\pi x)} \cos\big(\sin(2\pi x)\big) \qquad \text{and} \qquad f_2(x) = f_1(x) + \Big(\frac{\sin(50\pi x)}{50\cos(\pi x)}\Big)^2. \tag{3.1}$$

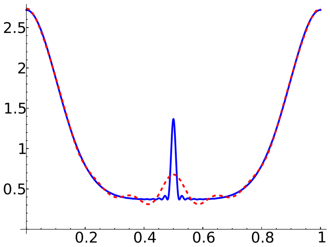
Both functions differ in a scaled and displaced version of Fejér kernel (1.51) and in this way f_2 in $[0,1]$ is like f_1 with a peak of height 1 and width approximately $1/25$ at $x = 1/2$. These $C^\infty(\mathbb{T})$ functions equal their Fourier expansion which admit the closed form

$$f_1(x) = \sum_{n=0}^{\infty} \frac{\cos(2\pi n x)}{n!} \qquad \text{and} \qquad f_2(x) = -\frac{1}{50} + \sum_{n=0}^{\infty} \Big(\frac{1}{n!} + \frac{(-1)^n}{25}\Big(1 - \frac{n}{50}\Big)_+\Big) \cos(2\pi n x), \tag{3.2}$$

where the index $+$ indicates the positive part. The following figures contain the plots of the functions f_1 and f_2 and the approximation (the dashed line) truncating these Fourier series up to $n = 8$.



f_1 and $S_8 f_1$



f_2 and $S_8 f_2$

The peak occurring in a short amount of time (or space, if you prefer) causes a small change in 50 Fourier coefficients and we have to enlarge a lot the range of frequencies to get a good approximation.

The underlying idea, as we saw, is the uncertainty principle. It is impossible to see fine details with a limited range of frequencies. How to fight against uncertainty principle? You cannot defeat theorems but one can play the usual game in mathematics: If you do not like the conclusions, circumvent them changing the hypotheses. Changing the usual complete orthonormal system $\{e(nx)\}_{n\in\mathbb{Z}}$ in $L^2(\mathbb{T})$ by another containing a multiple of $f_2 - f_1$, to represent the peak we would only need to spend a Fourier coefficient. Yes, it sounds as a bad joke but gives the idea of fighting peaks with peaks.

Let us move to $L^2(\mathbb{R})$ to address the problem in successive steps. Our intuition from Fourier analysis and (1.101) tell us that, due to the lack of compactness of $\mathbb{R}$, we deal here with different animals, Fourier integrals instead of Fourier series, but the epitome of wavelets that we will study in the next subsection, will recover the series for $L^2(\mathbb{R})$.

A first natural idea is to introduce *windows*. They are different from the discrete windows that we introduced in §2.2.3 to design linear filters but there is a common underlying idea. Have in mind a window as an approximation of a compactly supported function, something that lives mostly in an interval, the one we are looking at. For us a window is a real function $w : \mathbb{R} \longrightarrow \mathbb{R}$ with $\|w\|_2 = 1$ that we assume to be as regular as we wish, for instance rapidly decreasing, because this is a temporary approach. If we are interested in representing details of a certain width we should choose w having the most of its mass in an interval of the same size. If we "localize" the harmonics $e(\xi x)$ of Fourier analysis with $w(x)e(\xi x)$ we will be able to reproduce functions living in the approximate support of the window, whatever it means. To analyze any function we must move the window along $\mathbb{R}$. It suggests to introduce the *windowed Fourier transform*, also called *short-time Fourier transform* when w is compactly supported,

$$G_w f(\xi, b) = \int_{-\infty}^{\infty} f(x)\overline{w}_{\xi,b}(x)\, dx \qquad \text{with} \quad w_{\xi,b}(x) = w(x-b)e(\xi x). \tag{3.3}$$

There exists a fair enough inversion formula and also a Parseval identity. We state them here under overkilling regularity conditions although there is an L^2 version [Bré02, Th.D.1.2].

Proposition 3.1.1. *Let w be as before. For any rapidly decreasing f we have*

$$f(x) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G_w f(\xi, b) w_{\xi,b}(x)\, d\xi db \tag{3.4}$$

and

$$\|f\|_2^2 = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \big|G_w f(\xi, b)\big|^2\, d\xi db. \tag{3.5}$$

Proof. By Parseval identity (1.74) and $\widehat{w}_{\xi,b}(t) = e\big((\xi - t)b\big)\widehat{w}(t-\xi)$, we have

$$G_w f(\xi, b) = \int_{-\infty}^{\infty} \widehat{f}(t)e(tb)\widehat{w}(\xi - t)e(-\xi b)\, dt. \tag{3.6}$$

Note that the conjugate of $\widehat{w}(u)$ is $\widehat{w}(-u)$ because w is real. Changing the order of integration and using the inversion formula for the Fourier transform,

$$\int_{-\infty}^{\infty} G_w f(\xi, b)e(\xi x)\, d\xi = \int_{-\infty}^{\infty} \widehat{f}(t)e(tb)e\big(t(x-b)\big)w(x-b)\, dt = w(x-b)f(x). \tag{3.7}$$

Then the right hand side of the first formula of the statement is $f(x)\|w\|_2^2 = f(x)$.

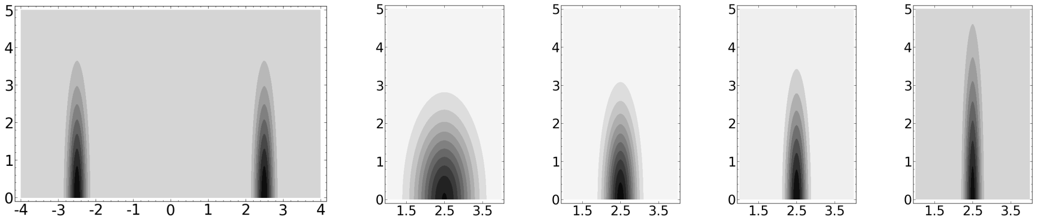
On the other hand, applying Parseval identity (1.74) to (3.7),

$$\int_{-\infty}^{\infty} \left|G_w f(\xi,b)\right|^2 d\xi = \int_{-\infty}^{\infty} \left|w(x-b)f(x)\right|^2 dx \tag{3.8}$$

and integrating on b, this is $\|w\|_2^2\|f\|_2^2 = \|f\|_2^2$. □

The definition (3.3) and the equality (3.6) used in the proof indicate that w cuts the function and $\widehat{w}$ cuts its Fourier transform. According to Theorem 1.2.9, in a certain vague sense the "best" choice is $w(x) = \pi^{-1/4}\lambda^{1/2}e^{-\lambda^2x^2/2}$ with λ^{-1} somewhat the size of the details we want to observe. In this case $G_w f(\xi,b)$ is called the *Gabor transform*.

As an illustration of how the windowed Fourier transform works, look the following contour plots (darker means larger values) of the Gabor transform $|G_w f(\xi,b)|$ where f is the function $e^{-32(x+5/2)^2} + e^{-32(x-5/2)^2}$ representing two peaks. The horizontal axis is b and the vertical axis is ξ. The complete plot is symmetric with respect to both axes.



$\lambda = 8$ $\lambda = 2$ $\lambda = 4$ $\lambda = 7$ $\lambda = 12$

As expected, the most of the mass is concentrated around $b = \pm 5/2$. The width of the peaks and of w coincide for $\lambda = 8$. For higher λ the latter is smaller and we have to pay with extra higher frequencies (if you use a too short measuring-tape you will have to use it many times). On the other hand, for λ much smaller the situation gets closer to usual Fourier analysis and for λ very close to zero we would obtain something approaching to a horizontal band i.e., not depending on b. Roughly speaking, the blobs have less area for λ close to 8, meaning that we have to consider "less values" of b and ξ to get a good approximation of the function.

The obvious shortcoming of the windowed Fourier transform is that it imposes a fixed size for the details we can analyze efficiently. The success of wavelets in practice relies on the capability to manage infinitely many windows at the same time that operate at different scales. A *wavelet* is essentially a fixed profile to be translated and scaled to make the windows.

There are three important avatars of wavelets. The first one is related to the windowed Fourier transform and we will define it right now. The second, mathematically more challenging, is related to the construction of orthonormal systems spanning $L^2(\mathbb{R})$. The third one is a discrete version, very simple from the mathematical point of view but the most useful in practice.

Following [Pin02] we call *continuum wavelet* to any $\psi \in L^2(\mathbb{R}) - \{0\}$ satisfying

$$\int_{-\infty}^{\infty} |\xi|^{-1} |\widehat{\psi}(\xi)|^2 \, d\xi < \infty. \tag{3.9}$$

We say that ψ is *normalized* if this integral equals 1. We can always normalize a continuum wavelet multiplying it by a positive real constant.

Given a certain wavelet ψ we define the *wavelet transform* associated to it as the operator that applies $f \in L^2(\mathbb{R})$ into

$$W_\psi f(a,b) = \int_{-\infty}^{\infty} f(x)\overline{\psi}_{a,b}(x)\, dx \qquad \text{where} \quad \psi_{a,b}(x) = \frac{1}{\sqrt{|a|}}\psi\Big(\frac{x-b}{a}\Big), \; a,b \in \mathbb{R}, \quad a \neq 0. \tag{3.10}$$

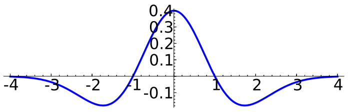
Here $\psi_{a,b}$ plays the role of a variable window where b gives the position and a the scale. Note that $\|\psi_{a,b}\|_2 = \|\psi\|_2$ then Cauchy-Schwarz inequality assures that for each $f \in L^2(\mathbb{R})$ its wavelet transform is bounded.

We expect some oscillation in ψ because we are mimicking $w_{\xi,b}$ rather than w. The condition (3.9) requires, for $\widehat{\psi}$ continuous, $\widehat{\psi}(0) = \int_{-\infty}^{\infty} \psi = 0$ and then involves a minimal oscillation. In general, the vanishing of moments $\int_{-\infty}^{\infty} x^j \psi(x)\, dx$ until certain n implies a better behavior of the wavelet transform. We do not expand this idea here. It will reappear later.

Anything with zero average, a minimal regularity and decay qualifies to be a continuum wavelet. Let us review three celebrated examples.

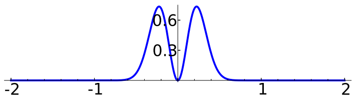
A normalized continuum wavelet is the so-called *Mexican hat wavelet*, with a self-explanatory name,

$$\psi(x) = \frac{1}{\sqrt{2\pi}}(1-x^2)e^{-x^2/2} \tag{3.11}$$



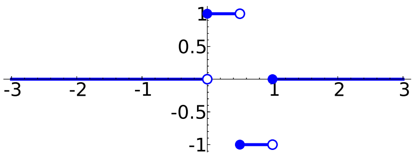
Its Fourier transform vanishes with order 2 at the origin and shows a quicker exponential decay:

$$\widehat{\psi}(\xi) = (2\pi\xi)^2 e^{-2\pi^2\xi^2} \tag{3.12}$$



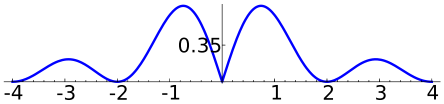
The next two wavelets receive proper names and appear in any academic textbook because they provide examples with rather explicit calculations in the next subsection. Actually, they are not very practical. The first one is the *Haar wavelet* and it was introduced at the beginning of the 20th century [Haa10] to address a problem that would be central in the development of wavelets. It is a humble horizontal broken line:

$$\psi(x) = \begin{cases} 1 & \text{if } 0 \le x < 1/2, \\ -1 & \text{if } 1/2 \le x < 1, \\ 0 & \text{otherwise.} \end{cases} \tag{3.13}$$



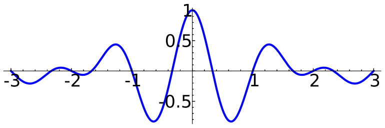
Its Fourier transform vanishes with order 1 at $\xi = 0$ and decays as $|\xi|^{-1}$, then (3.9) is assured. We plot $|\widehat{\psi}|$ because $\widehat{\psi}$ is complex.

$$\widehat{\psi}(\xi) = \frac{\big(1 - e(-\xi/2)\big)^2}{2\pi i\xi}, \qquad |\widehat{\psi}| \tag{3.14}$$



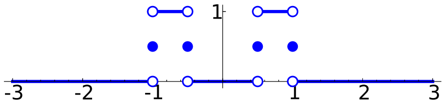
The *Shannon wavelet* involves the function sinc introduced in (1.60):

$$\psi(x) = \operatorname{sinc}\big(\frac{x}{2}\big)\cos\big(\frac{3\pi x}{2}\big) \tag{3.15}$$



Its Fourier transform is almost as simple as the Haar wavelet:

$$\widehat{\psi}(\xi) = \chi^*_{[-1,-1/2]}(\xi) + \chi^*_{[1/2,1]}(\xi) \tag{3.16}$$



where $\chi^*_{[a,b]}$ means the characteristic function of the interval $[a,b]$ putting $1/2$ as the value at the extremes.

For general normalized continuum wavelets, one can deduce an inversion formula and a Parseval identity formally similar to Proposition 3.1.1. Due to the low regularity assumed here the proof and even the statement requires finer considerations (as it happens with the Fourier transform in L^2 [DM72]). We state the result in this general context but we assume some regularity for the proof (see [Pin02, §6.2] or [Bré02, D2·2] for a proof without this assumption).

Proposition 3.1.2. *Let ψ be a normalized continuum wavelet. Then for every $f \in L^2(\mathbb{R})$*

$$f(x) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} W_\psi f(a,b)\psi_{a,b}(x)\,\frac{dadb}{a^2}. \tag{3.17}$$

where the integrals are understood as principal values[1] *at $a = 0$ and $a, b = \infty$. Moreover*

$$\int_{-\infty}^{\infty} |f|^2 = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \big|W_\psi f(a,b)\big|^2\,\frac{dadb}{a^2}. \tag{3.18}$$

Proof. Let us assume $\psi \in L^1$ (in this way $\widehat{\psi} \in L^\infty \cap C$) and $f, \widehat{f} \in L^1 \cap C^1$ to have the Fourier inversion formula at every point thanks to Theorem 1.2.4.

The Fourier transform of $\psi_{a,b}$ is $|a|^{1/2}e(-b\xi)\widehat{\psi}(a\xi)$ by (1.71) and (1.72). Then Parseval identity for the Fourier transform (1.74) implies

$$W_\psi f(a,b) = |a|^{1/2}\int_{-\infty}^{\infty} \widehat{f}(\xi)\overline{\widehat{\psi}}(a\xi)e(b\xi)\,d\xi. \tag{3.19}$$

[1] This means $\int_{|b|<M_1}\int_{\epsilon<|a|<M_2}$ with $\epsilon \to 0$ and $M_1, M_2 \to \infty$ and it converges in L^2 to the function f.

This means that $|a|^{1/2}\widehat{f}(\xi)\overline{\widehat{\psi}}(a\xi)$ is the Fourier transform of $W_\psi f(a,\cdot)$. On the other hand the Fourier transform of $\overline{\psi}_{a,\cdot}(x)$ is $|a|^{1/2}e(-x\xi)\overline{\widehat{\psi}}(a\xi)$. Then again by Parseval identity,

$$\int_{-\infty}^{\infty} W_\psi f(a,b)\psi_{a,b}(x)\,db = \int_{-\infty}^{\infty} W_\psi f(a,b)\overline{\overline{\psi}_{a,b}(x)}\,db = |a|\int_{-\infty}^{\infty} \widehat{f}(\xi)|\widehat{\psi}(a\xi)|^2 e(x\xi)\,d\xi. \tag{3.20}$$

Integrating against a^{-2} and making the change $a \mapsto a/\xi$ this is

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \widehat{f}(\xi)|a|^{-1}|\widehat{\psi}(a\xi)|^2 e(x\xi)\,d\xi da = \int_{-\infty}^{\infty} \widehat{f}(\xi)e(x\xi)\,d\xi \int_{-\infty}^{\infty} |a|^{-1}|\widehat{\psi}(a)|^2\,da. \tag{3.21}$$

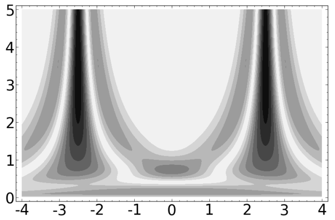
The first integral factor is $f(x)$ by the inversion formula and the second is 1 by the normalization condition. Then we have proved (3.17).

By (3.19) and the standard Parseval identity,

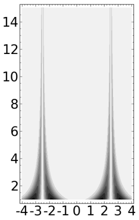
$$\int_{-\infty}^{\infty} |W_\psi f(a,b)|^2\,db = |a|\int_{-\infty}^{\infty} |\widehat{f}(\xi)|^2|\widehat{\psi}(a\xi)|^2\,d\xi. \tag{3.22}$$

Integrating against a^{-2} as before, we obtain (3.18). □

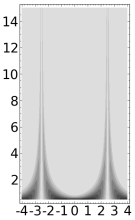
Let us illustrate the wavelet transform with the example we studied before. In some sense in a wavelet the inverse of the scale $A = 1/a$ corresponds to the frequency. The first figure below is the plot of $|W_\psi f(1/A,b)|$ for the Mexican hat wavelet and the function $e^{-32(x+5/2)^2} + e^{-32(x-5/2)^2}$ that we analyzed with the Gabor transform. To keep the analogy, A is in the vertical axis and in the same range as before.



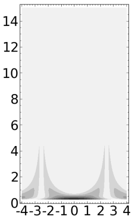
|W$_\psi$f(1/A, b)|



A > 1



A > 0.5



A > 0.25

The other plots corresponds to $|W_\psi f(1/A,b)|$ for the characteristic function of the interval $[-1,1]$ in a larger range. They show that the large values corresponding to A close to zero mask the rest of the values when distributing the contour levels. This gives an idea about the variation of these levels.

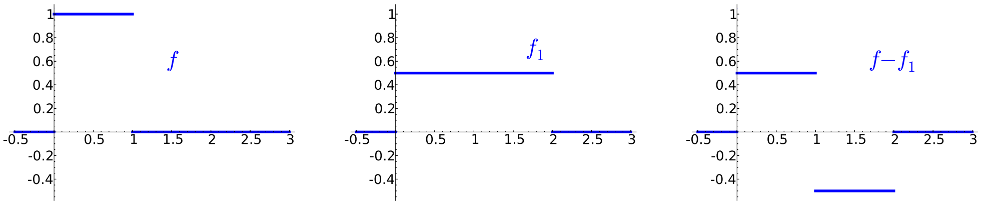
The conclusion to extract from the asymptotes is that $\psi_{a,b}$ with small scale a (large "frequency" $1/A$) only contribute when there are fine details to study. One may argue that apparently we got much better results with the Gabor transform because now the area occupied by the relevant values could be infinite. This conclusion is unclear because if we only consider absolute values of the transforms for ξ and A large, in Proposition 3.1.1 we lose the cancellation induced by the oscillation of $w_{\xi,b}$ while in Proposition 3.1.2 when $A = 1/a$ grows the mass of $\psi_{a,b}$ is constrained to a set of size comparable to a.

Truly both transforms are not very practical as they have been defined because computing numerically highly oscillatory integrals on $\mathbb{R}^2$ is not so simple to implement. In the next subsection we will see a Fourier series expansion with wavelets that is more computational friendly because it allows to consider approximations by partial sums. The most common approach in applications employs a fully discrete wavelet transform involving only a finite number of values that it is only vaguely related to (3.10).

Suggested Readings. Many texts employ the windowed Fourier transform or other alike transform to motivate the wavelet transform ant to compare them. In [Dau92, §1.2] there is an interesting discussion about this latter point. Another references are [Mal09, §4.2, §4.3], [Chu92, Ch.3] and [Kai94, Ch.2].

3.1.2 The theoretical framework

To get some intuition about the idea behind multiresolution analysis, let us consider the characteristic function f of the interval $[0,1)$ and let us try to analyze it in terms of the Haar wavelet (3.13) following an iterative procedure. We define f_1 to be the function that gives the average of f in the doubled interval $[0,2)$ along this interval and that is 0 otherwise. The difference is easily related to the Haar wavelet ψ.



In a formula, $f(x)-f_1(x)=2^{-1}\psi(x/2)$. Now f_1 is the same as f changing the scale in the X and Y axes by factors 2 and 2^{-1} and we repeat the procedure defining in general f_n as the average of f_{n-1} in its doubled support to get

(3.23)
$$f(x)-f_1(x)=\frac{1}{2}\psi(\frac{x}{2}),\quad f_1(x)-f_2(x)=\frac{1}{4}\psi(\frac{x}{4}),\ \dots\quad f_{n-1}(x)-f_n(x)=\frac{1}{2^n}\psi(\frac{x}{2^n}),\dots$$

Weierstrass M-test allows to sum all of these formulas to get

$$f(x)=\sum_{n=1}^{\infty}2^{-n}\psi(x/2^n) \tag{3.24}$$

with uniform convergence. We have also convergence in L^2 but not in L^1 because the integral of ψ vanishes and the integral of f does not.

Imagine that instead this silly f we have something more involved, say a function in L^2 (as smooth as you wish, if you prefer so). It can be approximated by step functions. If the steps are of width 2^{-m} we can perform the same analysis with each step as we did with the characteristic function of $[0,1)$ but now applying the scaling and translation $x\mapsto 2^m x-k$. When the approximation by step functions is finer we will obtain higher positive exponents

$m-n$ in the powers of 2. In the limit, assuming the convergence, we would obtain an expansion of any $f \in L^2$ in terms of the Haar wavelet:

$$f(x) = \sum_{j,k\in\mathbb{Z}} a_{jk}\psi(2^j x - k). \tag{3.25}$$

Now there is something mathematically interesting that motivates the early introduction of the Haar wavelet [Haa10]. It turns out that the functions $\psi(2^j x - k)$ are orthogonal with the $L^2(\mathbb{R})$ scalar product. As $\|\psi\|_2 = 1$, the functions

$$\psi_{jk}(x) = 2^{j/2}\psi(2^j x - k) \qquad \text{with} \quad j, k \in \mathbb{Z} \tag{3.26}$$

are orthonormal in $L^2(\mathbb{R})$ and it allows to express the coefficients in (3.25) in a closed form as we did for the Fourier coefficients in (1.34). In a formula, we have for $f \in L^2$

$$f = \sum_{j,k\in\mathbb{Z}} c_{jk}\psi_{jk} \qquad \text{with} \quad c_{jk} = \int_{-\infty}^{\infty} f\overline{\psi}_{jk}. \tag{3.27}$$

The conjugation is superfluous but we keep it to have the result for any *orthonormal basis* of $L^2(\mathbb{R})$. Here we follow the slightly confusing common terminology: In Hilbert spaces, orthonormal basis means complete orthonormal system. It is not a basis in the sense of linear algebra because in the latter only finite linear combinations are considered.

With (3.27) we have a kind of self-similar alternative for Fourier series and we could have obtained (3.24) from it. The self-similarity and the compact support are good having in mind the previous subsection, because they allow to separate neatly different scales. On the other hand, the lack of regularity does not seem very convenient.

Proposition 3.1.2 suggested a kind of orthogonality of normalized continuum wavelets but (3.27) is, no doubt, a cleaner formula and motivates to introduce a new concept of wavelet, probably the favorite for mathematicians. We define an *orthonormal wavelet* (very often simply a *wavelet*) to be a function $\psi \in L^2(\mathbb{R})$ such that (3.26) form an orthonormal basis for $L^2(\mathbb{R})$. In particular, (3.27) holds in L^2 sense, for orthonormal wavelets.

If we want to compete with Fourier system, we would like to have good convergence properties for smooth functions, say for instance $f \in C_0^\infty$. Let us think for instance about c_{j0}, if ψ is bounded we clearly have $c_{j0} = O(2^{j/2}\|f\|_1)$ that goes to 0 when $j \to -\infty$. On the other hand, using the n-th Taylor approximation of f at 0,

$$2^{j/2}c_{j0} = \int_{-\infty}^{\infty} f(2^{-j}x)\overline{\psi}(x)\,dx = \sum_{m=0}^{n-1} 2^{-jm}\frac{f^{(m)}(0)}{m!}M_m + O\big(2^{-jn}\|f^{(n)}\|_\infty\|x^n\psi\|_1\big) \tag{3.28}$$

where $M_m = \int_{-\infty}^{\infty} x^m\psi(x)\,dx$. Then if we have more vanishing moments we have a quicker decay when $j \to \infty$. It can be proved that the very definition of orthonormal wavelet if ψ is smooth and $O\big(|x|^{-\alpha}\big)$ implies $M_m = 0$ for $m < \alpha - 1$ [HW96, §2.3].

The aim is to construct orthonormal wavelets, mainly theoretically in this subsection. First of all, let us state a nice result characterizing the orthonormality under translations.

Proposition 3.1.3. *Let $f \in L^2(\mathbb{R})$. Then $\{f(\cdot - k)\}_{k\in\mathbb{Z}}$ is an orthonormal system if and only if*

$$\sum_{k\in\mathbb{Z}} |\widehat{f}(\xi + k)|^2 = 1 \qquad \textit{almost everywhere.} \tag{3.29}$$

Proof. By Parseval identity,

$$\int_{-\infty}^{\infty} \overline{f}(x) f(x-k)\, dx = \int_{-\infty}^{\infty} |\widehat{f}(\xi)|^2 e(-k\xi)\, d\xi = \int_0^1 \sum_{l\in\mathbb{Z}} |\widehat{f}(\xi + l)|^2 e(-k\xi)\, d\xi \tag{3.30}$$

where in the last equality the reasoning is as in (2.10) and the interchange of the sum and the integral is justified by Lebesgue's dominated convergence theorem. If $\{f(\cdot - k)\}_{k\in\mathbb{Z}}$ is orthonormal then the sum in (3.29) minus 1 defines an integrable function on $\mathbb{T}$ and the integral equality shows that all of its Fourier coefficients are zero, then it is zero almost everywhere[2] [Zyg88, §I.6]. The converse is similar. □

It is possible to generalize this result in the following way: The sum in (3.29) is bounded by positive constants c_1 and c_2 from below and above if and only if $\{f(\cdot - k)\}_{k\in\mathbb{Z}}$ is a *Riesz system* with the same constants. This means

$$c_1 \sum |\lambda_k|^2 \le \big\| \sum \lambda_k f(\cdot - k)\big\|_2^2 \le c_2 \sum |\lambda_k|^2 \qquad \text{for any } \{\lambda_k\}_{k\in\mathbb{Z}} \in \ell^2(\mathbb{Z}). \tag{3.31}$$

As a curiosity, for the Haar wavelet the orthonormality of the translates is obvious but (3.29) is not and the same can be said for the characteristic function of $[-1/2, 1/2)$. Recalling (3.14) and (1.60) we have, respectively

$$\sum_{k\in\mathbb{Z}} \frac{\sin^4\big(\pi(\xi+k)/2\big)}{(\xi+k)^2} = \frac{\pi^2}{4} \qquad \text{and} \qquad \sum_{k\in\mathbb{Z}} \frac{\sin^2(\pi(\xi+k))}{(\xi+k)^2} = \pi^2. \tag{3.32}$$

For the Shannon wavelet (3.15) and any of its translations the condition (3.29) is obvious by (3.16) and Proposition 3.1.3 gives the orthogonality with zero calculations.

There exists a full characterization of orthonormal wavelets using Proposition 3.1.3 and a corresponding result involving the scaling [HW96, §7.1] but it does not any clue about how to fulfill our aim of constructing wavelets.

The common theoretical approach is to define firstly the spaces in which we are going to work at each scale and the "brick" we are going to use. To get (3.25) for the Haar wavelet we employed the approximation of L^2 functions in spaces of step functions with step-width a power of 2. Each step is a scaled version of the characteristic function of $[0, 1)$ that becomes our brick.

[2] I do not resist the temptation of citing [New74] with a surprising and short complex analysis proof of the analogue of this fact for Fourier integrals.

In general, a *multiresolution analysis*, abbreviated *MRA*, is a sequence of nested subspaces $V_j \subset V_{j+1} \subset L^2(\mathbb{R})$, $j \in \mathbb{Z}$, and a function $\phi \in L^2(\mathbb{R})$ such that

$$\left\{ \begin{array}{ll} \text{a)} & \{\phi(\cdot - k)\}_{k\in\mathbb{Z}} \text{ is an orthonormal basis of } V_0, \\ \text{b)} & V_j = \{f \,:\, f(2^{-j}\cdot) \in V_0\}, \\ \text{c)} & \bigcup V_j \text{ is dense in } L^2(\mathbb{R}) \text{ and } \bigcap V_j = \{0\}. \end{array} \right. \tag{3.33}$$

The function is called the *scaling function* of the MRA. It is the "brick" that dictates the subspaces when we combine the first and the second properties. Note that $2^{1/2}\phi(2x-k)$ is an orthonormal basis of V_1 and in general $2^{j/2}\phi(2^jx-k)$ is an orthonormal basis of V_j. It does not mean that ϕ is an orthonormal wavelet. At the contrary, the inclusion $V_{-1} \subset V_0$ implies that the functions $2^{-1/2}\phi(x/2-k)$ can be expanded in terms of $\phi(x-k)$, in particular they are not orthogonal to them. Let c_n be the coefficients of $2^{-1/2}\phi(x/2)$ and define $\sqrt{2}m_0(-\xi)$ having these Fourier coefficients:

$$2^{-1/2}\phi(x/2) = \sum_{k\in\mathbb{Z}} c_k\phi(x-k) \qquad \text{and} \qquad m_0(\xi) = 2^{-1/2}\sum_{k\in\mathbb{Z}} c_k e(-k\xi). \tag{3.34}$$

The role of $m_0(\xi)$ is clarified if we take Fourier transforms in the first equation to get

$$\widehat{\phi}(2\xi) = m_0(\xi)\widehat{\phi}(\xi). \tag{3.35}$$

Then $m_0 \in L^2(\mathbb{T})$ indicates how to filter the frequencies of the signal ϕ when the scale is changed. In the literature m_0 is called the *low-pass filter* of the MRA. In some way, the functions in V_{-1} oscillate twice slower than in V_0 then m_0 suppresses high frequencies, it is truly a low-pass filter.

Comparing with the example with the Haar wavelets, the V_j would be the spaces of finer and finer step functions and the wavelets of different scales live in the "differences" between consecutive spaces. More precisely, let us define W_j to be the orthogonal complement of V_j in V_{j+1}

$$W_j = V_{j+1} \cap V_j^{\perp}. \tag{3.36}$$

If we find a ψ such that $\{\psi(\cdot - k)\}_{k\in\mathbb{Z}}$ is an orthonormal basis of W_0, then $\{\psi_{jk}\}_{k\in\mathbb{Z}}$ is an orthonormal basis of W_j for each j and this is enough to prove that ψ is a wavelet because the third property of (3.1.4) implies

$$V_j = \bigoplus_{k<j} W_k \qquad \text{and} \qquad L^2(\mathbb{R}) = \overline{\bigoplus_{j\in\mathbb{Z}} W_j}. \tag{3.37}$$

In some sense, W_k is the "difference" between V_{k+1} and V_j and these sums telescope resembling the argument in (3.23) and (3.24).

The main result in this subsection says that if we have a MRA we can construct an orthonormal wavelet $\psi \in W_0$ from ϕ and m_0.

Theorem 3.1.4. *Let ϕ be the scaling function of a MRA and m_0 the associated low-pass filter defined by* (3.34) *or* (3.35). *Then a function $\psi \in W_0$ is an orthonormal wavelet if and only if*

$$\widehat{\psi}(2\xi) = e(\xi)\nu(2\xi)\overline{m}_0(\xi + \frac{1}{2})\widehat{\phi}(\xi) \tag{3.38}$$

almost everywhere for some $\nu \in L^2(\mathbb{T})$ with $|\nu(\xi)| = 1$.

The proof of this result takes some effort. Before entering into it let us see two examples producing the simplest orthonormal wavelets.

If ϕ is the characteristic function of $[0, 1)$ the V_j's are spaces of step functions which become finer when j grows and the conditions of MRA are fulfilled. It is clear

$$2^{-1/2}\phi(x/2) = 2^{-1/2}\phi(x) + 2^{-1/2}\phi(x - 1). \tag{3.39}$$

Then by (3.34), $m_0(\xi) = \big(1+e(-\xi)\big)/2$. By Theorem 3.1.4 we have infinitely many wavelets to our disposal choosing ν. Let us take $\nu(\xi) = -e(-\xi)$ to get

$$2\widehat{\psi}(2\xi) = \widehat{\phi}(\xi) - e(-\xi)\widehat{\phi}(\xi). \tag{3.40}$$

Taking Fourier inverses, $\psi(x/2) = \phi(x) - \phi(x - 1)$ that is just the definition of the Haar wavelet (3.13).

Now we take $\phi(x) = \operatorname{sinc} x$. Note that $\widehat{\phi} = \chi^*_{[-1/2,1/2]}$ with χ^* as in (3.16) that satisfies (3.29) trivially, then the first condition in (3.33) is fulfilled for V_0 generated by the orthonormal basis, the second can be taken as a definition and the third could be checked noting that the Fourier transforms of the finite linear combinations of $\phi(2^j x - k)$ are step functions as before, but we skip this point now because later we will see a general result that gives a simple condition to get the last property in (3.33). The relation (3.35) proves $m_0 = \chi^*_{[-1/4,1/4]}$ for $|\xi| < 1/2$ and by the periodicity

$$m_0(\xi) = \sum_{k\in\mathbb{Z}} \chi^*_{[-1/4,1/4]}(\xi + k) \qquad \text{almost everywhere.} \tag{3.41}$$

Choosing $\nu(\xi) = e(-\xi)$ in (3.38), we have

$$\widehat{\psi}(2\xi) = e(-\xi)\big(\chi^*_{[-1/2,-1/4]}(\xi) + \chi^*_{[1/4,1/2]}(\xi)\big) = e(-\xi)\widehat{\psi}_S(2\xi) \tag{3.42}$$

where ψ_S is the Shannon wavelet as in (3.15) and (3.16). Then we conclude that $\psi_S(x-1/2)$ is an orthonormal wavelet.

The choice $\nu(\xi) = -e(-\xi)$ employed before leads to a neat expansion of the wavelet in terms of the scaling function

Corollary 3.1.5. *If ϕ is the scaling function of a MRA then*

$$\psi(x) = \sqrt{2}\sum_{k\in\mathbb{Z}}(-1)^k\overline{c}_{1-k}\phi(2x - k), \tag{3.43}$$

with c_k as in (3.34), *is an orthonormal wavelet.*

Proof. By (3.1.4) with $\nu(\xi) = -e(-\xi)$ and the formula for m_0 in (3.34),

$$\widehat{\psi}(\xi) = -2^{-1/2} \sum_{k\in\mathbb{Z}} (-1)^k \overline{c}_k e((k-1)\xi/2)\widehat{\phi}(\xi/2). \tag{3.44}$$

By Fourier inversion

$$\psi(x) = -2^{1/2} \sum_{k\in\mathbb{Z}} (-1)^k \overline{c}_k \phi(2x+k-1), \tag{3.45}$$

that is the result when $k \mapsto 1-k$. □

To prove Theorem 3.1.4, we proceed in two steps characterizing the Fourier transform of the functions belonging to the space V_{-1} and to its orthogonal complement in V_0. We follow [HW96, §2.2] through [LW18] to keep the usual normalization of the Fourier transform.

Lemma 3.1.6. *In a MRA with scaling function ϕ,*

$$V_{-1} = \Big\{ f \in L^2(\mathbb{R}) \ : \ \widehat{f}(\xi) = p(2\xi) m_0(\xi)\widehat{\phi}(\xi) \quad \textit{for some} \quad p \in L^2(\mathbb{T}) \Big\}. \tag{3.46}$$

Lemma 3.1.7. *In a MRA with scaling function ϕ,*

$$W_{-1} = \Big\{ f \in L^2(\mathbb{R}) \ : \ \widehat{f}(\xi) = e(\xi) p(2\xi) \overline{m}_0(\xi + \frac{1}{2})\widehat{\phi}(\xi) \quad \textit{for some} \quad p \in L^2(\mathbb{T}) \Big\}. \tag{3.47}$$

The first result follows easily from the definition of V_{-1} through the second property of (3.33).

Proof of Lemma 3.1.6. We know that $\{\phi(\cdot - k)\}_{k\in\mathbb{Z}}$ spans V_0 then any $f \in V_{-1}$ can be expanded as

$$f(x) = \sum_{k\in\mathbb{Z}} a_k \phi(x-k) \qquad \text{with} \quad \{a_k\}_{k\in\mathbb{Z}} \in \ell^2(\mathbb{Z}). \tag{3.48}$$

Taking Fourier transforms,

$$\widehat{f}(\xi) = 2\sum_{k\in\mathbb{Z}} a_k e(-2k\xi)\widehat{\phi}(2\xi) = 2m_0(\xi)\widehat{\phi}(\xi) \sum_{k\in\mathbb{Z}} a_k e(-2k\xi) \tag{3.49}$$

by (3.35). As $\{a_k\}_{k\in\mathbb{Z}} \in \ell^2(\mathbb{Z})$ is arbitrary, every function of $L^2(\mathbb{T})$ evaluated at 2ξ can be written as the previous sum. □

In the proof of the second result, a noticeable identity plays a role which we separate for further reference.

Lemma 3.1.8. *Let m_0 be the low-pass filter of a MRA, defined by* (3.34)*. Then*

$$|m_0(\xi)|^2 + |m_0(\xi + 1/2)|^2 = 1 \quad \textit{almost everywhere.} \tag{3.50}$$

Proof. By Proposition 3.1.3, (3.35) and the periodicity of m_0,

$$(3.51)\quad 1 = \sum_{k\in\mathbb{Z}} |\widehat{\phi}(2\xi + k)|^2 = |m_0(\xi)|^2 \sum_{2|k} |\widehat{\phi}(\xi + k/2)|^2 + |m_0(\xi + 1/2)|^2 \sum_{2\nmid k} |\widehat{\phi}(\xi + k/2)|^2.$$

In the right hand side both sums equal 1 by Proposition 3.1.3, the first one by a direct application and the second one replacing ξ by $\xi - 1/2$. □

Proof of Lemma 3.1.7. If $f \in V_0$ then $f(x) = \sum a_k\phi(x-k)$ with $\{a_k\}_{k\in\mathbb{Z}} \in \ell^2(\mathbb{Z})$ or, equivalently, the functions of V_0 are characterized by

$$(3.52)\qquad \widehat{f}(\xi) = q(\xi)\widehat{\phi}(\xi) \qquad \text{with} \quad q(\xi) = \sum_{k\in\mathbb{Z}} a_k e(-k\xi) \in L^2(\mathbb{T}).$$

The necessary and sufficient condition to have $f \in V_{-1}^{\perp}$ is, after Parseval identity and Lemma 3.1.6,

$$(3.53)\qquad 0 = \int_{-\infty}^{\infty} \overline{p}(2\xi)\overline{m}_0(\xi)q(\xi)|\widehat{\phi}(\xi)|^2\, d\xi = \sum_{k\in\mathbb{Z}} \int_0^1 \overline{p}(2\xi)\overline{m}_0(\xi)q(\xi)|\widehat{\phi}(\xi+k)|^2\, d\xi$$

for every $p \in L^2(\mathbb{T})$, where we have used the periodicity for the last equality. Using Proposition 3.1.3 and defining $F(\xi) = e(-\xi)\overline{m}_0(\xi)q(\xi)$, the condition is

$$(3.54)\qquad 0 = \int_0^1 e(\xi)\overline{p}(2\xi)F(\xi)\, d\xi = \int_0^{1/2} e(\xi)\overline{p}(2\xi)\big(F(\xi) - F(\xi+1/2)\big)\, d\xi.$$

As p is arbitrary in $[0, 1/2]$, this is the same as saying that F is 1/2-periodic. Lemma 3.1.8 implies that $m_0(\xi)$ and $m_0(\xi + 1/2)$ are bounded and do not vanish simultaneously. Then we can write $F(\xi) = F(\xi + 1/2) = \overline{m}_0(\xi)\overline{m}_0(\xi + 1/2)p(2\xi)$ with $p \in L^2(\mathbb{T})$ arbitrary. Then $q(\xi) = e(\xi)\overline{m}_0(\xi + 1/2)p(2\xi)$ and substituting in (3.52) the proof is finished. □

Proof of Theorem 3.1.4. Let us check firstly that any orthonormal wavelet $\psi \in W_0$ is of the form (3.38). It is easy to see by the second property of (3.33) and the definition (3.36),

$$(3.55)\qquad f \in W_0 \qquad \text{if and only if} \qquad f(2^j\cdot) \in W_j.$$

Considering $j = -1$ and taking Fourier transforms, we have by Lemma 3.1.7

$$(3.56)\qquad \widehat{\psi}(2\xi) = e(\xi)p(2\xi)\overline{m}_0(\xi + \frac{1}{2})\widehat{\phi}(\xi).$$

We apply Proposition 3.1.3 separating even and odd integers as in the proof of Lemma 3.1.8,

$$(3.57)\quad \sum_{k\in\mathbb{Z}} |\widehat{\psi}(2\xi+k)|^2 = |p(2\xi)|^2|m_0(\xi+1/2)|^2 \sum_{2|k} |\widehat{\phi}(\xi+k/2)|^2 + |p(2\xi)|^2|m_0(\xi)|^2 \sum_{2\nmid k} |\widehat{\phi}(\xi+k/2)|^2.$$

The sums are 1 by Proposition 3.1.3 and Lemma 3.1.8 gives $|p(2\xi)| = 1$ as required in (3.38).

We have to prove now that (3.38) defines an orthonormal wavelet. It belongs to W_0 by Lemma 3.1.7 and (3.55) with $j = -1$. We can write any $f \in W_{-1}$ as

$$\widehat{f}(\xi) = p(2\xi)\overline{\nu}(2\xi)\widehat{\psi}(2\xi) \tag{3.58}$$

If $\sum a_k e(-k\xi)$ is the Fourier expansion of the periodic function $p(\xi)\overline{\nu}(\xi)$, taking inverse Fourier transforms

$$f(x) = \frac{1}{2}\sum_{k\in\mathbb{Z}} c_k \psi(x/2 - k). \tag{3.59}$$

Then $\{\psi_{-1k}\}_{k\in\mathbb{Z}}$ spans W_{-1} and it is orthonormal proceeding as in (3.57). By (3.55), $\{\psi_{0k}\}_{k\in\mathbb{Z}}$ is an orthonormal basis of W_0 and as explained in (3.37) it is enough to assure that ψ is an orthonormal wavelet. □

In case you are wondering, it is possible to obtain examples of wavelets not associated to any MRA [Vya09, Th.3.4] but they cannot have a continuous Fourier transform with good decay [HW96, §7.3, Cor.3.16].

Suggested Readings. The rigorous development of the theory can be read in books addressed to a mathematical audience. My favorites are [Bré02], [Pin02] and [HW96]. The latter, one of the pioneering textbooks, takes quite an effort to discuss some properties related to analysis, like convergence or atomic decomposition of functions. This is also the case in [Woj97]. A brief and clear introduction to wavelets is also included in [GG12]. By reasons that probably rely on tradition, my impression is that very rarely the literature about wavelets uses the most standard normalization of the Fourier transform (1.36), [Pin02] is an exception.

3.1.3 Construction of wavelets

In principle Theorem 3.1.4 and Corollary 3.1.5 offer a recipe to create wavelets: Take the scaling function of a MRA, compute the coefficients in (3.34) and you are done; applying the formulas the outcome must be an orthonormal wavelet. The problem is that it is not clear how difficult is to create a MRA. Let us say that one starts with a function ϕ such that $\{\phi(\cdot - k)\}_{k\in\mathbb{Z}}$ is orthonormal and define V_0 to fulfill the first property in (3.33). The second property can be also taken as a definition but the third property seems very hard to check. Why should $\bigcup V_j$ generate $L^2(\mathbb{R})$?

Let us adopt an optimistic attitude thinking about the case of the Haar wavelet. The scaling function was the characteristic function of $[0, 1)$ and it was as easy as saying that any L^2 function is a limit of step functions. If instead of the characteristic function of $[0, 1)$ we have something like a tent function, forgetting about the first property we have that the step functions are replaced by piecewise linear functions once we have adjusted the width of the tent function to avoid empty room. This also works and in general it seems that anything that can resemble a "lump" after scaling should work. It is actually true, under natural technical conditions, if finer scales are really finer and you avoid zero average functions the third property in (3.33) is assured.

Theorem 3.1.9. *Let $\phi \in L^2(\mathbb{R})$ be such that $\{\phi(\cdot - k)\}_{k\in\mathbb{Z}}$ is orthonormal, $\phi(x) = \sum_{k\in\mathbb{Z}} a_k\phi(2x-k)$ for some $\{a_k\}_{k\in\mathbb{Z}} \in \ell^2(\mathbb{Z})$ and $\widehat{\phi}$ is continuous at 0 with $\widehat{\phi}(0) \neq 0$. Then the spaces V_j spanned by $\{\phi(2^j\cdot -k)\}_{k\in\mathbb{Z}}$ define a MRA with scaling function ϕ.*

As a byproduct of the proof it follows that the only possibility to fulfill the hypotheses in this result is

$$|\widehat{\phi}(0)| = 1. \tag{3.60}$$

The second condition assures that we are not leaving "empty room" when considering a finer scale and the third is the zero average ban.

Proof. Let us consider the projection operator onto V_j

$$\pi_j f = \sum_{k\in\mathbb{Z}} \langle f, \phi_{jk}\rangle \phi_{jk} \qquad \text{with} \quad \phi_{jk}(x) = 2^{j/2}\phi(2^j x - k). \tag{3.61}$$

If we prove $\|\pi_j f\|_2 \to 0$ when $j \to -\infty$ for every $f \in L^2(\mathbb{R})$ we will have deduce $\bigcap V_j = \{0\}$ because $f \in \bigcap V_j$ would satisfy $\pi_j f = f$ for every j. Projections are bounded operators then, by density we can assume $f \in C_0^\infty$. If $[-M, M]$ contains the support of f

$$\|\pi_j f\|_2^2 = \sum_{k\in\mathbb{Z}} \big|\langle f, \phi_{jk}\rangle\big|^2 \le \|f\|_2^2 \sum_{k\in\mathbb{Z}} \int_{-M}^{M} |\phi_{jk}|^2 = \|f\|_2^2 \int_{U_j} |\phi|^2 \tag{3.62}$$

where $U_j = \bigcup_{k\in\mathbb{Z}} \big(-k - 2^j M, -k + 2^j M\big)$. As $\bigcap U_j$ has vanishing measure, we have $\|\pi_j f\|_2 \to 0$ by Lebesgue's dominated convergence theorem.

If $\overline{\bigcup V_j} \neq L^2(\mathbb{R})$ then taking $f \neq 0$ in the orthogonal complement of the union we would have $\|f - \pi_j f\|_2 = \|f\|_2$ for every j, then it is enough to prove that under our hypotheses

$$\lim_{j\to+\infty} \|f - \pi_j f\|_2 \neq \|f\|_2 \qquad \text{for any } f \in L^2(\mathbb{R}) - \{0\}. \tag{3.63}$$

Again by density we can assume $\widehat{f} \in C_0^\infty$. If $[-M, M]$ contains the support of $\widehat{f}$

$$\|f - \pi_j f\|_2^2 = \|f\|^2 - \|\pi_j f\|_2^2 = \|f\|^2 - \sum_{k\in\mathbb{Z}} \big|\langle f, \phi_{jk}\rangle\big|^2 = \|f\|^2 - \sum_{k\in\mathbb{Z}} \Big| \int_{-M}^{M} \widehat{f}\,\overline{\widehat{\phi}_{jk}} \Big|^2 \tag{3.64}$$

by Parseval identity. For j large enough we can replace M by 2^{j-1}, then the integral equals

$$\int_{-2^{j-1}}^{2^{j-1}} \widehat{f}(\xi) 2^{-j/2} \overline{\widehat{\phi}}(2^{-j}\xi) e\big(-2^{-j}k\xi\big)\, d\xi = 2^{j/2} \int_{-1/2}^{1/2} \widehat{f}(2^j\xi)\overline{\widehat{\phi}}(\xi) e(-k\xi)\, d\xi. \tag{3.65}$$

This means that it is the k-th Fourier coefficient of the periodic extension of the function $2^{j/2}\widehat{f}(2^j\xi)\overline{\widehat{\phi}}(\xi)$. Hence, Parseval identity again but this time for Fourier series, proves

$$\|f - \pi_j f\|_2^2 = \|f\|^2 - 2^j \int_{-1/2}^{1/2} |\widehat{f}(2^j\xi)|^2 |\widehat{\phi}(\xi)|^2\, d\xi = \|f\|^2 - \int_{-M}^{M} |\widehat{f}(\xi)|^2 |\widehat{\phi}(2^{-j}\xi)|^2\, d\xi. \tag{3.66}$$

By the continuity, $\widehat{\phi}(2^{-j}\xi) \to \widehat{\phi}(0)$, therefore $\|f - \pi_j f\|_2^2 \to \|f\|_2^2 - |\widehat{\phi}(0)|^2 \|f\|_2^2$ and (3.63) follows.

Finally, note that $\overline{\bigcup V_j} = L^2(\mathbb{R})$ implies $\|f - \pi_j f\|_2 \to 0$ then necessarily $|\widehat{\phi}(0)| = 1$. □

Let us see how the application of Theorem 3.1.9 gives rise to specific examples of wavelets. We illustrate it with the *Meyer wavelet* and the *Franklin wavelet*. In both cases one looks for a trick to get automatically the needed scaling relation between $\phi(x)$ and $\phi(2x-k)$.

We can force ϕ to satisfy (3.29) adjusting $\widehat{\phi}$ as the square root of a real function. The idea of doing this fulfilling at the same time the scaling relation leads to *Meyer wavelets* [Mey87] [LM86]. We start with any nonincreasing function $f:[0,1]\longrightarrow[0,1]$ satisfying

$$f(x)+f(1-x)=1 \qquad \text{and} \qquad f(x)=1 \quad \text{for } x\in[0,1/3]. \tag{3.67}$$

Then one defines

$$\phi(x)=\int_{-\infty}^{\infty}F(\xi)e(x\xi)\,d\xi \qquad \text{with} \quad F(\xi)=\begin{cases}\sqrt{f(|\xi|)} & \text{for } |\xi|\le 1,\\ 0 & \text{for } |\xi|\ge 1.\end{cases} \tag{3.68}$$

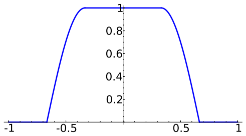
It is clear that F is as smooth as f that is our choice. Then $\phi=\widehat{F}$ is as smooth and as quickly decaying as we wish. Note that (3.29) reduces to

$$\sum_{k\in\mathbb{Z}}|F(\xi+k)|^2=f(|\xi|)+f(|1+\xi|)=f(-\xi)+f(1+\xi)=1 \qquad \text{for } -1\le\xi\le 0 \tag{3.69}$$

and by the periodicity it holds for every $\xi\in\mathbb{R}$. It is not difficult to check that under (3.67) we have $\widehat{\phi}(2\xi)=m_0(\xi)\widehat{\phi}(\xi)$ where m_0 is the periodic extension of $F(2\xi)$ for $|\xi|\le 1/2$. The key point is that $m_0(\xi)$ and $\widehat{\phi}(2\xi)$ both vanish for $\xi\in[1/3,2/3]$ and they coincide for $\xi\in[0,1/3]$ (see the details in [Pin02, §6.4.4]). Taking inverse Fourier transforms one concludes that we have a scaling relation for ϕ as required in Theorem 3.1.9.

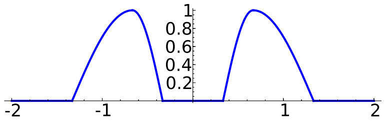
For $f=\chi^*_{[-1/2,1/2]}$ restricted to $[0,1]$ one gets $\phi(x)=\operatorname{sinc}x$ and from it the Shannon wavelet (up to a translation), as we have seen. In this sense, the Shannon wavelet is the simplest of the Meyer wavelets. A nontrivial continuous choice of f due to Meyer is

$$f(x)=\begin{cases}1 & \text{if } x\in[0,1/3],\\ \cos^2\big((3x-1)\pi/2\big) & \text{if } x\in[1/3,2/3],\\ 0 & \text{if } x\in[2/3,1].\end{cases} \qquad F(x)= \tag{3.70}$$



Using (3.68) and (3.38) with $\nu=1$ one obtains (see the calculations in [Bré02, D4·1])

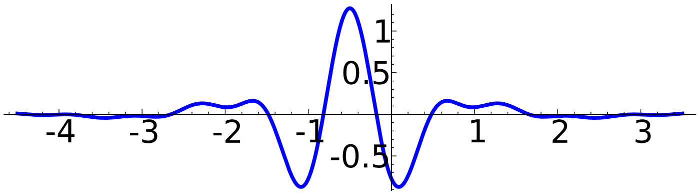
$$\widehat{\psi}(\xi)=\begin{cases}-e^{i\pi\xi}\cos\left(\frac{3\pi}{2}|\xi|\right) & \text{if } \frac13\le|\xi|\le\frac23,\\ e^{i\pi\xi}\sin\left(\frac{3\pi}{4}|\xi|\right) & \text{if } \frac23\le|\xi|\le\frac43,\\ 0 & \text{otherwise.}\end{cases} \qquad e^{-i\pi\xi}\widehat{\psi}= \tag{3.71}$$



Inverting this Fourier transform is a tedious trivial computation giving

$$\psi(x)=\frac{1}{3\pi}F\Big(\frac{2x+1}{3}\Big) \qquad \text{with} \qquad F(x)=\frac{4\cos(\pi x)}{1-4x^2}+\frac{8\cos(4\pi x)}{1-16x^2}+\frac{24x\sin(2\pi x)}{(1-4x^2)(1-16x^2)}. \tag{3.72}$$

This is sometimes called the *Meyer wavelet*. The aspect of this fully explicit wavelet $\psi \in C^\infty \cap L^1$ is:



Another way of using Theorem 3.1.9 is looking for a function f with an easy relation between $f(x)$ and $f(2x-k)$, as in (3.39) to get the Haar wavelet, and adjust in some way (3.29). Let us choose the *tent function* $f(x) = \max(1-|x|, 0)$. It satisfies

$$f(x) = \frac{1}{2}f(2x+1) + f(2x) + \frac{1}{2}f(2x-1). \tag{3.73}$$

The set $\{f(\cdot - k)\}_{k\in\mathbb{Z}}$ generates the piecewise linear L^2 functions that equal their linear interpolation at the integers. This set is almost orthogonal because only contiguous elements have nonzero scalar product. It does not satisfy (3.29) but the Riesz system condition (3.31). In fact using Poisson summation formula it can be proved

$$\sum_{k\in\mathbb{Z}} |\widehat{f}(\xi+k)|^2 = \frac{1}{3} + \frac{2}{3}\cos^2(\pi\xi). \tag{3.74}$$

Now the trick is to force (3.29) dividing the Fourier transform by the square root of this quantity, say $r(\xi)$. In this way our candidate for scaling function is $\widehat{\phi}(\xi) = \widehat{f}(\xi)/r(\xi)$. In general a relation like (3.73) is equivalent to $\widehat{f}(2\xi) = p(\xi)\widehat{f}(\xi)$ for $p \in L^2(\mathbb{T})$. Then $\widehat{\phi}(2\xi) = q(\xi)\widehat{\phi}(\xi)$ with $q(\xi) = r(\xi)p(\xi)/r(2\xi) \in L^2(\mathbb{T})$ or equivalently $\phi(x)$ can be expanded in terms of $\phi(2x-k)$ and Theorem 3.1.9 proves that we have a MRA (see also [Bré02, D4·2]). In our case, we have

$$\widehat{\phi}(\xi) = \frac{\sqrt{3}\widehat{f}(\xi)}{\sqrt{1+2\cos^2(\pi\xi)}} = \frac{\sqrt{3}\sin^2(\pi\xi)}{\pi^2\xi^2\sqrt{1+2\cos^2(\pi\xi)}}. \tag{3.75}$$

Note that Fourier inversion in the first equality allows to write $\phi(x) = \sum \lambda_k f(x-k)$ and this implies that ϕ is linear on each interval $[k, k+1]$, hence any $g \in V_1$ can be got from linear interpolation of the values $g(k/2)$. It particular it applies to the wavelet associated to ϕ.

Once we have the scaling function, (3.35) gives

$$m_0(\xi) = \sqrt{\frac{1+2\cos^2(\pi\xi)}{1+2\cos^2(2\pi\xi)}}\,\cos^2(\pi\xi). \tag{3.76}$$

Now, substituting in (3.1.4) with $\nu = 1$ and replacing ξ by $\xi/2$, we get

$$\widehat{\psi}(\xi) = e(\xi/2)h(\xi) \qquad \text{with} \quad h(\xi) = \frac{4\sqrt{3}\sin^4(\pi\xi/2)}{\pi^2\xi^2\sqrt{1+2\cos^2(\pi\xi)}}\sqrt{\frac{1+2\sin^2(\pi\xi/2)}{1+2\cos^2(\pi\xi/2)}}. \tag{3.77}$$

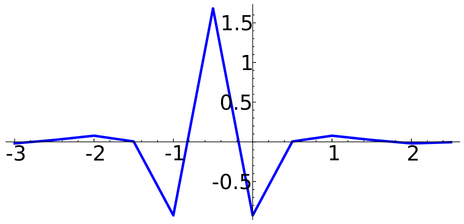
Finally, ψ can be obtained with linear interpolation of the values

$$\psi(k/2) = \int_{-\infty}^{\infty} h(\xi) e((k+1)\xi/2)\, d\xi = \int_{-\infty}^{\infty} h(\xi) \cos\left(\pi(k+1)\xi\right)\, d\xi. \tag{3.78}$$

Numerical calculations give
(3.79)

$$\begin{cases} \psi(-3) = \psi(2) & = & -0.020 \\ \psi(-5/2) = \psi(3/2) & = & 0.024 \\ \psi(-2) = \psi(1) & = & 0.077 \\ \psi(-3/2) = \psi(1/2) & = & 0.005 \\ \psi(-1) = \psi(0) & = & -0.927 \\ \psi(-1/2) & = & 1.682 \end{cases}$$

$\psi =$



The resulting wavelet is called the *Franklin wavelet* because in [Fra28], published in 1928, the orthogonalization of translations of the tent function was employed to get a continuous orthonormal system.

One could also apply the previous method to a piecewise quadratic function or to a cubic B-spline or to higher degree spline. This gives wavelets with higher regularity (they are called in general *spline wavelets*).

We finish this subsection mentioning the allegedly best method to construct wavelet. We do not deepen into it because it will reappear in some way in the discrete setting. The idea is that if we normalize $\phi(0) = 1$, which is harmless after (3.60), the iteration of (3.35) allows to define ϕ out of m_0 with

$$\widehat{\phi}(\xi) = \prod_{j=1}^{\infty} m_0(2^{-j}\xi). \tag{3.80}$$

A key point is the convergence. It is clear that we need $m_0(0) = 1$. Taking as granted this product makes sense, in principle it is not clear whether (3.29) holds. If we come back to the proof of Lemma 3.1.8 we see that it is equivalent to impose (3.50). In [Pin02, §6.5.1] to solve these problems it is required a non vanishing condition to avoid the product to diverge to zero and some strong continuity around zero. Namely, if we invent any $m_0 \in L^2(\mathbb{T})$ with $m_0(0) = 1$ satisfying (3.50) and the technical conditions

$$|m_0(\xi)| \ge c > 0 \quad \text{for } |\xi| \le \frac{1}{4} \qquad \text{and} \qquad |m_0(\xi) - 1| \log^2 \frac{1}{|\xi|} < c' \quad \text{for } |\xi| \le \frac{1}{2} \tag{3.81}$$

then (3.80) makes sense and gives the scaling function of a MRA and we can produce a wavelet with Theorem 3.1.4 or its corollary. I. Daubechies exploited the case in which $m_0(\xi)$ is a trigonometric polynomial. In this situation, the inverse Fourier transform of $m_0(2^{-j}\xi)$ is having in mind (1.37), a sum of Dirac deltas concentrated in smaller zones when j grows. Inverting (3.80) we get a convolution of them leading to a compactly supported scaling function ϕ and Corollary 3.1.5 proves that the corresponding wavelet has also compact support. The regularity of them is related to the degree. In this way one obtains localized wavelets with arbitrary regularity. The only drawback is that they do not admit explicit formulas. As we will see this is less important in practice when working in the discrete setting.

Suggested Readings. In the references already mentioned [Bré02], [Pin02] and [HW96], one can find the theory of the construction of wavelets. The case of compactly supported wavelets is discussed with detail in the two latter. In [Kam07, Ch.10] it is achieved a good balance between the development of the theory and the practice as a motivation (and the same applies to the whole book). One can find in the literature *biorthogonal wavelets*, a more general definition of wavelet dropping the condition of giving an orthonormal basis allowing more freedom for constructing them. They are discussed in [Mal09, §7.4] and in [Her18, §10]. This latter reference is a very good outreach paper (for mathematicians) that surveys all the contents of this chapter.

3.2 Wavelets in practice

3.2.1 The discrete wavelet transform

Daubechies introduced in [Dau88] the construction of a family of compactly supported wavelets as described above. Unlike the Haar wavelet, they are continuous and in fact the regularity can be tuned.

Arguably *Daubechies wavelets* constitute the most universally known, the most interdisciplinary, examples of wavelets. Instead of using the general results we have studied so far, we are going to introduce them in terms of linear filters acting on finite discrete signals. In principle this has nothing to do with our definition of wavelets but later we will see the connection. In fact it is completely equivalent to construct Daubechies wavelets using Theorem 3.1.9 and Corollary 3.1.5, or (3.80) as in [Dau92], and to construct them as a limit of the application of filters to a finite discrete signal when its length goes to infinity. In practice we usually have access only to a finite set of values or the signal is per se of this kind, then having this finite version of the wavelets that gives "mathematical wavelets" in the limit is very interesting. The majority of the real world applications labeled as based on wavelets are actually based on these finite wavelets and harmonic analysts should celebrate it without any remorse.

Consider a finite signal represented by a vector $\vec{x} = (x_0, x_1, \dots, x_{N-1})^t \in \mathbb{C}^N$ where $N = 2^K$ with $K \ge 2$. If we recall the strategy to detect edges in images with linear filters, we have many possibilities to detect a lack of "regularity" of $\vec{x}$. For instance, the filter $x_j \mapsto x_j - x_{j+1}$ that we could say to be associated to $(1, -1)$ gives small values except when there is a jump. To avoid problems with the indexes we assume that they warp around modulo N, then x_0 follows x_{N-1}. If we wish the filter to be more sensitive to the variations of $\vec{x}$ we could use $(1, -2, 1)$ that behaves as a second derivative (recall the Laplacian filter). We consider instead a longer filter $\vec{f} = (f_0, f_1, f_2, f_3) \in \mathbb{R}^4$ to have an extra degree of freedom. If it acts as a second derivative it must annihilate linear polynomials, $\sum_j (a + bj) f_j = 0$, hence

$$f_0 + f_1 + f_2 + f_3 = 0 \qquad \text{and} \qquad 0f_0 + 1f_1 + 2f_2 + 3f_3 = 0. \tag{3.82}$$

Let us impose a kind of independence of the filter with itself under 2-translations, meaning $(f_0, f_1) \cdot (f_2, f_3) = 0$, and finally we require the normalization condition $\|\vec{f}\| = 1$. Hence we have

$$f_0 f_2 + f_1 f_3 = 0 \qquad \text{and} \qquad f_0^2 + f_1^2 + f_2^2 + f_3^2 = 1. \tag{3.83}$$

This makes 4 equations with 4 unknowns. Before continuing, let us see briefly how this idea generalizes to a longer filter $(f_0, f_1, \dots, f_{2L-1}) \in \mathbb{R}^{2L}$. In this case, we get L equations instead of (3.82) if the filter acts as an L-th derivative:

$$\sum_{j=0}^{2L-1} j^l f_j = 0 \qquad \text{for} \quad 0 \le l < L. \tag{3.84}$$

The independence under 2-translations and the normalization condition are comprised in

$$\sum_{0 \le j, j+2l < 2L} f_j f_{j+2l} = 2\delta_l \qquad \text{for} \quad 0 \le l < L \tag{3.85}$$

where $\delta_0 = 1$ and $\delta_l = 0$ otherwise. In this way we have $2L$ equations with $2L$ unknowns and (3.82) and (3.83) correspond to take $L = 2$. Note the algebraic identity

$$\Big(\sum_{j=0}^{2L-1} (-1)^j f_j \Big)^2 + \Big(\sum_{j=0}^{2L-1} f_j \Big)^2 = 2 \sum_{j=0}^{2L-1} f_j^2 + 4 \sum_{l=0}^{L-1} \sum_{0 \le j, j+2l < 2L} f_j f_{j+2l} \tag{3.86}$$

that implies that the first square equals 2. As (3.84) and (3.85) are invariant by a global sign change, we can safely extract the positive square root and assume

$$\sum_{j=0}^{2L-1} (-1)^j f_j = \sqrt{2}. \tag{3.87}$$

Let us come back to the case $L = 2$. With this convention about the alternating sum (3.82) and (3.83) have a unique solution

$$f_0 = \frac{1 - \sqrt{3}}{4\sqrt{2}}, \qquad f_1 = \frac{\sqrt{3} - 3}{4\sqrt{2}}, \qquad f_2 = \frac{3 + \sqrt{3}}{4\sqrt{2}}, \qquad f_3 = -\frac{1 + \sqrt{3}}{4\sqrt{2}}. \tag{3.88}$$

Consider the following circulant $N/2 \times N$ matrix (each row is a circular 2-shift of the preceding)

$$F_N = \begin{pmatrix} f_0 & f_1 & f_2 & f_3 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & f_0 & f_1 & f_2 & f_3 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & f_2 & f_3 \\ f_2 & f_3 & 0 & 0 & 0 & 0 & 0 & \dots & f_0 & f_1 \end{pmatrix} \tag{3.89}$$

and the same matrix flipping $\vec{f}$ and putting alternating signs

$$G_N = \begin{pmatrix} -f_3 & f_2 & -f_1 & f_0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & -f_3 & f_2 & -f_1 & f_0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & -f_1 & f_0 \\ -f_1 & f_0 & 0 & 0 & 0 & 0 & 0 & \dots & -f_3 & f_2 \end{pmatrix}. \tag{3.90}$$

The relations (3.83) assure that the rows of F_N are orthonormal and the same applies to those of G_N, due to (3.83). Moreover any row of F_N is orthogonal to any row of G_N then

$$T_N = \begin{pmatrix} G_N \\ F_N \end{pmatrix} \quad \text{is an orthogonal matrix:} \quad T_N^{-1} = T_N^t. \tag{3.91}$$

Note that $F_N\vec{x}$ is the result of applying the filter $\vec{f}$ at even indexes. On the other hand $f_0 - f_1 + f_2 - f_3 = \sqrt{2}$ implies that $2^{-1/2}G_N\vec{x}$ is a kind of blur operator that averages nearby values. Forgetting about the scale, T_N decomposes the signal into $G_N\vec{x}$ that gives a rough approximation of $\vec{x}$ losing one half of the information about the signal and into $F_N\vec{x}$ that gives the details. If we want to enter into a coarser scale we can use $T_{N/2}$ to decompose $G_N\vec{x}$. Iterating, we establish a finite sequence of approximations and details $\vec{a}_j, \vec{d}_j \in \mathbb{C}^{N/2^j}$ given by

$$\begin{pmatrix} \vec{a}_{j+1} \\ \vec{d}_{j+1} \end{pmatrix} = T_{N/2^j}\vec{a}_j \quad \text{for } 0 \le j < K-1 \text{ with } \vec{a}_0 = \vec{x}. \tag{3.92}$$

If we apply this relation backwards using $T^{-1}_{N/2^j} = T^t_{N/2^j}$, we can easily recover $\vec{x} = \vec{a}_0$ from $\vec{a}_{K-1}, \vec{d}_{K-1}, \dots, \vec{d}_1$ where $\vec{a}_{K-1} \in \mathbb{C}^2$ is a kind of global average of the signal and $\vec{d}_j \in \mathbb{C}^{N/2^j}$ is the responsible of the details at level j. The linear endomorphism

$$W : \vec{x} \in \mathbb{C}^N \longmapsto \left(\vec{a}^t_{K-1}, \vec{d}^t_{K-1}, \dots, \vec{d}^t_1\right)^t \in \mathbb{C}^N \tag{3.93}$$

is called the D4 *wavelet transform* where D stands for Daubechies and 4 indicates the length of the filter[3]. Using (3.91) it can be deduced that it is an orthogonal map [RVF09, Prop.7.3]. In general if we use a filter $(f_0, f_1, \dots, f_{2L-1}) \in \mathbb{R}^{2L}$ with f_j the solution of (3.84) and (3.85) under (3.87) we get the D2L *wavelet transform*. In practice, W and its inverse are efficiently computed with (3.92). These transformations are generically know as *discrete wavelet transforms* and abbreviated as DWT.

If for a signal $\vec{x}$ we have

$$W(\vec{x}) = \sum_{j=1}^{N} \lambda_j \vec{e}_j \quad \text{with } \{\vec{e}_1, \dots, \vec{e}_N\} \text{ the canonical basis and } \lambda_j = \vec{e}_j \cdot W(\vec{x}), \tag{3.94}$$

then taking inverses we obtain

$$\vec{x} = \sum_{j=0}^{N-1} \lambda_j \vec{W}_j \qquad \text{where } \vec{W}_j = W^{-1}(\vec{e}_j) \text{ and } \lambda_j = \vec{W}_j \cdot \vec{x}. \tag{3.95}$$

[3] Some people use this name for a single step in (3.92).

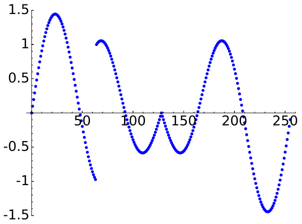
In this way $\vec{x}$ has been analyzed in terms of the "finite wavelets" $\vec{W}_j$ that embody information about different scales of $\vec{x}$. Surely you will think that the name wavelet is here merely a poetic license. We will see later than when $N \to \infty$ they become bona fide wavelets. Note that the theoretical framework to introduce the actual wavelets required a nontrivial amount of harmonic analysis while defining $\vec{W}_j$ only requires linear algebra.

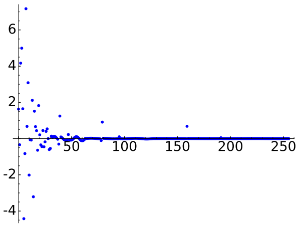
It turns out that the "harmonics" $\vec{W}_j$ in the decomposition (3.95) become more localized when j grows (this is not obvious now) and bypass the global nature of classical Fourier analysis. Recall that this aim was our motivation to introduce wavelets in the continuous setting.

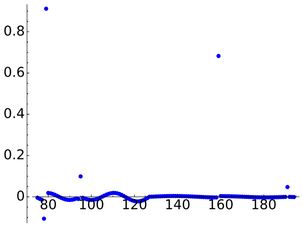
Let us illustrate the interest of the decomposition (3.95) with an example. For $N = 256$ and $\omega = 2\pi/N$ consider the signal

$$(3.96) \qquad x_j = \begin{cases} 0 & \text{if } N/4 < j \le N/2, \\ \sin(3\omega j) + \frac{1}{2}\sin(2\omega j) & \text{otherwise,} \end{cases} \qquad \text{for} \quad j = 0, 1, \dots, N-1.$$

It has a jump at $N/4 = 64$, as shown in the first figure.

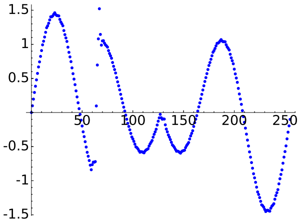


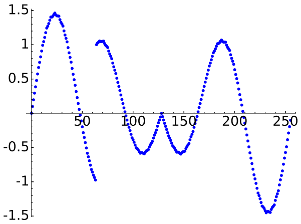


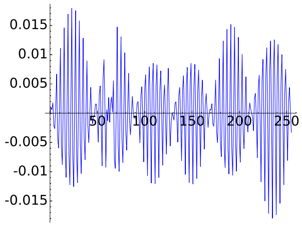


The second figure shows its D4 wavelet transform. It looks as a typical DFT of a regular function, with initial large values and an eventual decay[4]. The third figure is a scaled version of a part of the previous plot. The jump causes a higher value at the place 64 of $\vec{d}_1$ and then we expect a slightly singular point at the place $256 - 64 = 192$ of the wavelet transform. In fact, we observe also singular points at the places 79, 80, 96 and 160.

If we disregard three quarters of the information putting $\lambda_j = 0$ in (3.95) for $j \ge 64$ we still obtain a good reconstruction except for some points close to the jump. This reconstruction is represented in the first figure below.







[4] Perhaps this decay of the wavelet transform puzzles you if you do not buy the localization of $\vec{W}_j$. The idea is that $\vec{d}_1$ consists of increments between near values and then for smooth signals it should be small while higher $\vec{d}_j$ involve more separated values. The scaling plays a role. For example, the normalized derivative filter $2^{-1/2}(1,-1)$ would give elements like $2^{-1/2}(x_j - x_{j+1})$ in $\vec{d}_1$ and like $2^{-1/2}(x_j - x_{j+2})$ in $\vec{d}_2$. If the signal is smooth we expect $(x_j - x_{j+2})/(x_j - x_{j+1}) \approx 2$ away from critical points.

If we preserve the five λ_j corresponding to the aforementioned singular places, the reconstruction is almost perfect. The second figure shows its aspect and the third the error. The noticeable point here is that around one fourth of the wavelet coefficients are enough to faithfully recover the signal. In fact one can get a better rate. This is the gate to signal compression.

Recall that by the previous definitions of a multiresolution analysis (3.33) and (3.37), a wavelet ψ is expressed as an in principle infinite linear combination of the scaled translated scaling functions $\phi(2x-k)$. Let us say that this linear combination is finite:

$$\psi(x) = \sqrt{2}\sum_{j=0}^{2L-1} f_j\phi(2x-j) \qquad \text{with} \quad f_j \in \mathbb{R}. \tag{3.97}$$

The wavelet ψ and the scaling function ϕ are orthogonal by (3.36) with $j=0$ and with the notation of (3.34) this is equivalent to

$$0 = \frac{1}{2}\int_{-\infty}^{\infty}\psi(x)\bar{\phi}(x-m)\,dx = \sum_{j=0}^{2L-1}\sum_k f_j\bar{c}_k\int_{-\infty}^{\infty}\phi(2x-j)\bar{\phi}(2x-2m-k)\,dx = \sum_{j=0}^{2L-1} f_j\bar{c}_{j-2m}. \tag{3.98}$$

The most simple way to achieve this for any m is to put symmetric alternating signs choosing

$$c_j = (-1)^{j+1}f_{2L-1-j} \quad \text{for } 0 \le j < 2L \qquad \text{and} \qquad c_j = 0 \quad \text{otherwise}. \tag{3.99}$$

Then

$$\phi(x) = \sqrt{2}\sum_{j=0}^{2L-1}(-1)^{j+1}f_{2L-1-j}\phi(2x-j). \tag{3.100}$$

Our assumption of having a finite sum in (3.97) seems unmotivated but it is necessary if we want ϕ to have compact support because if ϕ is supported on $[-s,s]$ then $\phi(2x-j)$ with $|j|>3s$ cannot appear in the (3.100) expansion of ϕ because $\phi(x)$ and $\phi(2x-j)$ are orthogonal. Note also that if ϕ is compactly supported then ψ is compactly supported too thanks to (3.97).

Our conditions on the filters acquire a sound meaning in this continuous formulation: The orthonormality of $\{\phi(\cdot-k)\}_{k\in\mathbb{Z}}$ gives (3.85) and the normalization $\int_{-\infty}^{\infty}\phi = 1$, recall (3.60), gives (3.87). On the other hand, (3.84) comes from imposing $\int_{-\infty}^{\infty}x^l\psi = 0$ for $0 \le l < L$. The quickest way to see this is to consider the Fourier transform of (3.97) and take derivatives at the origin [RW98, §5.3.3].

One may suspect that there is a complete equivalence between constructing some MRAs with compactly supported wavelets and a choice of a finite filter with the properties (3.85), (3.87) and (3.84) at least for $l=0$. This is "almost" true. The glitch is that for a function $\int_{-\infty}^{\infty}\phi = 1$ satisfying (3.100), we have to put an extra condition to assure that $\{\phi(\cdot-k)\}_{k\in\mathbb{Z}}$ orthonormal is equivalent to (3.85). The reverse implication may fail but it does not in "generic" cases. The topic is treated broadly in [Dau92, Ch.6] and in [Dau92, Th.6.3.6] there is a summary of minimal conditions. The first condition in (3.81) implies them.

The requirement $\int_{-\infty}^{\infty} x^l \psi = 0$ for $0 \le l < L$ assures some regularity for the scaling function and hence for the wavelet via (3.97). Let us see a crude result of this kind [Dau92, §7.1.1].

Proposition 3.2.1. *Let ϕ with $\int_{-\infty}^{\infty} \phi = 1$ be the scaling function of an MRA satisfying* (3.100)*. If* (3.84) *holds, or equivalently $\int_{-\infty}^{\infty} x^l \psi = 0$ for $0 \le l < L$, then $\phi \in C^\alpha$ for*

$$0 < \alpha < L - 1 - \log_2 \max |f| \qquad \textit{with} \quad f(x) = \frac{\sum_{j=0}^{2L-1} (-1)^{j+1} f_{2L-1-j} e(-jx)}{\sqrt{2}\cos^L(\pi x)}. \tag{3.101}$$

Note that (3.84) implies that the numerator of f has a zero of order L at $x = 1/2$ then f has removable singularities, in fact it defines an entire function.

Proof. With the notation of (3.34) we have $c_j = (-1)^{j+1} f_{2L-1-j}$ and (3.80) gives

$$\widehat{\phi}(\xi) = \prod_{j=1}^{\infty} \cos^L\Big(\frac{\pi\xi}{2^j}\Big) \prod_{j=1}^{\infty} f\Big(\frac{\xi}{2^j}\Big) = (\operatorname{sinc} \xi)^L \prod_{j=1}^{\infty} f\Big(\frac{\xi}{2^j}\Big). \tag{3.102}$$

The second equality comes from $\cos x = \sin(2x)/(2\sin x)$ getting a well known telescoping product [Pin02, §6.5.3].

We have $f(0) = 1$ because $\int_{-\infty}^{\infty} \phi = 1$ implies (3.87) hence $|f(x) - 1|/|x|$ is bounded. If $2^{N-1} \le |\xi| < 2^N$ with $N \in \mathbb{Z}^+$ the general theory of infinite products [Ahl78, §4.1] gives

$$|\widehat{\phi}(\xi)| \le C|\xi|^{-L} \prod_{j=1}^{N} f\Big(\frac{\xi}{2^j}\Big) \le C|\xi|^{-L} \max |f|^N. \tag{3.103}$$

If the range (3.101) is not empty for any α in it there exists another $\alpha' > \alpha$ and we get

$$\max |f|^N < 2^{N(L-1-\alpha')} \le (2|\xi|)^{L-1-\alpha'}. \tag{3.104}$$

The conclusion is that $\widehat{\phi}(\xi)$ decays at least as $|\xi|^{-\alpha'-1}$ and it implies by Fourier inversion $\phi \in C^\alpha$ (cf. [Gra08, §3.2.2]). □

For $L = 2$ and (3.88) after some calculations [Wal08, §3.2.2] one gets the closed formula

$$f(x) = e(-3x/2)\big(\cos(\pi x) + i\sqrt{3}\sin(\pi x)\big) \tag{3.105}$$

which implies $\max |f| = \sqrt{3}$. Then the result proves that the underlying wavelet and the scaling function are in any C^α with $\alpha < 1 - \log_2 \sqrt{3} = 0.2075\dots$ while in reality α can be pushed at most to something close to 0.55 [Dau92, §7.2] then they are fractal-like.

After the previous considerations, if we overlook some technical point, the filters we have considered appear as coefficients in the expansion of compactly supported wavelets but it is unclear the relation to the vectors $\vec{W}_j$. The surprise is that when the dimension is very large, plotting the coordinates of $\vec{W}_j$ to the right scale we get the approximation of the wavelet and the scaling function. The explanation relies on the following *cascade algorithm* that works under a minimal regularity and constitutes the usual way of approximating scaling functions and wavelets.

Proposition 3.2.2. *If $\phi \in C_0^\alpha$, $0 < \alpha < 1$, with $\int_{-\infty}^{\infty} \phi = 1$ is the scaling function of an MRA that satisfies* (3.100) *then the sequence of step functions $\{\phi_n\}_{n=0}^{\infty}$ given by*

$$\phi_{n+1}(x) = \sqrt{2} \sum_{j=0}^{2L-1} (-1)^{j+1} f_{2L-1-j} \phi_n(2x - j) \qquad \text{with} \quad \phi_0 = \chi_{[-1/2,1/2)} \tag{3.106}$$

converges uniformly to ϕ. In fact $2^{\alpha n}\|\phi - \phi_n\|_\infty$ is bounded.

The support of ϕ_n is contained in the interval $\big[-2^{-n-1}, 2^{-n-1} + (2L-1)(1-2^{-n})\big]$, as a simple inductive argument shows. Then letting $n \to \infty$ one gets that the support of the scaling function is that of the filter.

Corollary 3.2.3. *If ϕ is as before, its support is contained in* $[0, 2L-1]$.

Proof of Proposition 3.2.2. After iterating j times we obtain a formula for ϕ_j in terms of ϕ_0. Evaluating it at $k/2^j$ it follows that $\phi_j(k/2^j)$ is the coefficient of $\phi_0(2^j x - k)$. By (3.100), the formula also holds replacing ϕ_0 and ϕ_j by ϕ and then we can get this coefficient also with a scalar product. This gives the identity

$$\phi_j\Big(\frac{k}{2^j}\Big) = 2^j \int_{-\infty}^{\infty} \phi(x)\bar{\phi}(2^j x - k)\, dx = \int_{-\infty}^{\infty} \phi\Big(\frac{x+k}{2^j}\Big)\bar{\phi}(x)\, dx. \tag{3.107}$$

It implies

$$\phi\Big(\frac{k}{2^j}\Big) - \phi_j\Big(\frac{k}{2^j}\Big) = \int_{-\infty}^{\infty} \Big(\phi\Big(\frac{k}{2^j}\Big) - \phi\Big(\frac{x+k}{2^j}\Big)\Big)\bar{\phi}(x)\, dx. \tag{3.108}$$

As $\phi \in C_0^\alpha$ we have on each interval $|x - k/2^j| \le 2^{-j-1}$

$$\Big|\phi(x) - \phi_j\Big(\frac{k}{2^j}\Big)\Big| \le \Big|\phi(x) - \phi\Big(\frac{k}{2^j}\Big)\Big| + \Big|\phi\Big(\frac{k}{2^j}\Big) - \phi_j\Big(\frac{k}{2^j}\Big)\Big| \le C 2^{-j\alpha} \tag{3.109}$$

with a constant C not depending on the interval by the compactness of the support. □

By (3.97), to approximate the wavelet it is enough to consider

$$\psi_{n+1}(x) = \sqrt{2} \sum_{j=0}^{2L-1} f_j \phi_n(2x - j). \tag{3.110}$$

Define $\vec{A}_j(x)$ as a vector function with m-th coordinate $\phi_j(2^{-j}x - m)$, $0 \le m < N/2^j$ and $\vec{D}_j(x)$ with m-th coordinate $\psi_j(2^{-j}x - m)$, $0 \le m < N/2^j$. We can read (3.106) and (3.110) saying that (3.92) holds changing $\vec{a}_j$ and $\vec{d}_j$ by $\vec{A}_j(x)$ and $\vec{D}_j(x)$.

Note that $\vec{A}_0(x)$ evaluated at m gives $\vec{e}_m$. Then for instance in the case $L = 2$ corresponding to D4 the discrete wavelet transform (3.93) satisfies

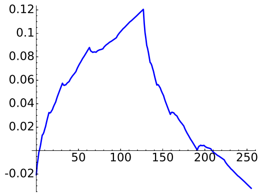
$$W(\vec{e}_m) = \Big(\phi_{K-1}(2^{1-K}m), \phi_{K-1}(2^{1-K}m - 1), \psi_{K-1}(2^{1-K}m), \psi_{K-1}(2^{1-K}m - 1), \dots\Big)^t. \tag{3.111}$$

This implies

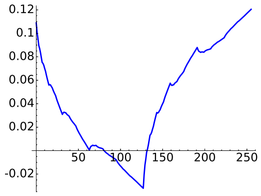
$$\phi_{K-1}(2^{1-K}m) = \vec{e}_1 \cdot W(\vec{e}_m) = W^{-1}(\vec{e}_1) \cdot \vec{e}_m. \tag{3.112}$$

Then the coordinates of $\vec{W}_1$ are approximations of the scaling function. The same holds with $\vec{W}_2$ except for a translation and the rest of the $\vec{W}_j$ give approximations to translated scaled versions of the wavelet, being less precise as j grows. The same happens with $L > 2$ but in this case the $\vec{a}_j$ appearing at the beginning of the D2L wavelet transform has more coordinates and then we obtain more that 2 shifted copies of the scaling functions before getting the wavelet.

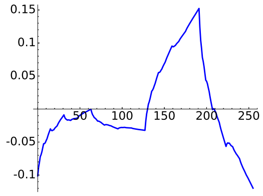
To illustrate this ideas, in the following figures there are plots of $\vec{W}_j$ for $j = 1, 2, 3, 10$ corresponding to $N = 256$ and D4. For $j = 1$ and 2 we get shifted approximations of the scaling function and for $j = 3$ an approximation of the wavelet. The last figure for $j = 10$ is a scaled version of the wavelet that gives a best idea of the compact support because the coordinates have not time to wrap around.



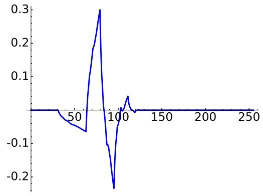
$\vec{W}_1$ for D4



$\vec{W}_2$ for D4

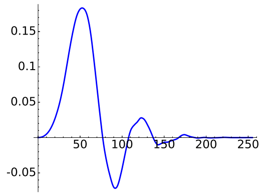


$\vec{W}_3$ for D4

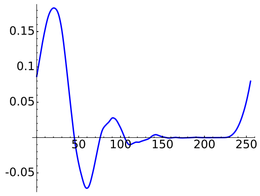


$\vec{W}_{10}$ for D4

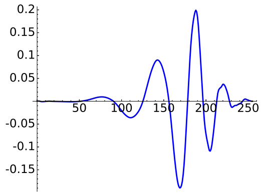
For D12 we get shifted approximations of the scaling function for $1 \leq j \leq 8$ and the wavelet shows up for $j > 8$. This is because 8 is the least power of 2 less or equal than L.



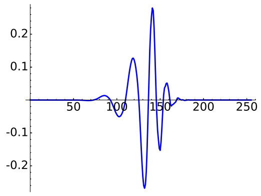
$\vec{W}_1$ for D12



$\vec{W}_8$ for D12



$\vec{W}_9$ for D12



$\vec{W}_{20}$ for D12

These plots confirm that more vanishing moments (greater L) implies more regularity. It can be proved that the continuous wavelet approximated with D12 is in C^2 and, as it was mentioned before, for D4 it is in a space slightly better than $C^{1/2}$ then it is not even differentiable. The case D6 is quite curious because detailed plots of the wavelet with the cascade algorithm seem to show that it is no differentiable but it can be proved that it is in a space only slightly better than $C^{1.08}$. No wonder it is so difficult to get intuition from the numerical results. For more on the regularity see [DL92].

Suggested Readings. The approach in terms of filters in this subsection is loosely based on [Wal08] and the few paragraphs in [Ger99]. The overwhelmed reader is addressed to these references. On the other hand, a very good source theoretically oriented is [Dau92]. If you are interested in the history of wavelets, the book [HW06] collects a number of foundational papers and you may find fruitful the introduction of [Dau88].

3.2.2 Some examples and applications

Wavelets have had an important impact in many areas due to the possibility of easily implement the discrete wavelet transform or the cascade algorithm in computer software. This has shadowed the, perhaps exaggerated, initial fuss over the possible relevance of wavelet theory in problems of abstract harmonic analysis. When a mathematical topic extends beyond the mathematical books it becomes more than doubly interesting because mathematics constitutes less than one half of the scientific world.

We consider here some applications of the discrete wavelet transform to signal processing. The first one is to compress finite signals. Essentially we are going to transform a signal into another keeping almost the same information with many zero values. Without entering now into considerations about how the compression is implemented, we play around with the rough common sense idea that storing a vector is more economical if the most of its coordinates are zero. A modern Cantorian mathematician knows that this is a theoretical lie because we can reduce N real numbers to only one with the same information for instance alternating their digits. In fact since our undergraduate times we know how to do it with (countable) infinitely many! But computers and software format standards are ultra-Kroneckerian, the real numbers in computers are not real (pun intended) but integers of limited size. The quantization enters into the game and the theory around compression becomes interesting enough to be an important part of a big topic called information theory. We postpone it to a future section. Here we take for granted that many zeros imply a good compression rate. If you need a justification now, skip to §4.2.

We have seen the main idea in a previous example: setting many values of the DWT to zero has not a noticeable effect on the signal even if it is not regular everywhere. Now we are going to elaborate more this idea designing an algorithm to do it automatically. All the time we focus on lossy compression. We make zeros with a negligible loss in the reconstruction and we can tune the meaning of "negligible loss" but not to reduce it to null.

Given a finite real signal $\vec{x} = \{x_n\}_{n=0}^{N-1}$ with N a power of 2, we transform it with

$$\vec{x} \longmapsto \mathcal{T}_\lambda(\mathrm{DWT}(\vec{x})) \tag{3.113}$$

where DWT is a discrete wavelet transform and $\mathcal{T}_\lambda$ is the thresholding function that replaces each coordinate of absolute value less than λ by zero and leaves the rest invariant. This is called in the jargon *hard thresholding* while *soft thresholding* employs more regular functions. With a good choice of λ, (3.113) is likely to give a vector with many zero coordinates and by common sense there should exist methods to lossless compress it (that we shall study in a future section). To recover the signal one uncompresses the result to get $\vec{u}$, the right hand side of (3.113), and obtains an approximation of the initial signal with

$$\vec{u} \longmapsto \text{inverse } \mathrm{DWT}(\vec{u}). \tag{3.114}$$

The million-dollar question is how to choose λ. This *thresholding* is a topic by itself (see [Mal09, §11.3-4]). Here we will see a method based on the energy, the fancy physical name for the square norm of a vector. Recall or learn that in physics the energy depends on the square of the amplitude then in (3.95) we define the energy of $\lambda_j\vec{W}_j$ as $|\lambda_j|^2$ and the total energy of the signal is $\sum|\lambda_j|^2$. We know that W is an orthogonal map then this energy and the square norm of the signal coincide. This conservation of the energy also holds with a normalization factor for the DFT, as we proved in (2.55). For the DCT it is also true with the normalization factor $\sqrt{2/N}$ except for the first coordinate that must be multiplied by $1/\sqrt{N}$.

When we select a nontrivial threshold λ we lose energy in the reconstruction. Let us say that α is the proportion of the energy that we allow to leak. Consider the energy spectrum $E_0 \ge E_1 \ge \cdots \ge E_{N-1}$ composed by the ordered energies of the elemental waves $\lambda_j\vec{W}_j$. In plain words, the ordered squared coordinates of the DWT. Then we can disregard the energies E_j with $j > j_0$ if they add less than α times the total energy. It leads to the threshold formula

$$\lambda = \sqrt{E_{j_0}} \qquad \text{with} \quad j_0 = \min\Big\{ j \,:\, \alpha\sum_{n=0}^{N-1} E_n > \sum_{n=j+1}^{N-1} E_n \Big\}. \tag{3.115}$$

Note that $\sqrt{E_j}$ are the absolute values of the coordinates of $\text{DWT}(\vec{z})$ and then if α is very small we are forced to choose $j_0 = N-1$, because an empty sum is defined as 0, λ is the smallest absolute value and $\mathcal{T}_\lambda$ is the identity, no new zeros are introduced. In general the number of zero coordinates in $\mathcal{T}_\lambda\big(\text{DWT}(\vec{z})\big)$ is $N-1-j_0$. One can argue that (3.115) is useless because replaces our freedom and so our doubts about how to chose λ by how to choose α. The point is that there is no prior intuition in λ while α is meaningful because it gives the relative error, with the ℓ^2 norm, of the approximation in the reconstruction. Namely,

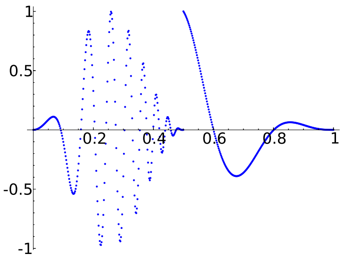
$$\|\vec{x} - \text{recons}(\vec{x})\| \approx \|\vec{x}\|\sqrt{\alpha}. \tag{3.116}$$

We do not have an exact equality because the energy spectrum is discrete.

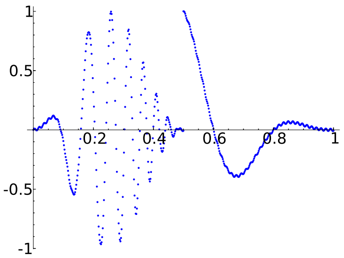
Let us see with detail the underlying ideas and the numerical results for the finite signals $\vec{x} = \{x_n\}_{n=0}^{N-1}$ and $\vec{z} = \{z_n\}_{n=0}^{N-1}$ with $N = 512$ given by the following formulas where $t = n/N$ is the "time":

$$x_n = \begin{cases} \sin^2(2\pi t)\cos(60\pi t^2) & \text{if } t < 1/2, \\ 4(1-t)^2\sin(5\pi t) & \text{if } t \ge 1/2 \end{cases} \qquad \text{and} \qquad z_n = x_n + 0.01\sin(128\pi t). \tag{3.117}$$

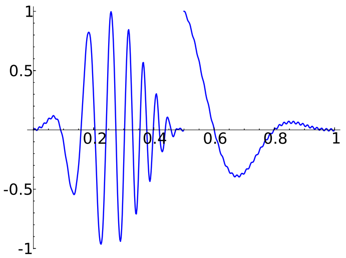
Their aspect is shown here with the horizontal axis scaled by $1/N$ i.e., in terms of t. The last figure connecting the dots may seem more informative.



Point plot of $\vec{x}$



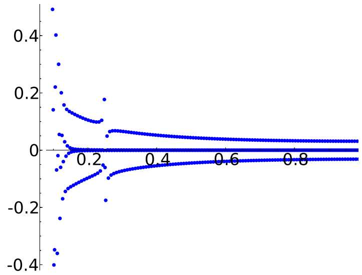
Point plot of $\vec{z}$



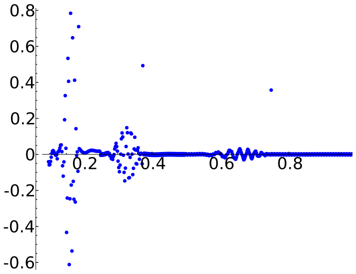
Joint plot of $\vec{z}$

We infer that $\vec{x}$ contains a wave of increasing frequency in its first half and a smooth function after a discontinuity. If you do not distinguish $\vec{z}$ and $\vec{x}$ from the plot note that it is plain with the formulas to deduce that the former is a periodic perturbation of the latter with frequency $64\,Hz$. In the last figure the line is shaky.

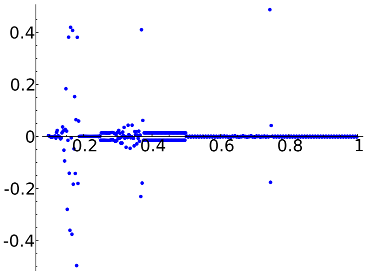
Let us consider the plots of the DCT of $\vec{z}$ and of the DWTs of $\vec{z}$ corresponding to D4 and D8. Again for presentation the horizontal axis is rescaled in such a way that N becomes 1. This axis is restricted to values greater than 0.1 to avoid the first high values to mask the rest. Finally the DCT is represented multiplied by $\sqrt{2/N}$ with respect to the definition in Proposition 2.2.2 to be consequent with the energy conservation.



DCT of z



D4 DWT of z

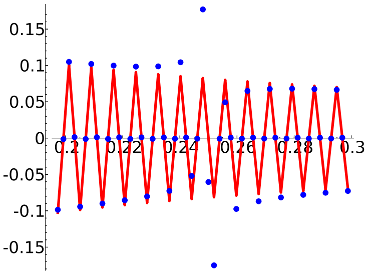


D8 DWT of z

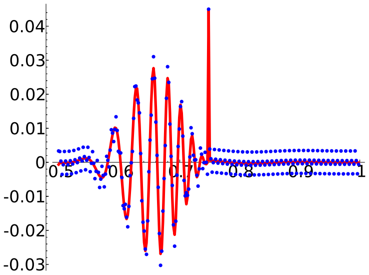
Let us spend some time exploring some aspects of these plots. The symmetric peak of the DCT around 0.25 corresponds to one quarter of the Nyquist frequency $256\,Hz$, the maximal observable frequency. It is caused by the perturbation, $128\pi = 256/4 \cdot 2\pi$. Reducing this peak in an adequate manner would switch off the perturbation as we did in an example of periodic noise in §2.2.4. The whole DCT feels the discontinuity of $\vec{z}$ and it does not decay very quickly. Instead of showing a slow decaying oscillation, the two discrete wavelet transforms are mostly a collection of peaks more or less in the same places and very small values. The last peak appears at 0.75. As the first detail vector $\vec{d}_1$ occupies the second half of the DWT in (3.93), this means that the peak reflects something that is happening in the middle of $\vec{z}$. It is the discontinuity. In the setting of $L^2(\mathbb{R})$ orthonormal wavelets, the limit of the cascade algorithm, we would need to employ very localized wavelets in its representation. Of course we also need other wavelets of smaller frequencies, this gives some replicas of the peak and in fact with a little of faith and a little of experimentation playing with N one can imagine the vestiges of an underlying self-similarity (D4 is a better support for this faith). Coming back to reality, the perturbation is trigonometrical, not localized, and of high frequency, then we need many translations of the most localized wavelets to take care of it. This suggests that both DWTs finish with a tiny positive more or less constant amplitude.

According to the previous explanations, let us try to predict the aspect of the same figures for $\vec{x}$. In this case the symmetric peak of the DCT near $1/4$ must disappear and the slow decay due to the discontinuity should stay. On the other hand for the DWTs the peak near 0.75 should be preserved and the tiny amplitude of the last values should reduce to almost nothing.

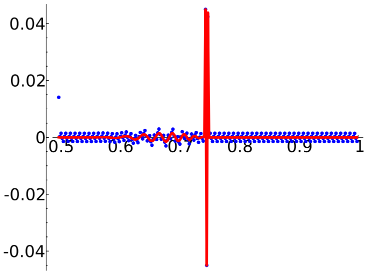
Our predictions are confirmed in the following plots where the points corresponding to $\vec{x}$ appear joined and to better appreciate the scale of the last oscillations, the vertical axis in the DWTs has been cropped to $[-0.045, 0.045]$, the peaks are not fully represented[5]. The horizontal axis is restricted to $[1/2, 1]$ and to $[0.2, 0.3]$ in the case of the DCT.



DCT of $\vec{x}$ and $\vec{z}$



D4 DWT of $\vec{x}$ and $\vec{z}$



D8 DWT of $\vec{x}$ and $\vec{z}$

After this discussion for $\vec{x}$ it seems very likely that for reasonable values of α we get more zeros with $\mathcal{T}_\lambda(\mathrm{DWT}(\vec{x}))$ than with $\mathcal{T}_\lambda(\mathrm{DCT}(\vec{x}))$ where λ in each case is computed with (3.115). It also sounds natural to expect a better performance with D8 than with D4 because the signal is mostly regular.

The case of $\vec{z}$ requires deeper considerations. The relative energy of the perturbation is $\|\vec{z} - \vec{x}\|^2/\|\vec{x}\|^2 \approx 3.4 \cdot 10^{-4}$ then for α bigger than this quantity we cannot see the perturbation. It does not mean that the approximation is bad. The following table shows for $\alpha = 4 \cdot 10^{-4}$ an ℓ^∞ error, the absolute error, around 0.02 with the DWT. This is a 2% of the amplitude of the signal, difficult to observe in the graphs.

$\boxed{\vec{x}}$	DCT	DWT4	DWT8
# zeros	251	412	449
threshold	3.19e-02	3.43e-02	6.05e-02
ℓ^∞ error	6.57e-02	2.59e-02	3.73e-02
ℓ^2 error	1.73e-01	1.74e-01	1.72e-01

$\boxed{\vec{z}}$	DCT	DWT4	DWT8
# zeros	250	395	437
threshold	3.19e-02	2.55e-02	3.58e-02
ℓ^∞ error	6.56e-02	2.05e-02	2.18e-02
ℓ^2 error	1.73e-01	1.74e-01	1.71e-01

Note that for both signals and for every considered transform the ℓ^2 error is almost constant and very close to $\|\vec{x}\|\sqrt{\alpha} \approx 0.17\dots$ as dictated by (3.116).

Regarding compression the important thing to notice is the amount of zeros. The figures for the DWT with D4 and D8, denoted DWT4 and DWT8, are much higher than those for the DCT, as expected. Note for instance that for $\vec{x}$ using DWT8 we only need $512 - 449 = 63$ nonzero values to recover the signal with this precision, more of the 87% of the coordinates are set to 0 and the result is difficult to distinguish from the original with the naked eye. Recall that when we studied JPEG we also kept the global aspect of an image putting a lot of zeros. Actually in both cases we are using the same idea with different harmonics.

If we take α a small fraction of the relative energy of the perturbation then we will be able to see it. Actually the experiments show that a fraction like $1/4$ is enough. Choosing $\alpha = 10^{-4}$, which corresponds to $1/2$ of the previous ℓ^2 error, we get the following table.

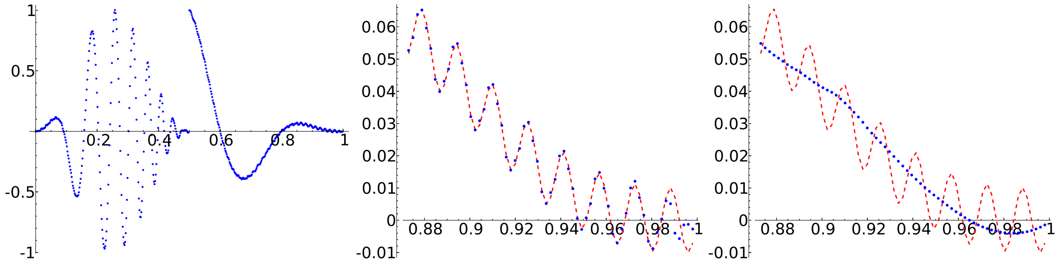
[5] I am afraid you should take my word about the good approximation of these peaks. If not, write your own program.

The absolute error also decreases but we pay price with a reduction on the number of zeros and then in the compression.

$\boxed{\vec{x}}$	DCT	DWT4	DWT8
# zeros	228	370	434
threshold	3.14e-02	1.67e-02	2.27e-02
ℓ^∞ error	2.06e-02	1.36e-02	1.49e-02
ℓ^2 error	8.46e-02	8.66e-02	8.63e-02

$\boxed{\vec{z}}$	DCT	DWT4	DWT8
# zeros	227	336	360
threshold	3.14e-02	1.43e-02	1.41e-02
ℓ^∞ error	2.05e-02	1.16e-02	1.11e-02
ℓ^2 error	8.46e-02	8.70e-02	8.69e-02

Visually the reconstructed plot using (3.114) is indistinguishable from the original signal. Note the comparison of the cases $\alpha = 4 \cdot 10^{-4}$ and $\alpha = 10^{-4}$ in a zoom showing the last 64 values with the dashed line indicating the graph of the function defining $\vec{z}$.



$\alpha = 10^{-4}$ D8 reconstruction   Detail $\alpha = 4 \cdot 10^{-4}$ D8   Detail $\alpha = 10^{-4}$ D8

Are the DWTs better than the DCT to compress $\vec{z}$ for any α? The answer is no. The perturbation is almost a pure tone for the DCT and requires a simple peak to encode it but for the DWT, it is a collection of small bumps that has to reproduce with highly localized wavelets. Then for α giving a threshold λ so small that is less than the small finishing values of the DWTs we can force the number of zeros to be greater with the DCT. This is achieved for instance at $\alpha = 10^{-6}$ that gives the following table:

$\boxed{\vec{x}}$	DCT	DWT4	DWT8
# zeros	214	247	382
threshold	3.95e-03	1.71e-03	2.07e-03
ℓ^∞ error	3.53e-03	1.35e-03	1.09e-03
ℓ^2 error	8.52e-03	8.60e-03	8.63e-03

$\boxed{\vec{z}}$	DCT	DWT4	DWT8
# zeros	213	104	187
threshold	3.95e-03	2.08e-03	1.44e-03
ℓ^∞ error	3.48e-03	2.37e-03	1.07e-03
ℓ^2 error	8.48e-03	8.49e-03	8.65e-03

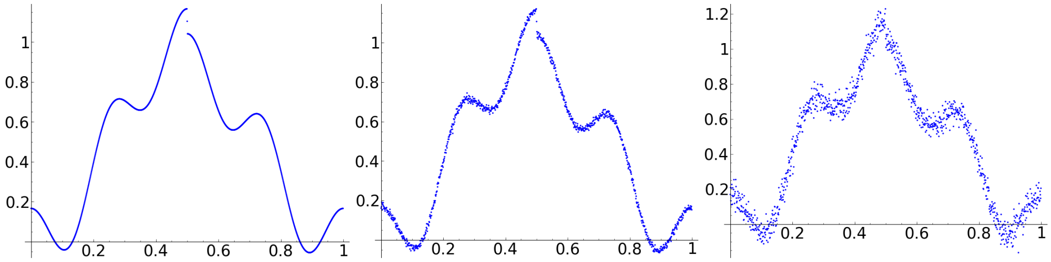
Note that for $\vec{x}$ the DWTs are still better. Even at this level the proportion of zeros using DWT8 is quite noticeable, almost 3/4 of the coordinates are zero.

A topic closely linked to compression is noise reduction. In the previous example if we would have considered the perturbation as noise we could have cleared $\vec{z}$ manipulating the DCT or using (3.113) and (3.114) with α greater than $\|\vec{z} - \vec{x}\|^2/\|\vec{x}\|^2$.

In the common language and in signal processing usually one considers the *noise* as something chaotic. The most interesting kind of noise is the Gaussian noise that we already considered in the context of image processing. For a finite discrete signal it corresponds to take a sample of a distribution $N(0, \sigma)$ and sum each value to each coordinate. It appears as a cloud of random points around the graph of the signal. For instance, consider $\vec{x} = \{x_n\}_{n=0}^{N-1}$ with $N = 1024$ given by the following formula where $t = n/N$:

$$x_n = \frac{15 - \operatorname{sgn}(t-1/2)}{16}\sin^2(\pi t) + \frac{1}{6}\cos(8\pi t). \tag{3.118}$$

The aspect of the Gaussian noise in this signal, using t in the horizontal axis, is:



Signal $\vec{x}$ Noisy $\sigma = 0.01$ Noisy $\sigma = 0.04$

In the last case the noise masks the jump discontinuity.

If the signal represents sound and at certain point there is a long enough silence it is not difficult to figure out what is σ but in the general situation we do not know it. Once we estimate it we proceed as in the compression setting using (3.113) and (3.114) and it leads to choose λ capable of eliminating the contribution of the noise.

To complete this program we follow [DJ94] (see also [Mal09, §11.3]). The starting idea is that almost all of the last $N/2$ coordinates of (3.92) are pure noise because for a typical signal they are very small except for the presence of discontinuities. Orthogonal maps transform Gaussian distributed vectors into vectors with the same distribution then we expect the coordinates of $\vec{d}_1$ to be a sample of a distribution $N(0,\sigma)$ except for some outliers corresponding to discontinuities. Using the sample standard deviation as an estimator for σ is not a good idea due to these possible outliers. A more robust estimator is the *median absolute deviation* defined as

(3.119)
$$\operatorname{MAD}\big(\{v_n\}_{n=0}^{N-1}\big) = \operatorname{median}\big(|v_0 - m|, \dots, |v_{N-1} - m|\big) \quad \text{with} \quad m = \operatorname{median}(v_0, \dots, v_{N-1}).$$

It can be proved that if $\{v_n\}_{n=0}^{N-1}$ is a sample of a distribution $N(0,\sigma)$ then one has the convergence in probability

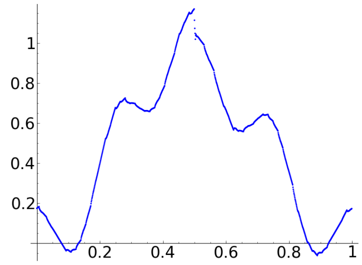
$$\operatorname{MAD}\big(\{v_n\}_{n=0}^{N-1}\big) \to K_0\sigma \qquad \text{with} \quad \int_0^{K_0} e^{-t^2/2}\,dt = \sqrt{\frac{\pi}{8}}. \tag{3.120}$$

The value of K_0 is usually approximated by 0.6745. On the other hand, without entering into details, in [DJ94] it is shown that under certain conditions the "best threshold" to eliminate the noise of the detail vectors $\vec{d}_j$ goes like $\sigma\sqrt{2\log N}$ when $N \to \infty$. This leads to take

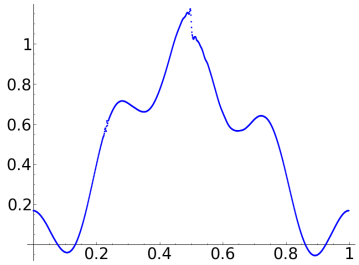
$$\lambda = \frac{\operatorname{MAD}(\vec{d}_1)}{K_0}\sqrt{2\log N} \approx 2.0967\,\operatorname{MAD}(\vec{d}_1)\sqrt{\log N}. \tag{3.121}$$

There is a little variation in the application of λ with respect to compression. Now in (3.113) we must use $\mathcal{T}_\lambda$ only on the detail vectors $\vec{d}_j$. For instance for D4 this means that we do not apply it on the two first coordinates and for D12 that we leave the first 8 coordinates unchanged irrespectively their value. The reason is clear, as the noise has zero mean its should not affect the approximation vectors.

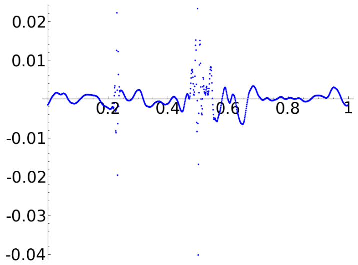
For the previous signal, applying (3.121) in the case $\sigma = 0.01$ for D4 and D12 one obtains the following results:



Denoising D4, $\sigma = 0.01$



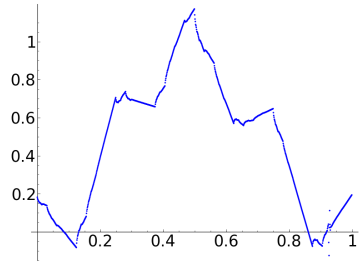
Denoising D12, $\sigma = 0.01$



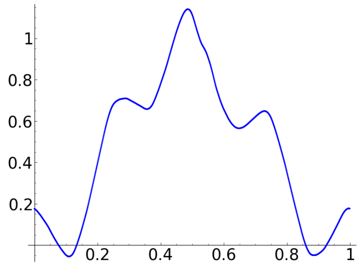
Error D12, $\sigma = 0.01$

The error is naturally greater near the discontinuity. The peak near 0.25 is just bad luck. By chance the noise reached there a high value in my experiment, it disappeared choosing another $N(0, 0.01)$ sample but I prefer to leave it to illustrate that this may happen in practice.

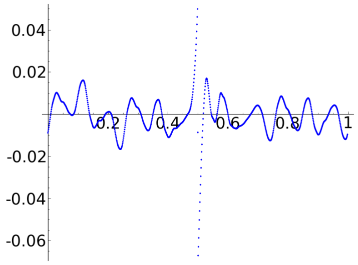
In the most challenging case $\sigma = 0.04$ the results are:



Denoising D4, $\sigma = 0.04$



Denoising D12, $\sigma = 0.04$



Error D12, $\sigma = 0.04$

The plot corresponding to D4 shows a lack of regularity but with D12 the result is quite good. A grumbler reader can argue that we had lost completely the jump but it is inevitable because we had already lost it in the noisy signal.

One can give a kind of phenomenological explanation for the lack of the regularity using D4 when the noise is large. In this case the threshold must be also large and then we are putting many $\vec{d}_j$ to $\vec{0}$. In the limit, if we take all of them to zero we have an approximation to a linear combination of the scaling functions by the cascade algorithm and these scaling functions are not regular.

To give an idea about the precision of employing the limit in (3.120) as an equality, in the $\sigma = 0.04$ example $\mathrm{MAD}(\vec{d}_1)/K_0$ gives approximately 0.04286 using D4 and 0.04042 with D12. An even better approximation is got for $\sigma = 0.01$.

Let us consider now discrete wavelet transforms in image processing. One of most celebrated applications is the format JPEG2000 but from the practical and theoretical point of view it is a bittersweet application. It has had a very limited success, the usual navigators do not support it, it involves a lot of technical details and moreover employs nonorthogonal wavelets (curiously called biorthogonal) that we have not covered here. The main ideas in this format are roughly as those we have seen before in the signal compression extended to two dimensions (see [Wal08] for more information, not a full description though).

Here we restrict ourselves to explain some points about the change of dimension and to give an academic example related to the use of finite wavelets with images.

As in previous subsections we consider a grayscale image as the matrix formed by the gray levels of the pixels. For simplicity we assume that A is an $N \times N$ matrix with N a power of 2. If we understand the image as a collection of columns, with $T_N A$ we apply to them the first step in (3.92). To do the same to the rows we can transpose, multiply by T_N and transpose again the result. Then the natural analogue of the first step in (3.92) is

$$A \mapsto T_N A T_n^t = \left(\begin{array}{c|c} G_N A G_N^t & G_N A F_N^t \\ \hline F_N A G_N^t & F_N A F_N^t \end{array}\right). \tag{3.122}$$

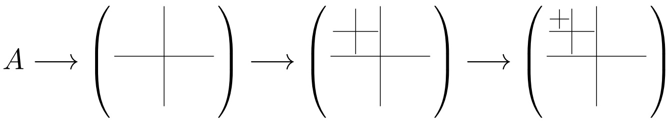
Recall that $G_N\vec{x}$ gives a kind of blur approximation to $\vec{x}$ and $\vec{F}_N\vec{x}$ a kind of derivative, the details. Then the upper left block gives a scaled approximation to A while the upper right block detects the vertical details, the difference between rows. Similarly $F_N A G_N^t$ embodies horizontal details and $F_N A F_N^t$ diagonal differences.

If we consider as details everything but the the first block, $A_1 = G_N A G_N^t$, the second step in (3.92) only should apply to it, replacing A_1 by $T_{N/2} A_1 T_{N/2}^t$. In general, if A_k is the upper left quarter of A_{k-1} the algorithm is

$$A_k \longrightarrow T_{N/2^k} A_k T_{N/2^k}^t \qquad \text{with} \quad A_0 = A, \tag{3.123}$$

where the arrow means that we have to replace A_k by the indicated expression in the $k+1$ step. In a scheme, we keep all the matrix elements from the current step except the upper left corner:

(3.124)



After applying k steps we obtain a matrix $W_k(A)$ that we call the k-iterated wavelet transform. In the case of D4 we can reach W_{K-1} when $N = 2^K$ which is a two dimensional analogue of the D4 wavelet transform. Note that it is not the same as applying W of (3.93) to the columns and to the rows of the result because we keep elements that involve some $G_N/2^j$.

The matrices $T_{N/2^k}$ are orthogonal then (3.123) can be inverted with

$$A_k \longrightarrow T_{N/2^k}^t A_k T_{N/2^k}. \tag{3.125}$$

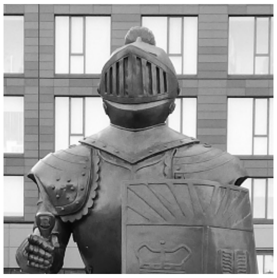
If we define the scalar product of real matrices in the usual way as $\langle A, B\rangle = \operatorname{Trace}(AB)$ then $A \mapsto W_k(A)$ becomes a linear orthogonal operator for each k.

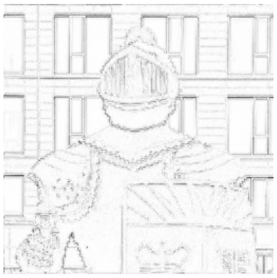
After these general considerations, as an application we are going to use $W_k(A)$ to construct a simple edge detector. Let Z_k be the operator that sets the first $N/2^k \times N/2^k$ block to zero and consider

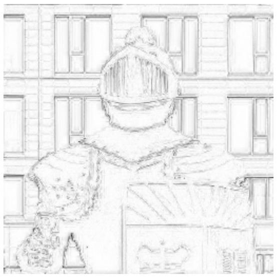
$$A \longmapsto (W_k^{-1} \circ Z_k \circ W_k)(A). \tag{3.126}$$

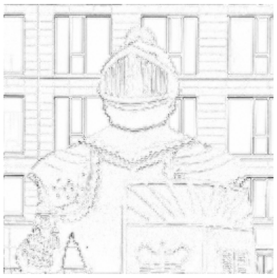
It acts as an edge detector. The explanation reduces to say that with Z_k we only leave the part of $W_k(A)$ corresponding to the details. The value of k is up to our will. The most reasonable is to take it small because otherwise we act on a coarse scale.

Here it is an example with a 512×512 image for D4 and D6.


Original

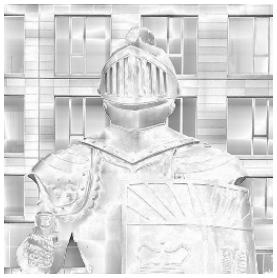

D4 with $k = 2$

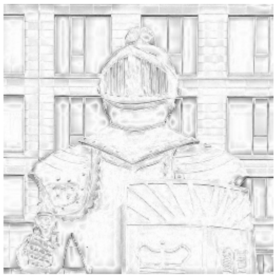

D4 with $k = 3$

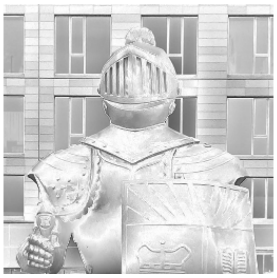

D6 with $k = 3$

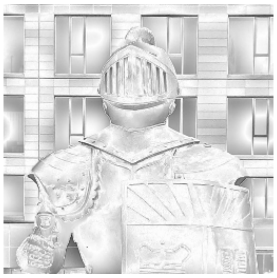
By aesthetic reasons, rather than $(W_k^{-1} \circ Z_k \circ W_k)(A)$ the images depicted here show minus the absolute value of its entries. The minus is to get a white background that is more appealing than the negative image and the absolute value is because the "derivative" involved in the filter can be negative or positive and then the background would appear after scaling in a middle gray.

If we increase the value of k the results are far from being edge detectors because we would include details in a coarse scale. On the other hand it can be used to create some effects.


D4 with $k = 6$


D6 with $k = 4$


D8 with $k = 7$


D12 with $k = 6$

In the last two images the values of k are maximal for this size. This means that we are keeping all the details.

There are endless possible variations. For instance, if in (3.126) we change Z_k for the map that multiplies the $N/2^k \times N/2^k$ block by a constant $0 < \alpha < 1$ then we will boost the influence of the details creating a sharpen effect. One can play using different filters acting on this block and the results are often unexpected.

As a last comment, the techniques of compression and noise reduction described above for one-dimensional finite signals work in the same way for images [RVF09].

Suggested Readings. In [Wal08] there are many examples, applications and proposed projects. Although its title, [RVF09] enters quite deeply in the theory but it also gives a number of applications, mainly to images, with good explanations. Many books on image processing, like [GW08], have chapters or sections devoted to wavelets but usually as a toolbox and they do not enter into details

3.3 Problem set and challenges

Note: These exercises were created for the assessment during the master course at UAM *Wavelets and signal processing* 2017/2018.

PROBLEM SET 3

Wavelets: theory and practice

Problems

1) With the notation of the notes, assume that when processing a black-and-white JPEG image, we have $a_{nm} = 8\alpha_n\alpha_m \cos\left(\frac{\pi n}{16}\right)\cos\left(\frac{\pi m}{16}\right)$ for every block if n and m are both even and $a_{nm} = 0$ otherwise. How does the image look like?

2) Prove the Fourier expansions included in (3.2).

3) For some questions of analysis it is convenient to consider the wavelet transform associated to the so-called *analytic wavelet* $\psi(x) = (x+i)^{-n-1}$ for some fixed $n \in \mathbb{Z}^+$. Prove that it is a continuum wavelet and normalize it.

4) Given a continuum wavelet $\psi \in C_0^\infty$ with vanishing moments $\int_{-\infty}^{\infty} x^j \psi(x)\,dx = 0$ for $0 \le j < n$. Prove that if f belongs to the Sobolev space $H^s(\mathbb{R})$ and $n > s > 0$ then $|a|^{-1-s} W_\psi f(a,b) \in L^2(\mathbb{R}^2)$.

5) Compute the Haar wavelet expansion of the function $f(x) = x$ for $x \in [0,1)$ and $f(x) = 0$ otherwise. In other words, compute the coefficients c_{jk} giving

$$f(x) = \sum_{j,k\in\mathbb{Z}} c_{jk}\psi(2^j x - k)$$

where ψ is the Haar wavelet.

Notes and hints

1) I expect a mathematical argument. It is not valid saying "it looks like this" without further explanations. A short code can help you to know what to prove.

2) This is not a problem to try your patience with calculations. There are not such calculations, you have to find the trick bypassing the integrals that, by the way, WolframAlpha® is not able to compute. Hint (end of the course gift): What happen when you compute $e^{\cos\alpha + i\sin\alpha}$ substituting $\cos\alpha + i\sin\alpha$ in the Taylor expansion of e^x? For the peak, you can use the Fourier expansion of the Fejér kernel appearing in the notes that you proved in the first problem set.

If you downloaded very early §3.1.1, note that I corrected a misprint in (3.1) in the last minute. The right version is the one currently uploaded.

3) If necessary, dust off your notes on complex analysis. If you compute $\widehat{\psi}$ explicitly I expect an explanation, something more informative that "WolframAlpha said so" because for this complex integrals you cannot fully trust on it. A hint is that they are called analytic wavelets because they verify $\widehat{\psi}(\xi) = 0$ for $\xi < 0$.

Actually, this kind of wavelets are defined for $n \in \mathbb{R}^+$ and $\widehat{\psi}$ can be computed explicitly in this general case with complex analysis techniques too. In case you are curious about the "questions of analysis", read the next exercise. This wavelet verifies the condition on the moments. For fractional n, it is employed to know if certain strange functions are Hölder continuous.

4) Recall that $H^s(\mathbb{R}) = \big\{ f \in L^2(\mathbb{R}) \ : \ |\xi|^s \widehat{f}(\xi) \in L^2(\mathbb{R}) \big\}$ for $s > 0$. Do not get impressed with this unusually abstract exercise. Try to link the moments with the Fourier transform and, of course, with the wavelet transform. Does the proof of the inversion formula for the wavelet transform in the notes ring a bell?

The motivation for this problem is to illustrate the fact that the more vanishing moments the better behavior of the wavelet transform, a point that we did not develop in the course.

5) In `http://matematicas.uam.es/~fernando.chamizo/dark/d_haar.html` I have plotted some partial sums. This can be useful to check your results or to guess them.

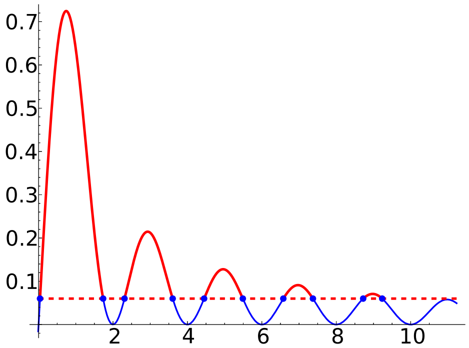
Experimental challenge: **The new wave**

Wavelets: theory and practice

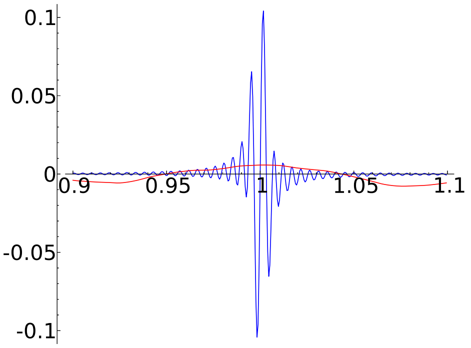
Experimental part

This experiment is demanding. If you are not good using mathematical software, consider to skip it. You will get a big help if you adapt the `SAGE` code following the link "The Meyer wavelet" in the complementary material.

Consider $f(x) = \operatorname{sgn}(x)\chi_{[-1/2,1/2]}(x)$ with Fourier transform $\widehat{f}(\xi) = 2(\pi\xi)^{-1}\sin^2(\pi\xi/2)$. Write a code that given $\delta > 0$ recovers f considering only the values with $|\widehat{f}| > \delta$, the part indicated in the first figure. In other words, approximate numerically $\int_{-\infty}^{\infty} g_\delta\big(\widehat{f}(\xi)\big)e(x\xi)\,d\xi$ where $g_\delta(x) = x$ for $|x| > \delta$ and vanishes otherwise. Write also a code to analyze f in terms of the Meyer wavelet and truncate the series $\sum c_{jk}\psi_{jk}$ to $\sum g_\delta(c_{jk})\psi_{jk}$.



Nonzero part of $g_{0.06}(\widehat{f})$



Fourier (blue) Meyer (red) for $\delta = 0.003$

Plot the approximate reconstruction in both cases and check, as depicted in the second figure, that wavelets (the new waves) avoid the strange artifacts introduced by Fourier analysis around $\pm n/2$ with $n \in \mathbb{Z}_{\ge 2}$. It is fair to mention that Fourier analysis is spotless in other zones and there it works better than this simple instance of wavelet.

Mathematical part

The experimental part is demanding enough to ask additional difficult mathematical questions. Just check the formula for $\widehat{f}$ and prove that Meyer's wavelet belongs to $C^\infty \cap L^1$. This is not automatic from the formula because of the possible vanishing of the denominators.

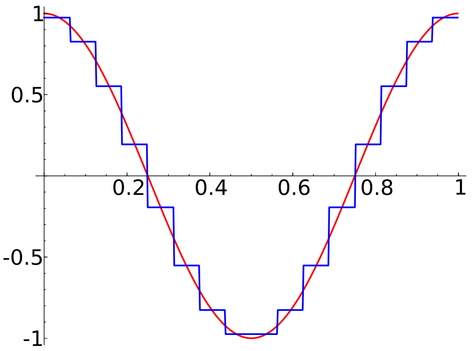
Experimental challenge: **Step by step**

Wavelets: theory and practice

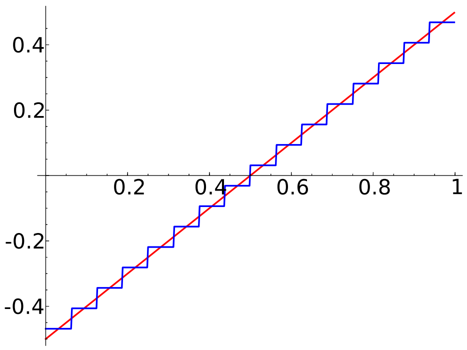
Experimental part

Choose a 1-periodic signal $f = f(t)$ with zero average on each period and such that you know an explicit formula for $F(t) = \int_0^t f$. This is the case, for instance, for a cosine wave or a sawtooth signal.

Write code with the software you prefer to find the contribution of $j < J$ to $\sum \sum c_{jk}\psi_{jk}$, the wavelet expansion of $f\chi_{[0,1]}$ with ψ the Haar wavelet. Plot the result for some values of J.



Cosine signal for J = 4



Sawtooth for J = 4

If you sample the result and repeat the data a number of times you can hear an approximation of the original signal quantized in time. As you can check in the complementary material, for a pure tone sometimes a strange high pitch appears. This is due to the high frequencies introduced by the sharp jumps.

Mathematical part

Prove that this procedure of truncating the wavelet expansion to $j < J$ is actually equivalent to quantize in time substituting the signal by the step function that gives its average on each interval $\big[2^{-J}k, 2^{-J}(k+1)\big)$, $k \in \mathbb{Z}$. This is very simple in many ways, for instance thinking about the multiresolution analysis.

Theoretical challenge: **Periodically explicit**

Wavelets: theory and practice

The challenge

For $J \in \mathbb{Z}^+$ let f_J be the 1-periodic extension of

$$\sum_{j=0}^{J-1} \sum_{k=0}^{2^j-1} 2^{-j-1}\Big(\psi(2^{j+1}x-k)+\psi(-2^{j+1}x-k)\Big) \qquad \text{for} \quad |x| \le \frac{1}{2},$$

where ψ is the Haar wavelet. Give a simple fully explicit formula for the coefficients c_n^J in the Fourier expansion $\sum c_n^J e(nx)$ of f_J.

Comments

Let $c_n^\infty = \lim_{J\to+\infty} c_n^J$. With your formula you can check that although $b_n^J = c_n^J / c_n^\infty \to 1$ for each fixed n with $c_n^\infty \ne 0$, we have $b_{2^{J+1}-1}^J \to -\infty$. This is related to the high pitch that one hears sometimes when quantizing in time a pure tone (see the experimental challenge "step by step").

Chapter 4

Some computational aspects

4.1 The Fast Fourier Transform

4.1.1 The basic algorithm

If there was a world scale survey among scientists about the most important algorithm, surely the *fast Fourier transform*, abbreviated FFT, would get a lot of votes. It is an algorithm to compute the DFT when N is large and its main assets are that it is very simple and at the same time very powerful.

Recall the definition in (2.51) of the DFT and imagine that it costs nothing to compute the N-th roots of the unity $e(n/N)$, for instance because they are stored in memory. The computation of $\widehat{x}_n$ requires N multiplications by the roots of the unity and the sum of the results. Then to compute the whole DFT $\widehat{x}_0, \dots, \widehat{x}_{N-1}$ by brute force we need a number of operations comparable to N^2. We have in mind a value of N so large that N^2 makes a bottleneck in a computer approach to a problem, but even with small values of N if one forces herself to do the computations by hand (recall that *human computers* were still in use in the early 70s), it is easy to guess some redundancy in the calculations due to the group structure of the roots of the unity. Officially the materialization of this vague idea into the FFT algorithm came to light in the celebrated paper [CT65] by J.W. Cooley and J.W. Tukey but perhaps it is fair to emphasize that the important point was to focus on the automatic calculation, already appearing in the title[1]. The method can be traced back to an old work of the *princeps mathematicorum* C.F. Gauss disguised as a trigonometrical interpolation help for calculations of orbits (needless to say, by hand!) [HJB85].

Let us assume that N is even, the key idea is summarized in a single line:

$$\widehat{x}_n = y_{n0} + e\Big(-\frac{n}{N}\Big) y_{n1} \quad \text{with} \quad y_{nj} = \sum_{m=0}^{N/2-1} x_{2m+j}\, e\Big(-\frac{mn}{N/2}\Big). \tag{4.1}$$

This is of course rather obvious, where is the win? The number of multiplications by brute force was before N^2. Now for each n by brute force y_{n0} and y_{n1} require $N/2$ each. So far

[1] As a matter of fact the second author is credited for introducing the word *bit*.

nothing up the sleeve because $N/2 + N/2$ gives the N multiplications needed for each $\widehat{x}_n$, in fact it is a little worse if we count the extra multiplication by the $e(-n/N)$ factor which by mysterious reasons is called *twiddle factor* in the literature. The important point is that y_{n0} and y_{n1} are $N/2$-periodic then it is only necessary to compute them for $0 \le n < N/2$ and the number of multiplications by brute force reduces to

$$M(N) = \frac{N}{2} \cdot \frac{N}{2} + \frac{N}{2} \cdot \frac{N}{2} + N = 2\Big(\frac{N}{2}\Big)^2 + N \tag{4.2}$$

where the last N comes from the multiplication by $e(-n/N)$. In principle this is noticeable but not very impressive, we have just improved the constant of the main term N^2 from 1 to 1/2. Note that the $(N/2)^2$ comes from computing the DFT for $N/2$ by brute force and if N is multiple of 4 we can repeat the idea once more to delay another step the use of brute force wining a new 1/2. The punchline is easy to guess: if $N = 2^k$ we have a *divide and conquer algorithm*, as it is said in computer science, or an inductive argument, as it is said in mathematics, that leads the computation to the trivial case $N = 1$ and via the relation

$$M(N) = 2M(N/2) + N \tag{4.3}$$

shows $M(N) = (k+1)2^k$ starting from $M(1) = 1$. The same can be said with additions and then the number of the floating point complex operations is

$$2(k+1)2^k = 2N\log_2(2N) \qquad \text{for} \quad N = 2^k. \tag{4.4}$$

Actually this is an upper bound because the single multiplication for $N = 1$ is by 1 and something better can said omitting this and other trivial operations [Yav68] [Bla10, §3.2]. Anyway we get something like $CN\log N$ with a moderate constant C and it is a fantastic improvement[2] to pass from N^2 to something almost linear.

The natural question is: What happens if our finite discrete signal has not a power of two of elements? The commonly more efficient answer is that you pretend it has, completing your signal to $N = 2^k$ with fake data. In the next subsection we will see some alternatives. If we have no qualms about padding invented data, from the point of view of the performance it is not as bad as it sounds. In the worst case scenario $N = 2^k + 1$ one has to forge $N - 2$ data to reach the next power of two and duplicating the data asymptotically duplicates the constant in the number of operations halving the performance. This is nothing in comparison with the brute force alternative. In fact this case is unrealistic because if N is large it would be harmless to forget a value of the signal considering $N - 1$.

A last comment is that we have assumed that all the N-th roots of the unity were precomputed. In reality this is not needed because with $e(1/N)$ we can compute the rest

[2] In [Ger99] it is claimed "When Cooley (then at IBM) first presented the FFT, IBM concluded that it was so significant it should be put in the public domain to prevent anyone from trying to patent it, and so it was published openly".

and it does not have a fundamental influence on the number of operations [SS03, §7.13], [Str86, §5.5].

Suggested Readings. The FFT is so important that the textbooks tend to present it with great verbosity (perhaps the same can be applied to this subsection). Sometimes one wonders how on earth did people introduce so many terminology out of a single line like (4.1). Three exceptions with quick and clear explanations are [Bré02], [Ger99] and [SS03]. The original paper [CT65] is still a good source but it goes beyond $N = 2^k$ skipping to our subsequent considerations.

4.1.2 The FFT for arbitrary length

There is quite a number of variants and generalizations of the original FFT algorithm [Bla10] [MR97]. We deal here with some of them addressed to treat the case $N \neq 2^k$ in an exact manner, without padding fake data that affect the result.

The role of the number 2 in (4.1), although the best choice, is not so fundamental. Let us say that we have a nontrivial factorization $N = N_1N_2$. Dividing by N_1 each $0 \le n < N$ is expressed as $N_1n_2 + n_1$ with $0 \le n_1 < N_1$ and $0 \le n_2 < N_2$. Using this in (2.51) and exchanging the role of N_1 and N_2 in the variable of summation, we have

$$\widehat{x}_n = \sum_{m_1=0}^{N_1-1}\sum_{m_2=0}^{N_2-1} x_{N_2m_1+m_2} e\Big(-\frac{(N_1n_2+n_1)(N_2m_1+m_2)}{N_1N_2}\Big) \tag{4.5}$$

that can be arranged as

$$\widehat{x}_n = \sum_{m_2=0}^{N_2-1}\Big(e\Big(-\frac{n_1m_2}{N}\Big)\sum_{m_1=0}^{N_1-1} x_{N_2m_1+m_2} e\Big(-\frac{n_1m_1}{N_1}\Big)\Big)e\Big(-\frac{n_2m_2}{N_2}\Big). \tag{4.6}$$

For $N_2 = 2$ we recover (4.1). For each m_2 the innermost sum is a DFT of size N_1. Once we have all the possible results, we have to multiply by the exponential and compute a DFT of size N_2 of for each choice of n_1. Then using this interpretation of the generalization of (4.1), the number of multiplications satisfies

$$M(N) = N_2M(N_1) + N + N_1M(N_2). \tag{4.7}$$

With the brute force estimate in the right hand side we have $N_2N_1^2 + N + N_1N_2^2 = N(N_1 + N_2 + 1)$ and if $N_2 > 2$ we still gain a factor to the trivial estimate N^2. This shows that if $N \neq 2^k$ we can apply the algorithm and the benefit will depend on the number of prime factors counting repetitions. The case $N = 2^k$ is optimal because it reaches the upper bound $\log_2 N$ for the number of prime factors. On the other hand, if N is a prime number the process cannot even start.

This presentation in terms of the factorization is the one appearing in the original paper [CT65] noticing that "Whenever possible, the use of $N = r^m$ with $r = 2$ or 4 offers important advantages for computers with binary arithmetic".

Rader algorithm [Rad68] addresses the case in which N is prime employing an argument based on convolutions and a basic arithmetical result. On the other hand *Bluestein* *algorithm* [Blu70] [RSR69] is also based on convolutions and allows to treat other values of N.

Before explaining these algorithms we point out that a conspicuous application of the FFT is the quick computation of the convolution $\vec{x} * \vec{y}$ of two discrete signals $\vec{x} = (x_0, \dots, x_{N-1})$ and $\vec{y} = (y_0, \dots, y_{N-1})$ defined as in (2.56). By brute force a convolution requires N^2 multiplications and the sum of their results. On the other hand, a way of reading (2.57) is saying that $\vec{x} * \vec{y}$ is the inverse DFT of the product of the DFTs of $\vec{x}$ and $\vec{y}$. With this roundabout computing $\vec{x} * \vec{y}$ requires three DFTs and N multiplications and this is very advantageous if N has many prime factors. Note also that if $\vec{y}$ is fixed its DFT can be precomputed and stored. A comment aside is that the product of two integers can be considered as a convolution of the collection of their digits and then it is possible to use the FFT to do humble integer multiplications. This is the basis of the famous *Schönhage-Strassen algorithm* that multiplies two numbers of n digits with an amount of bit operations comparable to $n \log n \log\log n$, which is quite impressive (can you estimate how many digit operation you need doing the calculation by hand?), but it is only efficient for large numbers because the involved constant is big.

Rader and Bluestein algorithms write the DFT of $\vec{x}$ as a convolution for an N different from the original expecting to apply the FFT algorithm in a favorable situation.

For Rader algorithm it is needed the following arithmetic well known fact that can be rephrased saying that the group $(\mathbb{Z}_N^*, \cdot)$ for N prime is cyclic. Although the proof is elementary, it is quite ingenious. It would be a very tough exercise for a freshperson.

Lemma 4.1.1. *If N is prime then there exist* primitive roots *modulo N, that is, numbers g such that g^j gives all the residue classes distinct from 0 for $0 \le j < N - 1$.*

Proof. The order of $n \in R = \{1, \dots, N-1\}$ is $o(n) = \min\{m > 0 \ : \ n^m \equiv 1 \pmod{N}\}$. We now that $n^{N-1} \equiv 1 \pmod{N}$ (Fermat's little theorem) and $o(n)$ divides $N-1$. Given $n_1, n_2 \in R$ it is not difficult to construct n_3 with order the least common multiple of $o(n_1)$ and $o(n_2)$ (consider $n_3 = n_1^k n_2$). Let M the least common multiple of $o(1), \dots, o(N-1)$. If $M = N - 1$ then we can construct a g with $o(g) = N - 1$ and it is a primitive root. Otherwise, if M properly divides $N-1$, $j^M - 1 \equiv 0 \pmod{N}$ for every $j \in R$ implies that $x^M - 1$ is divisible modulo N by every $x - j$ and this contradicts $M < N - 1$. □

Let g be a primitive root modulo N and define $v_j = \widehat{x}_{g^{-j}}$ and $w_l = \widehat{x}_{g^l}$. Then changing in (2.51) m by g^{-j} we have

$$w_l = x_0 + \sum_{j=1}^{N-1} v_j e\Big(-\frac{g^{l-j}}{N}\Big) = x_0 + \sum_{j=0}^{N-2} v_j e\Big(-\frac{g^{l-j}}{N}\Big). \tag{4.8}$$

Rader algorithm proposes to consider the sum as the convolution of $(v_0, \dots, v_{N-2}) \in \mathbb{C}^{N-1}$ and $\big(e(-g^0/N), \dots, e(-g^{N-2}/N)\big) \in \mathbb{C}^{N-1}$. This latter vector does not depend on the data. In this way one can compute all the w_l and hence the $\widehat{x}_n$ for $n \ne 0$ (for $\widehat{x}_0$ use the original definition) as a convolution of length $N-1$ that requires two DFTs. As N is prime, $N-1$ is not and one can apply the ideas above on the conventional FFT treating other large prime factors employing Rader algorithm again.

If N is not prime the Rader algorithm does not work because the existence of primitive roots is not guaranteed. There is nevertheless a way of saving part of the idea using a polynomial interpretation of the structure of the multiplicative group modulo N. This is the basis of the *Winograd algorithm* [Win78] [Bla10].

The Bluestein algorithm takes as starting point the trivial identity

$$\widehat{x}_n = e\Big(-\frac{n^2}{2N}\Big)\sum_{m=0}^{N-1}\Big(x_m e\Big(-\frac{m^2}{2N}\Big)\Big)e\Big(\frac{(m-n)^2}{2N}\Big). \tag{4.9}$$

It is tempting to claim that the sum is a convolution of length N with a fixed signal. First, it would be useless because it would pass from a DFT to two DFTs of the same length and second, it is not a convolution. It does not match the definition (2.56) because $e\big(-m^2/(2N)\big)$ is not N-periodic. Sometimes an expression of this kind is called a non cyclic convolution. Let us rephrase it in this way, if a_k is defined for $0 \le k < N$ and b_k is defined by $-N < k < N$, in general

$$\sum_{k=0}^{N-1} a_k b_{n-k} \ne \sum_{k=0}^{N-1} a_k b_{r_N(n-k)} \tag{4.10}$$

where $r_N(n-k)$ is the remainder when $n-k$ is divided by N. Let $N' \ge 2N-1$ and define $a_k^* = a_k$ if $0 \le k < N$ and $a_k^* = 0$ if $N \le k < N'$ and $b^*_{r_{N'}(k)} = b_k$ if $-N < k < N$ and $b^*_{r_{N'}(k)} = 0$ if $N \le k < N'-N+1$. Then

$$\sum_{k=0}^{N-1} a_k b_{n-k} = \sum_{k=0}^{N'-1} a_k^* b^*_{r_{N'}(n-k)} \qquad \text{for} \quad 0 \le n < N. \tag{4.11}$$

Note that the equality holds in this range of n but not necessarily beyond. Note also that this artificial padding of zeros is very different from the padding of fake data that we mentioned before because the later changes the value of the DFT and here we have an identity. With this identity we can consider that (4.9) is actually a convolution of length N' to our choice with the only restriction $N' \ge 2N-1$. Taking the first power of two in this range we can apply the basic algorithm of the FFT to compute the convolution.

The conclusion of all the available methods is that computing the FFT for a fixed N requires a number of operations comparable to $N\log N$ irrespectively the factorization [BCT91]. It makes the FFT a very efficient and powerful algorithm that deserves its fame.

Suggested Readings. As mentioned, the original paper on the FFT [CT65] considers the algorithm in terms of the factorization. A survey with a lot of information is [MR97] but note that its bulk deals with an abstract group setting that we have not considered. There is a whole chapter in [PM96] about the FFT. The book [Bla10] is devoted mostly to the FFT, its variants and problems related to the computation of the DFT.

4.2 Coding and data compression

4.2.1 Entropy and codes

Consider a finite probability space. This means a finite set $S = \{s_1, \dots, s_n\}$ such that each element s_j carries a probability p_j. Of course $p_j \ge 0$ and $\sum_j p_j = 1$. With a small abuse of notation we use S to mean indistinctly the set and the probability space. We also assume in practice $p_j > 0$ because $p_j = 0$ would indicate that s_j has no chance to appear and we can delete it from S.

From the point of view of information theory S is a set of symbols, an *alphabet*, for instance the usual characters appearing on a keyboard, that we intend to employ to transmit information (or whatever you write in emails). The big bang in this subject was the celebrated paper [Sha48] by Shannon. It was written before the development of computer science, the famous ENIAC (Electronic Numerical Integrator and Computer) started its military calculations barely two years before, but there was some interest in digital communications. To give an idea about how influential was [Sha48] it is enough to say that the nowadays widespread term *bit*, created by Tukey, appeared there for the first time in a printed paper.

Shannon wondered how to measure the amount of information contained in S, roughly speaking the "uncertainty" when we extract a sample from S (see [Rén84]). For instance, if $p_1 \approx 1$ and $p_2 = p_3 = \cdots = p_n \approx 0$ then the uncertainty is zero because we are fairly sure that we are going to pick almost all the time the element s_1. On the other hand when we play the lottery $p_1 = p_2 = \cdots = p_n = 1/n$ and the uncertainty is maximal. Shannon proceed theoretically and considered this uncertainty as a mathematical function of the probabilities $H = H(p_1, \dots, p_n)$ that he called *entropy*[3]. He imposed that H is increasing on n in the case $p_1 = p_2 = \cdots = p_n = 1/n$ (more lottery tickets, less chance of guessing the result) and that the uncertainty does not change if we divide the set into subsets (there are not lottery lucky spots if we take into account the number of tickets they sell). This can be written saying that for any $b_i \in \mathbb{Z}^+$ with $\sum_{i=1}^k b_i = n$, we have

$$H(\frac{1}{n}, \overset{n \text{ times}}{\dots}, \frac{1}{n}) = H(\frac{b_1}{n}, \frac{b_2}{n}, \dots, \frac{b_k}{n}) + \sum_{i=1}^{k} \frac{b_i}{n} H(\frac{1}{b_i}, \overset{b_i \text{ times}}{\dots}, \frac{1}{b_i}). \tag{4.12}$$

(Here S is thought to be divided into subsets of size b_i, see [Rom92, §1.1] [Mac03, §2.5] for more on this). A little lemma proves that the only continuous functions with these properties are $H = K \sum_j p_j \log p_j$ with K a negative constant. Due to the interest of the digital case it is convenient to take this normalization constant to be $-(\log 2)^{-1}$; so Shannon gave as the formula for the entropy

$$H(p_1, \dots, p_n) = -\sum_{j=1}^{n} p_j \log_2 p_j. \tag{4.13}$$

In the case $p_j = 0$ that we have avoided we should define $p_j \log_2 p_j = 0$ to keep the continuity. To grasp the convenience of the choice of the constant, note that if S contains

[3] The name and the notation came from a concept measuring the randomness of a physical system that appeared many years before in the work of L. Boltzmann on statistical mechanics and thermodynamics.

all the possible words of N zeros and ones (bits) with the same probability, we have $n = 2^N$, $p_j = 2^{-N}$ and hence random lists of N bits have entropy $H = N$. We shall analyze closely later this relation between the bit length and the entropy but first of all we have to introduce a way of digitizing the elements of S.

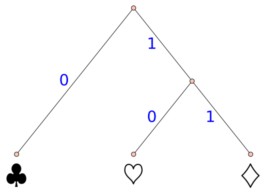
As we have suggested, the elements of S must be thought as the building blocks to construct something carrying information. For instance, they could be letters or words to compose a text. In modern world any piece of information eventually becomes a list of bits (zeros and ones). So we define a (binary) *code* as a function

$$C : S \longrightarrow B^* \qquad \text{with} \quad B^* = \{\text{nonempty finite lists of bits}\} \tag{4.14}$$

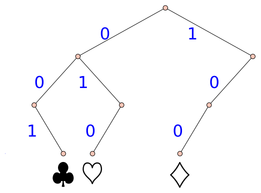
that extends by concatenation to lists of elements of S. For instance, if C is defined on $S = \{\clubsuit, \heartsuit, \diamondsuit\}$ as $C(\clubsuit) = \texttt{0}$, $C(\heartsuit) = \texttt{10}$, $C(\diamondsuit) = \texttt{11}$, then $C(\heartsuit\clubsuit) = \texttt{100}$ and $C(\clubsuit\heartsuit\diamondsuit) = \texttt{01011}$. Very often the finite lists of elements or bits are called *strings*. Given a list, ℓ denotes its *length*, the number of its elements. In the previous example $\ell(C(\heartsuit)) = 2$ and $\ell(C(\clubsuit\heartsuit\diamondsuit)) = 5$.

We want to avoid any ambiguity. With the previous S if we define $C(\clubsuit) = \texttt{0}$, $C(\heartsuit) = \texttt{1}$, $C(\diamondsuit) = \texttt{01}$, we will put in trouble any sequential decoder because `01` can mean $\clubsuit\heartsuit$ or $\diamondsuit$. To avoid this situation we restrict ourselves here to *prefix codes*, such that the image of an element of S cannot be the *prefix* (the starting bits) of the image of another.

If we decide that `0` means lower left and `1` lower right, each prefix code is represented uniquely by a rooted binary tree. Recall that in graph theory this means a *tree* (two vertexes are connected by only a path) in which we start by a root and any vertex has at most two children. The idea is quickly caught with the examples below rather than reading abstract definitions[4].



$C(\clubsuit) = \texttt{0}$
$C(\heartsuit) = \texttt{10}$
$C(\diamondsuit) = \texttt{11}$



$C(\clubsuit) = \texttt{001}$
$C(\heartsuit) = \texttt{010}$
$C(\diamondsuit) = \texttt{100}$

The elements of S can be identified with the *leaves* of the tree, the vertexes without children.

The simplest way to assure the prefix property is to proceed as in the last example assigning to each symbol different strings of bits of the same length. In computer science the most famous fixed length code is the ASCII (American Standard Code for Information Interchange) that encodes each standard character into a byte, a string of 8 bits. An also very famous code with the same purpose but variable length and the possibility of including more complicate characters is UTF-8 (8-bit Unicode Transformation Format) that extends ASCII. The image of the code contains strings of 8, 16, 24 and 32 bits.

[4] Graph theory usually employs the opposite convention than Nature: The root of the tree is put at the top and the leaves at the bottom.

Imagine that in $\{\mathtt{a},\mathtt{b},\mathtt{c},\mathtt{d}\}$ we define the most obvious code

$$C(\mathtt{a}) = \mathtt{00}, \qquad C(\mathtt{b}) = \mathtt{01}, \qquad C(\mathtt{c}) = \mathtt{10}, \qquad C(\mathtt{d}) = \mathtt{11} \tag{4.15}$$

and the alternative code

$$C'(\mathtt{a}) = \mathtt{0}, \qquad C'(\mathtt{b}) = \mathtt{10}, \qquad C'(\mathtt{c}) = \mathtt{110}, \qquad C'(\mathtt{d}) = \mathtt{111}. \tag{4.16}$$

Which one is better? It seems that the first one because transmitting N elements requires $2N$ bits while with C' the string formed by N consecutive $\mathtt{d}$'s requires $3N$. This reasoning is flawed if $\mathtt{d}$ has probability close to zero and $\mathtt{a}$ has probability close to one because in this case we will transmit very often $\mathtt{a}$ and it is convenient to use a shorter code for it at the expenses of putting a larger image of $\mathtt{d}$ that will be employed very seldom. Consequently, we define the *average length* ℓ_C of a code C on $S = \{s_1, \dots, s_n\}$ as the expectation of $\ell\big(C(s_j)\big)$. In a formula

$$\ell_C = \sum_{j=1}^{n} p_j \ell\big(C(s_j)\big). \tag{4.17}$$

To save bits, and ultimately to compress, we would like to have codes with the smallest average length as possible. The following result impose a limit in terms of the entropy.

Theorem 4.2.1 (source coding theorem)**.** *Given S we have*

$$H \le \min_{C} \ell_C < H + 1 \tag{4.18}$$

where $C : S \longrightarrow B^*$ runs on all possible codes and H is the entropy of S.

This is also called sometimes *Shannon's noiseless coding theorem* because it was stated by Shannon and there is a version that involves the possibility of noise [Ham80, Ch.10].

We consider here code as a synonym of prefix code but the result is valid for any uniquely decodable code, any code that can be decoded without ambiguity. In fact the proof is the same once one knows that for every uniquely decodable code C there exists a prefix code C_p such that $\ell(C(s_i)) = \ell(C_p(s_i))$. This nontrivial fact (see Proposition 4.2.3 below and [Rom92]) implies that in any result involving only the length and the entropy we can always restrict ourselves to prefix codes.

If we have only two elements $\{\mathtt{x},\mathtt{y}\}$ the best we can do is $C(\mathtt{x}) = \mathtt{0}$, $C(\mathtt{y}) = \mathtt{1}$. In this case it can be convenient to consider strings of a certain length instead elements. For instance if $\mathtt{x}$ has a much bigger probability a code like (4.16) will be fruitful changing $\{\mathtt{a},\mathtt{b},\mathtt{c},\mathtt{d}\}$ by $\{\mathtt{xx},\mathtt{xy},\mathtt{yx},\mathtt{yy}\}$. With this idea in mind, for $S = \{s_1, \dots, s_n\}$ we define S_N as the set of strings of length N carrying the probability induced by the assumption of independence. Namely,

$$S_N = \{s_{i_1}s_{i_2}\cdots s_{i_N} \ : \ s_{i_k} \in S\}, \qquad \operatorname{Prob}(s_{i_1}s_{i_2}\cdots s_{i_N}) = p_{i_1}p_{i_2}\cdots p_{i_N}. \tag{4.19}$$

The average length of a code on S_N is defined as before by

$$\ell_{C,N} = \sum_{s\in S_N} \operatorname{Prob}(s)\ell\big(C(s)\big) = \sum_{i_1,\dots i_N} p_{i_1}\cdots p_{i_N}\ell\big(C(s_{i_1}\cdots s_{i_N})\big). \tag{4.20}$$

It is very easy to check that the entropy of S_N is N times the entropy of S then Theorem 4.2.1 gives an apparently stronger consequence.

Corollary 4.2.2. *With the notation as before, we have*

$$H \le \frac{1}{N}\min_C \ell_{C,N} < H + \frac{1}{N}, \qquad \textit{in particular } \lim_{N\to\infty} \frac{1}{N}\min_C \ell_{C,N} = H. \tag{4.21}$$

where $C : S_N \longrightarrow B^*$ runs on all possible codes and H is the entropy of S.

This result implies that if we use the alphabet S, a generic text of N characters requires at least HN bits and in fact for N large we can get something arbitrarily close to the rate of H bits per character with a suitable encoding. In few words, the entropy is the optimal rate bits per character.

Coming back to the example $S = \{\mathtt{x}, \mathtt{y}\}$, if $p_1 = p_2 = 1/2$ then $C(\mathtt{x}) = \mathtt{0}$, $C(\mathtt{y}) = \mathtt{1}$ is optimal according to $H = 1$. We can rephrase this saying that every text composed with N characters of S has the same probability 2^{-N} and then there is no gain in grouping elements, we need N bits. On the other hand, if $p_1 = 0.75$, $p_2 = 0.25$ then we can encode (compress) a generic file of N characters into something less than $0.812N$ bits for N large because $0.811\dots$ is the value of the entropy.

As we will see later, the algorithms employed by the file compression software theoretically tend to reach the optimal rate given by the entropy.

The following result is pivotal in the proof of Theorem 4.2.1 and in fact in the mentioned relation between prefix codes and more general codes. It appeared in [McM56], after Shannon's paper.

Proposition 4.2.3 (Kraft's inequality). *Given a code C defined on $S = \{s_1, \dots, s_n\}$, the lengths $l_i = \ell(C(s_i))$ satisfy*

$$\sum_{i=1}^{n} 2^{-l_i} \le 1. \tag{4.22}$$

Moreover, given positive integers l_i verifying this inequality there is exist a (prefix) code C such that $l_i = \ell(C(s_i))$.

Proof. Without loss of generality we assume $l_1 \le l_2 \le \cdots \le l_n$. Consider the perfect binary tree $\mathcal{T}$ of depth l_n. This means that it takes l_n steps (edges) to go from the root to the leaves and each vertex has exactly two children.

We know that the tree corresponding to the code C is a subtree of $\mathcal{T}$. A vertex of $\mathcal{T}$ which is k steps below the root has 2^{l_n-k} descendant leaves, whence for each j there are $2^{l_n-l_j}$ such leaves of $\mathcal{T}$ under the vertex corresponding to $C(s_j)$. Then

$$2^{l_n} = \#\{\text{leaves of } \mathcal{T}\} \ge \sum_{j=1}^{n} 2^{l_n-l_j} \tag{4.23}$$

that gives the first part for the result.

For the second part, we reverse the construction. We take a vertex in $\mathcal{T}$ of depth l_1 i.e., separated l_1 edges from the root, and we eliminate all of its descendants, in particular $2^{l_n - l_1}$ leaves of $\mathcal{T}$ if $l_1 \neq l_n$. If $l_1 = l_n$ the vertex is a leaf and there are not descendants to eliminate. We proceed in the same way successively with $l_2, \dots, l_n$ not repeating the choice of vertexes. The inequality $2^{l_n} \geq \sum_{j=1}^{n} 2^{l_n - l_j}$ assures that we can perform this algorithm because the sum counts the leaves that we have eliminated and the vertexes with $l_j = l_n$ that were already leaves of $\mathcal{T}$. The selected vertexes are the leaves of a binary tree included in $\mathcal{T}$ and then corresponds to a prefix code with the specified lengths. □

With this we are ready to prove the source coding theorem.

Proof of Theorem 4.2.1. Consider the function $f(x_1, \dots, x_n) = -\sum p_i \log_2 x_i$ defined on the simplex $\sum x_i = 1$, $0 \leq x_i \leq 1$. Using the method of Lagrange multipliers it is easy to prove that the minimum of f is the entropy. Let γ be the inverse of the sum in Kraft's inequality. Using the minimum property and $\gamma \geq 1$, we have

$$H \leq f(2^{-l_1}\gamma, 2^{-l_2}\gamma, \dots, 2^{-l_n}\gamma) \leq f(2^{-l_1}, 2^{-l_2}, \dots, 2^{-l_n}) = \ell(C) \tag{4.24}$$

and this proves the lower bound in (4.18).

For the upper bound take $l_i \in \mathbb{Z}^+$ such that $-\log_2 p_i \leq l_i < 1 - \log_2 p_i$. Note that this implies $\sum 2^{-l_i} \leq \sum p_i = 1$. By the last part of Proposition 4.2.3, there exists a code C such that $l_i = \ell(C(s_i))$. Then

$$\ell(C) = \sum p_j l_j < \sum p_j (1 - \log_2 p_j) = 1 + H \tag{4.25}$$

and the proof is complete. □

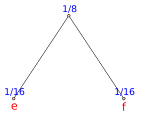
Suggested Readings. The old book [Rén84] is truly a masterpiece in originality and exposition. Also [Ham80] is a noticeable contribution (his author was not very happy with the existing bibliography for teaching and did a good job writing several textbooks in his late career). More standard books are [Mac03] and [Rom92]. The original paper [Sha48] perhaps can seem too weak for mathematicians and too abstract for engineers but it achieves its purpose of giving an attractive and readable overview of the subject. Part of the ideas can be found in previous mathematical papers by other authors but something like Shannon's exposition was necessary to capture the attention of people related to technology.

4.2.2 Huffman coding

The source coding theorem gives bounds for the possible average lengths ℓ_C. The natural intuition dictates that finding a code such that ℓ_C reaches a minimum, an optimal code, is a difficult task because the combinatorial possibilities to construct codes are enormous. Surprisingly there is a simple and efficient algorithm to find optimal codes. It was obtained by D.A. Huffman in 1952 [Huf52] when he was a graduate student and it became more important with the development of computer science. The resulting coding is called the *Huffman coding*.

Huffman algorithm constructs the tree of a prefix code out of S assigning successively to the two unused smallest probability elements a parent with the sum of their probabilities. The best way of understanding the simplicity of this rule is through toy examples.

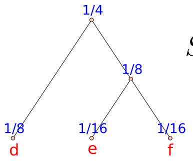
As a first example take $S = \{a, b, c, d, e, f\}$ where the elements of S have respective probabilities $p_1 = 1/2$, $p_2 = p_3 = p_4 = 1/8$, $p_5 = p_6 = 1/16$. We first select the smallest probabilities in $S_0 = S$. These are those of e and f, and the first step gives



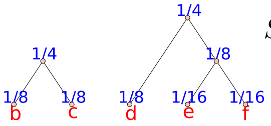
$$\begin{array}{lccccc} S_1 = \{ & a, & b, & c, & d, & \widehat{ef} \quad \} \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ \text{Prob} & \frac{1}{2} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \end{array}$$

where $\widehat{ef}$ means the parent of e and f. In general we denote $\widehat{xy}$ a common ascendant of x and y, a root of the subtree in which they are included.

Now there are several elements with the smallest probability 1/8 and there are several possible choices of a pair. Any of them gives the same average length. Let us choose for instance d and $\widehat{ef}$. In the next step there is not ambiguity because only b and c have the smallest probability. Then we choose them.

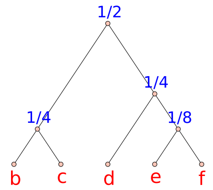


$$\begin{array}{lcccc} S_2 = \{ & a, & b, & c, & \widehat{df} \quad \} \\ & \downarrow & \downarrow & \downarrow & \downarrow \\ \text{Prob} & \frac{1}{2} & \frac{1}{8} & \frac{1}{8} & \frac{1}{4} \end{array}$$

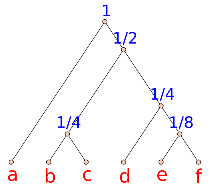


$$\begin{array}{lccc} S_3 = \{ & a, & \widehat{bc}, & \widehat{df} \quad \} \\ & \downarrow & \downarrow & \downarrow \\ \text{Prob} & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \end{array}$$

In the following step we are force to team up $\widehat{bc}$ and $\widehat{df}$. Finally we group their parent and a to get the root of the final tree.

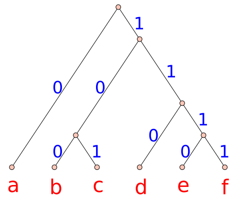


$$\begin{array}{lcc} S_4 = \{ & a, & \widehat{bf} \quad \} \\ & \downarrow & \downarrow \\ \text{Prob} & \frac{1}{2} & \frac{1}{2} \end{array}$$



$$\begin{array}{lc} S_3 = \{ & \widehat{af} \quad \} \\ & \downarrow \\ \text{Prob} & 1 \end{array}$$

Once the tree is constructed, we encode the descending path from the root to each leaf $s \in S$ as explained before putting zeros to the left and ones to the right (of course this is purely conventional). In our example we get



$$\begin{array}{ll} a \to 0 & d \to 110 \\ b \to 100 & e \to 1110 \\ c \to 101 & f \to 1111 \end{array}$$

The mean length is

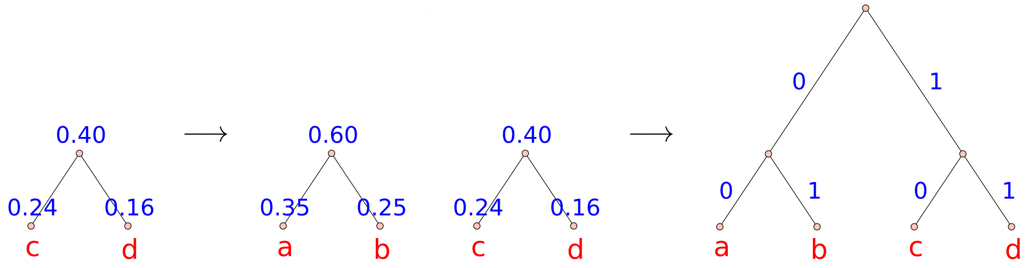
$$\ell_C = 1 \cdot \frac{1}{2} + 3 \cdot \frac{1}{8} + 3 \cdot \frac{1}{8} + 3 \cdot \frac{1}{8} + 4 \cdot \frac{1}{16} + 4 \cdot \frac{1}{16} = \frac{17}{8} \tag{4.26}$$

and the entropy

$$H = -\frac{1}{2}\log_2\frac{1}{2} - 3\frac{1}{8}\log_2\frac{1}{8} - 2\frac{1}{16}\log_2\frac{1}{16} = \frac{17}{8}. \tag{4.27}$$

Then the coding is optimal after Theorem 4.2.1.

The fact that the entropy bound is not always reached appeared implicitly in the proof of Theorem 4.2.1. Taking as granted that Huffman coding is always optimal it is easy to cook simple examples. For instance if we assign to the elements of $S = \{a, b, c, d\}$ more or less alike different probabilities then the algorithm will give the four possible 2-bit strings and the entropy will not be 2. So, choosing $p_1 = 0.35$, $p_2 = 0.25$, $p_3 = 0.24$, $p_4 = 0.16$ Huffman coding gives the following tree scheme:



that corresponds to the coding $a \to 00$, $b \to 01$, $c \to 10$, $d \to 11$. Clearly $\ell_C = 2$ and $H = 1.9472\ldots$ is close but not equal to ℓ_C.

Once we have understood the algorithm we are going to prove the result that shows its relevance. Being very rigorous rather that talking about "the Huffman coding" we would say "a Huffman coding" because if several probabilities coincide then there are more than one way to proceed. Perhaps it is not completely obvious with the previous examples that this could give different sets $\{\ell(C(S))\}$. Nevertheless ℓ_C remains unchanged for all the possibilities. Then to study the optimality we can pretend that there is only a Huffman coding.

Theorem 4.2.4. *Given S, the minimum value of ℓ_C for all possible codes $C : S \longrightarrow B^*$ is reached by the Huffman coding.*

A key part of the proof is the following auxiliary result (cf. [GP91, §1.7]).

Lemma 4.2.5. *Given S with $\#S \geq 2$ there exists a code $C : S \longrightarrow B^*$ giving a minimum of ℓ_C and such that there are two elements in S that are siblings in the corresponding tree and have the smallest probabilities.*

Proof. Let C be any code with ℓ_C minimum. If the maximal length is reached for the codification of only one element $s \in S$ then we can forget the last bit and we still have a valid code. This contradicts that ℓ_C is minimum. Then s has a sibling t. If $p_s > p_u$ for some $u \neq s$ then exchanging $C(u)$ and $C(s)$ would reduce ℓ_C getting again a contradiction, then p_s is a minimum of the probabilities. In the same way, swapping the codifications of u and t cannot reduce ℓ_C and hence

$$p_t\ell(C(t)) + p_u\ell(C(u)) \le p_t\ell(C(u)) + p_u\ell(C(t)). \tag{4.28}$$

This inequality proves that $p_u < p_t$ only can happen if $\ell(C(t)) = \ell(C(u))$ because $\ell(C(t))$ is maximal. Then we can exchange t by the $u \neq s$ with minimal probability without modifying the value of the average length. □

Proof of Theorem 4.2.4. Let $S = \{s_1, \dots, s_n\}$. For $n = 1, 2$ the result is trivial because $s_1 \mapsto 0$ and $s_1 \mapsto 0$, $s_2 \mapsto 1$ are optimal codes in these cases. We proceed by induction on n. We assume the result for $n-1$ and if $C : S \longrightarrow B^*$ gives a minimal value of ℓ_C we want to prove $\ell_C = \ell_H$ with $H : S \longrightarrow B^*$ the Huffman coding.

By Lemma 4.2.5 we can assume that s_1 and s_2 are siblings with $p_1 \le p_2 \le p_j$ for $j > 2$. Let $S' = \{z, s_3, \dots, s_n\}$ where z is the parent of s_1 and s_2 and we assign to it the probability $p_1 + p_2$. The code C induces a code $C : S' \longrightarrow B^*$ where $C'(z)$ is the codification of z in the tree of C. Clearly $\ell(C'(z)) = \ell(C(s_1)) - 1 = \ell(C(s_2)) - 1$, hence

$$\ell_C = p_1\ell(C(s_1)) + p_2\ell(C(s_2)) + \ell_{C'} - (p_1 + p_2)\ell(C'(z)) = p_1 + p_2 + \ell_{C'}. \tag{4.29}$$

On the other hand the definition of the Huffman coding implies that the smallest probability elements team up and we get the analogous formula $\ell_H = p_1 + p_2 + \ell_{H'}$ where H' is a Huffman encoding for S'. Subtracting both identities

$$\ell_C - \ell_H = \ell_{C'} - \ell_{H'}. \tag{4.30}$$

By the induction hypothesis the right hand side is nonpositive and the left hand side is nonnegative because ℓ_C is minimum. Hence $\ell_C = \ell_H$. □

Suggested Readings. The Huffman encoding and its optimal property is explained in a very elementary way for general audiences in [Rén84] and [Ham80]. For a more "academic" presentation see for instance [GP91] and [Rom92].

4.2.3 Data compression

Let me describe an experiment that you can easily reproduce. I have downloaded a version of "Hamlet" in plain text and its size is 191726 bytes, which is also the number of characters. The number of different symbols is 67 (note that there are capital letters, punctuation and spaces) and studying their frequencies the entropy is around 4.371. The source coding theorem suggests that this should be the minimal rate of bits per character using generic text compressors and with them the text could not be compressed below 104754 bytes (recall that each byte has 8 bits). The resulting size in bytes after applying some common data compressors was:

bzip2	gzip	zip	rar
57996	73816	69783	70756

Something is rotten in the state of Denmark and also in our argument because in the less favorable case we improve in almost a 30% the "optimal" estimate.

In part we have seen this situation before when we learned that grouping characters in blocks we can improve coding. If we focus in an English text, the words are close to be semantic units and we expect quite a number of repetitions of the same blocks with different length representing words.

People with low proficiency in English (like this author) will feel like idiots after reading claims as the one quoted in [Sto88, §1.6] (in [Sto88, §3.3] there is a more moderate claim): "the 50 most common English word types make up 45 percent of those written and 60 percent of those spoken". Does it mean that learning just 50 English words we will understand the 60% of a conversation? Of course not but it implies that shorthand can be very successful and with our silicon friends or servants we can devise a new close to optimal shorthand.

The problem of the entropy of English written texts was addressed by Shannon himself [Sha51] (see also [Sto88, §1.3]). With his primitive calculations, he noticed that the word "the" is the most common in English and its average frequency in a text is more than 7% and there is a "strong tendency of H to follow T", reflecting a lack of independence between consecutive letters. In [GP91, §2.2] it is mentioned that probably with a rate of 1.3 bits per letter one can encode any text in English.

The problem with this kind of linguistic approach is that does not match with in our globalized world, if you change the language you will need new software and still something else to compress data files. The ideal situation would be to develop an algorithm detecting dynamically what are the "words" composed by a variable number of symbols that introduce the redundancy. This ideal situation has been achieved in a theoretical setting and it performs fairly good in practice.

Although the proclaimed generality, it is convenient to have in mind texts to acquire some intuition and in the sequel we consider a file as a string of characters finishing with and EOF (end-of-file) character that we denote `#`. In ASCII files the characters correspond to bytes.

The origin of the modern compression software is the paper [ZL77] that introduced what is called today the LZ77 *algorithm* and its theoretical analysis. This algorithm considers a virtual *sliding window* that moves along the file that we want to compress. The left part of this window, which in practice is the largest by far, contains the past characters and it is called the *search buffer* because we will look for coincidences starting there. The right part of the window contains the future characters and it is called the *look-ahead buffer*. In each step of the algorithm we look for the longest match between the initial characters of the look-ahead buffer and a string starting at any point of the search buffer. The outgoing encoding is the triple given by the offset (the difference between their positions), the length of the match and the first non matching character in the look-ahead buffer. If there are not coincidences, the offset and the length are considered to be zero. After finished each step the window is moved to the right the last length plus one positions.

For example, let us say that the sliding window is of length 11 with a search buffer of size 7 and a look-ahead buffer of size 4. Consider a file that contains only the sentence `ban␣bananas!!!` where ␣ is to indicate a space. We start with an empty search buffer and during 4 steps the coding will be boring because the characters are new, there are not repetitions:

$$\begin{array}{|c|c|c|c|c|c|c||c|c|c|c|}\hline & & & & & & & \mathtt{b} & \mathtt{a} & \mathtt{n} & \text{␣} \\ \hline \end{array} \to (0,0,\mathtt{b})$$

$$\begin{array}{|c|c|c|c|c|c|c||c|c|c|c|}\hline & & & & & & \mathtt{b} & \mathtt{a} & \mathtt{n} & \text{␣} & \mathtt{b} \\ \hline \end{array} \to (0,0,\mathtt{a})$$

$$\begin{array}{|c|c|c|c|c|c|c||c|c|c|c|}\hline & & & & & \mathtt{b} & \mathtt{a} & \mathtt{n} & \text{␣} & \mathtt{b} & \mathtt{a} \\ \hline \end{array} \to (0,0,\mathtt{n})$$

$$\begin{array}{|c|c|c|c|c|c|c||c|c|c|c|}\hline & & & & \mathtt{b} & \mathtt{a} & \mathtt{n} & \text{␣} & \mathtt{b} & \mathtt{a} & \mathtt{n} \\ \hline \end{array} \to (0,0,\text{␣})$$

The next step is more interesting. We find in the look-ahead buffer `bana` and the string `ban` formed by the three first characters is in the search buffer 4 positions in the past. Then

$$\begin{array}{|c|c|c|c|c|c|c||c|c|c|c|}\hline & & & \mathtt{b} & \mathtt{a} & \mathtt{n} & \text{␣} & \mathtt{b} & \mathtt{a} & \mathtt{n} & \mathtt{a} \\ \hline \end{array} \to (4,3,\mathtt{a})$$

Now we have to move the sliding window four units to the right, the content of the look-ahead buffer becomes `nas!` and the starting string `na` gives a coincidence with offset 2. Hence the next encoding is

$$\begin{array}{|c|c|c|c|c|c|c||c|c|c|c|}\hline \mathtt{a} & \mathtt{n} & \text{␣} & \mathtt{b} & \mathtt{a} & \mathtt{n} & \mathtt{a} & \mathtt{n} & \mathtt{a} & \mathtt{s} & \mathtt{!} \\ \hline \end{array} \to (2,2,\mathtt{s})$$

In the next step we find the new character`!` and then again we have a trivial encoding

$$\begin{array}{|c|c|c|c|c|c|c||c|c|c|c|}\hline \mathtt{b} & \mathtt{a} & \mathtt{n} & \mathtt{a} & \mathtt{n} & \mathtt{a} & \mathtt{s} & \mathtt{!} & \mathtt{!} & \mathtt{!} & \mathtt{\#} \\ \hline \end{array} \to (0,0,\mathtt{!})$$

Now the content of the search buffer is `...s!` and the content of the look-ahead buffer is `!!#` and something funny occurs: we have a coincidence of length 2 with offset 1 that in part invades the look-ahead buffer. It is convenient to allow this possibility and the last step in the encoding is

$$\begin{array}{|c|c|c|c|c|c|c||c|c|c|c|}\hline \mathtt{a} & \mathtt{n} & \mathtt{a} & \mathtt{n} & \mathtt{a} & \mathtt{s} & \mathtt{!} & \mathtt{!} & \mathtt{!} & \mathtt{\#} & \\ \hline \end{array} \to (1,2,\mathtt{\#})$$

The LZ77 compression of the file would be the list of triples $(0,0,\mathtt{b})$, $(0,0,\mathtt{a})$, $(0,0,\mathtt{n})$, $(0,0,\text{␣})$, $(4,3,\mathtt{a})$, $(2,2,\mathtt{s})$, $(0,0,\mathtt{!})$, $(1,2,\mathtt{\#})$. For decompressing this list we have just to read each triple (a,b,c) as *"copy b of the previous characters starting a positions to the left and add the character c at the end"*. In our case

$$\begin{array}{cccccccc} (0,0,\mathtt{b}) & (0,0,\mathtt{a}) & (0,0,\mathtt{n}) & (0,0,\text{␣}) & (4,3,\mathtt{a}) & (2,2,\mathtt{s}) & (0,0,\mathtt{!}) & (1,2,\mathtt{\#}) \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ \mathtt{b} & \mathtt{a} & \mathtt{n} & \text{␣} & \mathtt{ban\,a} & \mathtt{na\,s} & \mathtt{!} & \mathtt{!!\,\#} \end{array}$$

Basic characters usually require 8 bits then, counting the EOF character, the original file would take $15 \cdot 8 = 120$ bits and for each triple we can encode the offset with 3 bits because it is a number between 0 and 7 and the length with 4 bits because the maximal length is 11. In total we have 15 bits per triple. Then we have not won anything in the compression and with a less repetitive sentence (like `ban␣tomatoes!!`) we would have indeed an apparent expansion. The reason is that it is a too short file. A longer text would give a good compression rate because the chance of repeated words or blocks increases.

As the matter of fact, in the actual compressing software the search buffer is about thousands of characters, or more properly bytes, and the look-ahead buffer about tens of characters [Sal07] then the "negative compression" of very small files is exaggerated. For instance, this example with `bzip`, `gzip`, `rar`, and `zip` gave respectively as size in bits of the compressed file 400, 264, 680 and 1496. In the latter case the "compressed" file is more than 10 times bigger than the original!

Two popular inheritors of LZ77 are LZ78 and LZW. They are quite similar among them and for the latter the decompression is sometimes less intuitive. Then we restrict the explanation to LZ78.

As before we consider a file as string of characters terminated by `#`. In LZ78 the movement of the sliding window, which does not exist in this algorithm, is substituted by a subdivision of the string into N sentences where each sentence is defined as a maximal substring that adds a new character to a previous sentence or to the empty set. In our example the subdivision into sentences is

1	2	3	4	5	6	7	8	9	10
`b`	`a`	`n`	␣	`ba`	`na`	`nas`	`!`	`!!`	`#`

then $N = 10$.

Let $s(n)$ be the n-th sentence and complete the definition with $s(0) = \emptyset$. Note that s is one to one, in the jargon this function is called the *dictionary*. Let $s^*(n)$ be the sentence $s(n)$ without the last character and $c(n)$ this last character. Then the encoding of the file with LZ78 is defined as the list of pairs

$$\Big\{\Big((s^{-1}\circ s^*)(n),\, c(n)\Big)\Big\}_{n=1}^{N}. \tag{4.31}$$

If $s(n)$ has only a character the first coordinate is 0. In our case we get the list $(0,\mathtt{b})$, $(0,\mathtt{a})$, $(0,\mathtt{n})$, $(0,␣)$, $(1,\mathtt{a})$, $(3,\mathtt{a})$, $(6,\mathtt{s})$, $(0,\mathtt{!})$, $(8,\mathtt{!})$, $(0,\mathtt{\#})$. It is easy to pass from (4.31) to the dictionary: In the n-th step to construct $s(n)$, one reads in the first coordinate the key of $s^*(n)$, which has already appeared, and once we have $s^*(n)$ we add the character $c(n)$ at the end to get $s(n)$.

In comparison to LZ77 the compressed file is composed by pairs instead of triples and this is good. In practice the file is divided into pieces to keep the number of bits required for the first coordinate under control. On the other hand, even in our example it is clear that there are more sentences, and hence pairs, in LZ78 than triples in LZ77 and this is bad[5]. Again the compression with LZ78 only applies to large files. For instance with the aforementioned text of Hamlet (191726 bytes) one obtains 35800 pairs with the first coordinate less than 35750. Then it can be encoded with 2 bytes, the list of pairs takes $35750 \cdot (2 + 1) = 107400$ bytes and we get a compression to the 56.01% of the original size. This could be improved with a better encoding (for instance Huffman). On the other hand `gzip` reduces the file to the 38.22% of the original size.

[5] On the internet one can find apparently informed people claiming that LZ77 is better than LZ78 and others claiming the opposite. The famous library `zlib` that is the base of many software compression implementations uses LZ77 but perhaps it is a matter of tradition or speed.

An important feature of the previous algorithms is that they are optimal “in the limit”. Let us prove a statement that makes rigorous this claim in the case of LZ78 (see [ZL77] for LZ77).

Theorem 4.2.6. *Consider the average length using* LZ78 *on strings $s = s_{i_1}s_{i_2}\cdots s_{i_N}$ of N symbols $s_{i_j}\in S$*

$$\ell_{\mathrm{LZ78},N} = \sum_{i_1,\dots i_N} p_{i_1}p_{i_2}\cdots p_{i_N}\ell_{\mathrm{LZ78}}(s_{i_1}s_{i_2}\cdots s_{i_N}) \tag{4.32}$$

where ℓ_{LZ78} is the length of the LZ78 *encoding of s. Then*

$$\lim_{N\to\infty}\frac{1}{N}\ell_{\mathrm{LZ78},N} = H \tag{4.33}$$

where H is the entropy of S.

Proof. Consider a string $s = s_{i_1}s_{i_2}\cdots s_{i_N}$, subdivide it into sentences according to LZ78 and for each i define

$$\mathcal{F}_i = \{\text{sentences of length } i\} \qquad \text{and} \qquad f_i = \#\mathcal{F}_i. \tag{4.34}$$

We have the *Ziv inequality*

$$\mathrm{Prob}(s) = p_{i_1}p_{i_2}\cdots p_{i_N} = \prod_i\prod_{f\in\mathcal{F}_i}\mathrm{Prob}(f) \le \prod_i\Big(\frac{1}{f_i}\sum_{f\in\mathcal{F}_i}\mathrm{Prob}(f)\Big)^{f_i} \le \prod_i f_i^{-f_i}, \tag{4.35}$$

where the first inequality follows from that of the arithmetic and geometric means and the second equality because the probability of any subset is at most one.

On the other hand, with analytical techniques (e.g. Lagrange multipliers, recall the proof of Theorem 4.2.1) or more elementary arguments it can be proved that

$$\sum x_i\log_2 x_i \ge -\log_2(\mu+1)-2 \quad \text{for any } x_i\in\mathbb{R}^+ \text{ with } \sum x_i = 1,\ \sum ix_i = \mu. \tag{4.36}$$

Let us choose $x_i = f_i/F$ with $F = \sum f_i$. As $\sum if_i$ is the sum of the lengths of the sentences, it equals N. Hence $\mu = N/F$ and this inequality adding $\log_2 F$ reads

$$\frac{1}{F}\sum_i f_i\log_2 f_i \ge \log_2 F - \log_2(N/F+1) - 2. \tag{4.37}$$

Using (4.35) we obtain

$$-\log_2\big(\mathrm{Prob}(s)\big) \ge F\log_2 F - F\log_2(N/F+1) - 2F. \tag{4.38}$$

Let us say that b is the number of bits needed to encode the symbols in S. Each pair in the output of LZ78 requires a number between 0 and F and a symbol, then we have the formula $\ell_{\mathrm{LZ78}}(s) = (B+b)F$, with B the smallest integer greater that $\log_2 F$, which allows to write the previous inequality as

$$-\log_2\big(\mathrm{Prob}(s)\big) \ge \ell_{\mathrm{LZ78}}(s) - F\log_2(N/F+1) - (b+3)F. \tag{4.39}$$

As $F = \sum f_i$ and $N = \sum i f_i$ with f_i nonnegative integers, F/N is arbitrarily small when $N \to \infty$ and we can find $g(N)$ independent of s with $g(N) \to 0$ such that the last two terms are bounded by $Ng(N)$. Multiplying by $N^{-1}\text{Prob}(s)$ we deduce

$$-\frac{1}{N}\text{Prob}(s)\log_2\left(\text{Prob}(s)\right) \ge \frac{1}{N}\text{Prob}(s)\ell_{\text{LZ78}}(s) - g(N)\text{Prob}(s). \tag{4.40}$$

When we sum over all possible strings s of length N and take the limit $N \to \infty$ the left hand side is H and the right hand side is $N^{-1}\ell_{\text{LZ78},N}$. The entropy bound is optimal, then this is enough to finish the proof. □

Regardless the theoretical interest of Theorem 4.2.6, the convergence is slow and in practice the results are far for being optimal. For instance, if we construct a file of length L containing a random sequence of characters of 1 byte `A` and `B` with the same probability the encoding $A \to 0$, $B \to 1$ is optimal (a random sequence cannot compressed by definition of "random") and reduces the size to $L/8$ so for 10^n we should get $125 \cdot 10^{n-3}$ bytes but the compression software mentioned before gives the following results for $2 \le n \le 8$

	10^2	10^3	10^4	10^5	10^6	10^7	10^8
`bzip`	60	227	1688	16117	160233	1601003	16011273
`gzip`	56	221	1674	15967	159077	1589446	15895727
`zip`	210	375	1828	16121	159231	1589600	15895881
`rar`	120	288	1763	16305	161447	1613430	16135477

The last column corresponds to files of 100 MB which is a respectable size in our computer especially if it corresponds to a text and still the space saving is less that 84.11% while it would converge to 87.5%.

Suggested Readings. The original paper [ZL77] is a little bit theoretical. In [Sal02], and with more details in [Sal07], there are descriptions of the basic algorithms LZ77, LZ78, LZW. The latter was involved in a famous controversy because its patent did not expired completely until 2004 and it could endanger the use of GIF files. In part PNG, which employs LZ77, was created to avoid this problem. By the way, for multimedia more than the lossless compression that we have seen here it is important to consider lossy algorithms in which part the information is lost but the compression rate is higher. An example is JPEG (see §2.2.2) and another more complicated is MP3.

Complementary material

There are a number of examples and computer programs freely available in

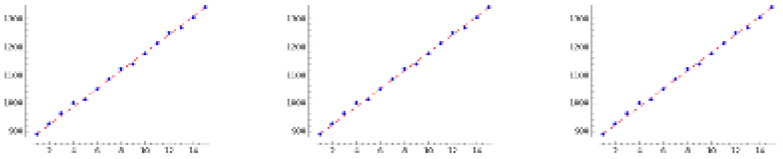
`http://matematicas.uam.es/~fernando.chamizo/dark/dark.html`

This material is a relevant help for this course. For instance, some images in the electronic or printed version can show a lack of quality and one can find in the posted complementary material full size versions of some of them. There are also variants and extensions of the examples discussed here and others completely new. On the other hand, the interested reader will be able of experimenting with the usually brief code. Taking into account the amount of text, images and code, more than 30 Mb, this material can constitute the starting point for many class projects.

Some caveats: Note that the aforementioned web page also contains material that does not belong to this course. No especial care was put into write elegant optimized code because the target was to give a quick small scale illustration of some topics. The label “matlab/octave” is employed when it has been checked under both systems. If not, it only appears the employed software (the code could also run under the alternative though). Especially for plotting, the computer algebra system sagemath is employed quite often. It can be freely downloaded or used online in `https://www.sagemath.org`.

Here it is a description of the items composing the complementary material. The ordering is alphabetical on the titles given to the items in the web page. Some thumbnails are provided to give a hint about the contents. They are always reduced versions of images included in the corresponding web page.

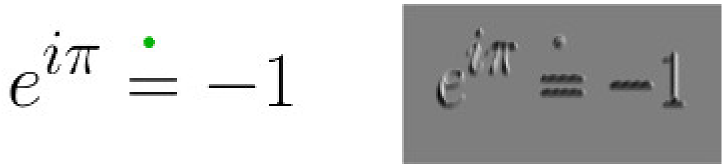
Title: **Anti-compression.** `d_anti_comp.html`
Thumbnails:



Related to: §4.2.3. Material: Graphs and code (python, sagemath and shell script).

Content: A successive application of the compressor `gzip` to a randomly generated file reveals a linear growth of the size of the output. The compressor becomes an expander after the first iteration.

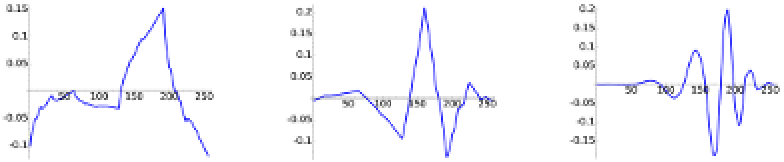
Title: **Bump map from image**. `d_bmap.html`
Thumbnails:



Related to: §2.2.4. Material: Images and code (matlab/octave).

Content: If we apply a blur linear filter to a binary and we represent the levels of gray as a third coordinate, we get a 3D version of the image. From a point of view close to the Z axis the effect is like a bas-relief. If we apply a high frequency oscillatory perturbation, the result is more appealing. In the example this is obtained by multiplication by $1 + 0.1 \sin(400xy)$.

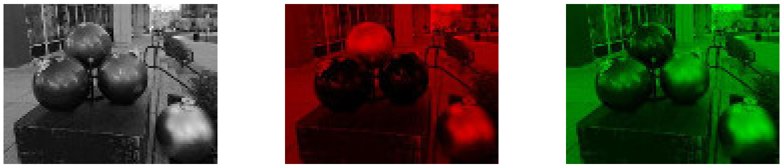
Title: **The cascade algorithm**. `d_cascade.html`
Thumbnails:



Related to: §3.2.1. Material: Images and code (octave, sagemath).

Content: Here it is employed the linear algebra interpretation of the cascade algorithm to plot approximations of the Daubechies wavelets. Namely, the entries of first rows approximate values of the scaling function and the rest approximations of the wavelets (up to shifting and scaling).

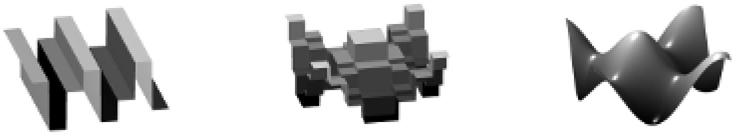
Title: **Color spaces**. `d_colors.html`
Thumbnails:



Related to: §2.2.2. Material: Images and code (octave).

Content: It is shown the role of the different channels in the color spaces RGB and $\mathrm{YC_bC_r}$ (employed internally by JPEG) applying a heavy quantization in each of them and comparing to the original test image.

Title: **DCT-II basis functions**. `d_basis.html`
Thumbnails:

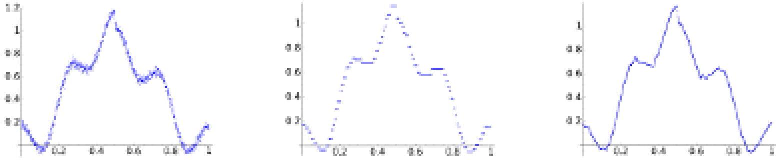


Related to: §2.2.1. Material: Images and code (matlab).

Content: This is plot of some of the basis function appearing in the DCT. Namely $\varphi_{kl}(x,y)=\varphi_k(x)\varphi_l(y)$ with $\varphi_k(x)=\cos(\pi k(x+1/2)/N)$. In the DCT, x and y only take integer values and then φ_{kl} can be visualized as a step surface. In the limit when $N\to\infty$, the quotient $(x+1/2)/N$ mimics a continuous variable and the surface becomes smooth.

Title: **Denoising with wavelets**. `d_denoisingw.html`

Thumbnails:

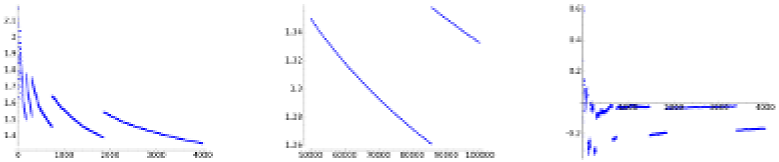


Related to: §3.2.2. Material: Images and code (octave).

Content: This is the application of the denoising method corresponding to take as threshold in the DWT an estimate (via a robust estimator) of the standard deviation of the Gaussian noise multiplied by $\sqrt{2\log N}$. The code included here plots the denoised signal and the error for the DWT corresponding to D2 (Haar), D4, D8 and D12.

Title: **Dictionary methods**. `d_dictionary.html`

Thumbnails:

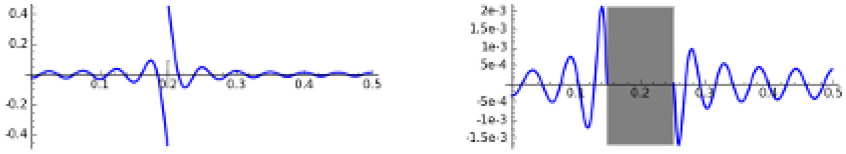


Related to: §4.2.3. Material: Graphs and code (sagemath).

Content: The compression algorithms LZ77, LZ78 and LZW are universal, they theoretically tend to the entropy bound. The general software for compression based on them gives result typically not so close to this optimal bound but one can blame some changes in the implementation. Here we consider the pure LZ78 and LZW algorithms applied to a random file and it is checked that the convergence is actually very slow. There is a band structure in the number of bits per character that remains unexplained.

Title: **Digital low-pass filters**. `d_lowpass.html`

Thumbnails:

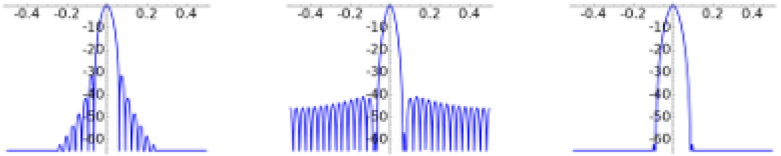


Related to: §2.2.3. Material: Graphs and code (sagemath).

Content: Some graphs are presented to show the absolute error in the passband and the stopband when constructing a digital low-pass filter based on the rectangular, Hann, Hamming and Blackman windows. It is also checked the accuracy of the Kaiser windows according to its parameters in an example.

Title: **Digital windows**. `d_dig_win.html`

Thumbnails:

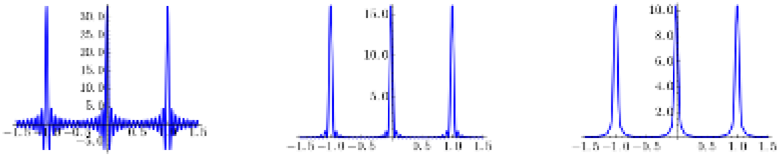


Related to: §2.2.3. Material: Graphs and code (sagemath).

Content: In some way the windows employed in digital signal processing try to imitate the behavior of a Dirac delta with a trigonometric polynomial. When its absolute value is represented in decibels, one aims for a thin main lobe and very small side lobes. The graphs show the plots corresponding to the rectangular, triangular, Hann, Hamming and Blackman windows.

Title: **Dirichlet, Fejér and more**. `d_dir_fej.html`

Thumbnails:

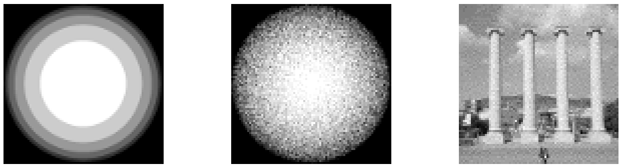


Related to: §1.2.1, §2.2.3. Material: Graphs and code (sagemath).

Content: The Dirichlet kernel and the Fejér kernel can be seen as the two first instances of applying Cesáro summation (C, k) to the sequence $e(nx)$. The graphs here are to illustrate that for $k = 2$ (or greater) we obtain functions that are in some way smoother but have thicker peaks. This is related to "Digital windows".

Title: **Dithering**. `d_dither.html`

Thumbnails:



Related to: §2.1.4. Material: Images and code (matlab/octave).

Content: With a geometric example it is shown that dithering is necessary and that introducing random noise before quantization is a cheap way to proceed although the results are not very impressive. With a photo it is shown the effect of the Bayer matrices and of the Floyd-Steinberg algorithm.

Title: **Edge detectors**. `d_edge.html`
Thumbnails:

Related to: §2.2.5. Material: Images and code (octave).

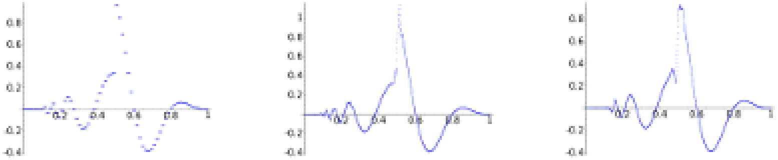
Content: They are shown examples of edge detecting filters. The simplest ones are discrete versions of the partial derivatives and of the Laplace operator and they give in general poor results. It is also considered the Sobel filter with some variants in the presentation of the resulting image and finally it is shown the effect of the Canny edge detector as implemented in octave.

Title: **Edges with wavelets**. `d_edge_wav.html`
Thumbnails:

Related to: §3.2.2. Material: Images and code (octave).

Content: It is shown that when we cancel the left upper corner of the 2D finite wavelet transform employed for images, then we obtain a primitive edge detector. If we take a thin corner i.e, if we set few elements to zero, the result gives to the example an unexpected metallic aspect.

Title: **The finite wavelet expansion**. `d_fin_wav.html`
Thumbnails:



Related to: §3.2.1, §3.2.2. Material: Graphs and code (octave and sagemath).

Content: A finite discrete signal with a discontinuity and oscillations of a wide range of frequencies is analyzed in terms of finite wavelets. Namely, they are employed the Haar wavelet and the finite Daubechies wavelets corresponding to D4 and D8. The plots show the approximations when the coefficients embodying the finer details are set to zero. This is the analogue of considering partial sums in the classical theory of Fourier series.

Title: **A fundamental pinch**. `d_fund_sol.html`

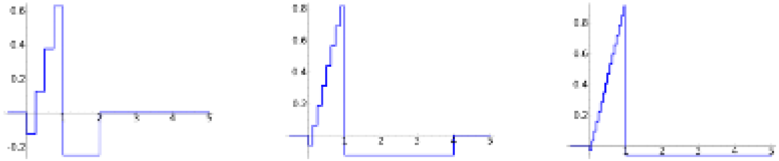
Thumbnails:

Related to: §1.2.5. Material: Images and code (matlab).

Content: It is considered the evolution of a large pinch on a sphere ruled by the wave equation. The relation with this course is that to study this evolution one needs to analyze the function into eigenfunctions of the Laplace-Beltrami operator. We take as a basis of these eigenfunctions the real spherical harmonics.

Title: **A Haar wavelet expansion**. `d_haar.html`

Thumbnails:

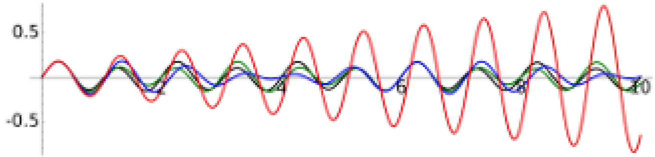


Related to: §3.1.2. Material: Graphs and code (sagemath).

Content: Consider the real function $f(x) = x$ on $[0, 1)$ and zero otherwise. If we analyze it in terms of the Haar wavelet we will get successive approximations by step functions. The plots show these step function and one has a visual idea about the convergence in L^2 but not in L^1.

Title: **Harmonic oscillations**. `d_har_osc.html`

Thumbnails:

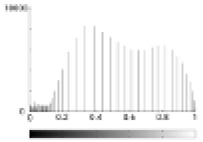


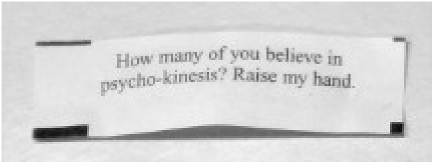
Related to: §1.1.1. Material: Graphs and code (sagemath).

Content: Some examples showing pure harmonic oscillations, the effect of the friction, the resonance phenomenon and the steady-state solution.

Title: **Histogram manipulation**. `d_histogram.html`

Thumbnails:





How many of you believe in psycho-kinesis? Raise my hand.

Related to: §2.2.5. Material: Images and code (octave).

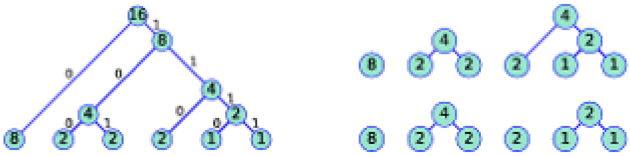
Content: Here they are applied three methods on two test images that have implications on their histograms. The first is the histogram equalization, the second is cropping the levels and stretching its distribution to have a more evenly distributed histogram and the third is a more complicated logarithmic model.

Title: **Homomorphic filtering**. `d_homom.html`
Thumbnails:

Related to: §2.2.4. Material: Images and code (octave, sagemath).

Content: This is the application of homomorphic filtering to enhance a photo not evenly illuminated. There are examples with several choices of the parameters and a comparison with the results obtained with ideal high-pass filters.

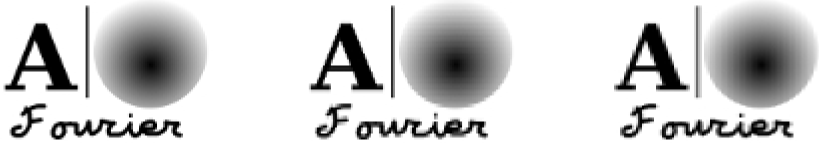
Title: **Huffman trees**. `d_huffman.html`
Thumbnails:



Related to: §4.2.2. Material: Images and code (sagemath).

Content: With some lines of code one can get the Huffman coding of a finite space of probability represented by a list of frequencies or probabilities. With a little more effort it is also possible also to plot the corresponding tree and any intermediate forest (the set of subtrees at a certain step).

Title: **JPEG**. `d_jpeg2.html`
Thumbnails:



Related to: §2.2.2. Material: Images and code (matlab).

Content: On a test image it is shown the effect of canceling some of the DCT coefficients appearing in the JPEG format. In part this is an old version of “Masking the JPEG Fourier coefficients”.

Title: **The Kuwahara filter**. `d_kuwa.html`

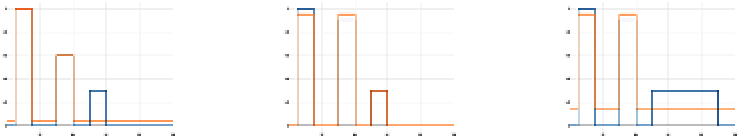
Thumbnails:

Related to: §2.2.5. Material: Images and code (octave, sagemath).

Content: Some examples are shown illustrating the application of the Kuwahara filter for noise reduction and to create an appealing artistic effect.

Title: **The Lloyd-Max algorithm**. `d_lloyd.html`

Thumbnails:

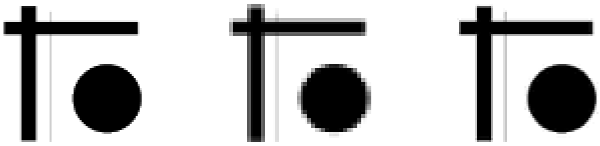


Related to: §2.1.2. Material: Graphs and code (matlab).

Content: Here they are imposed three levels of quantization and the mean squared quantization error minimization is approached with the Lloyd-Max algorithm as implemented in matlab. There are some examples giving weird results.

Title: **Masking the JPEG Fourier coefficients**. `d_jpeg3.html`

Thumbnails:

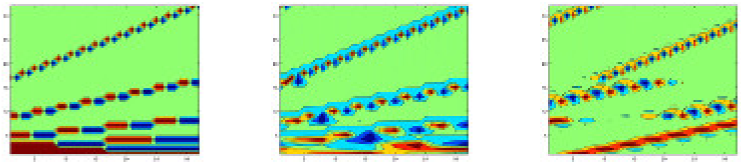


Related to: §2.2.2. Material: Images and code (matlab).

Content: It is shown the effect of canceling (masking) some of the DCT coefficients appearing of a JPEG image. The topic is the same addressed in “JPEG”, probably this version is more illustrative.

Title: **The matrix of the DWT**. `d_matrix_dwt.html`

Thumbnails:



Related to: §3.2.1. Material: Images and code (octave).

Content: It is provided a (not optimized) code to compute the matrix of the DWT. There are some examples with small dimensions for D2 (Haar), D4 and D12. To grasp the idea about the structure of the matrix and the connection with the cascade algorithm, it is considered the contour plot of the values of matrix for $N = 32$ with a certain normalization of the rows.

Title: **Mean vs. median**. `d_meamed.html`

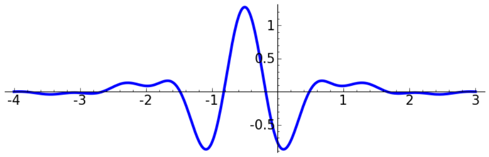
Thumbnails:

Related to: §2.2.5, §2.2.4. Material: Images and code (matlab).

Content: It shows how effective is the use of a median filter to clean an image affected by salt-and-pepper noise. A linear mean filter gives poor results. The noise is introduced on each color channel.

Title: **The Meyer wavelet**. `d_meyer.html`

Thumbnails:

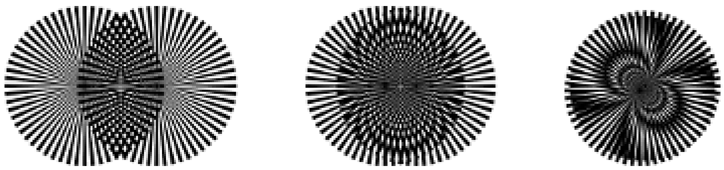


Related to: §3.1.3. Material: Graphs and code (sagemath).

Content: It is depicted the Meyer wavelet and some basis elements of the corresponding multiresolution analysis. To illustrate that they form really a basis, the Mexican hat function is analyzed in terms of them. When the number of harmonics grows, we get a quite good approximation.

Title: **Moiré effect**. `d_moire.html`

Thumbnails:

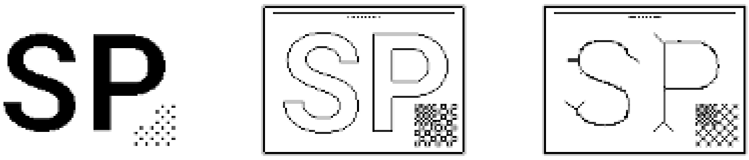


Related to: §2.3 (challenge "Moiré, qu'est-ce que c'est?"). Material: Images.

Content: Overlapping two equal circles divided into thin circular sectors alternating black and transparent colors, we see something very close to the contour plot of the electrostatic potential of a point dipole.

Title: **Morphological filters**. `d_morpho.html`

Thumbnails:

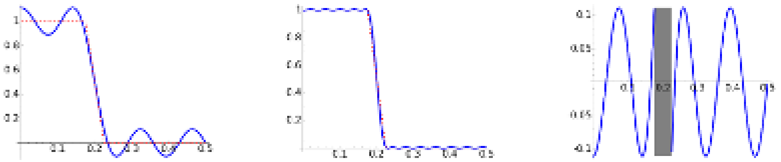


Related to: §2.2.5. Material: Images and code (octave).

Content: It is shown the effect of many morphological filters as implemented in octave on two test images. The examples correspond to almost all the possible morphological operations that can be done with octave.

Title: **The Parks-McClellan algorithm**. `d_pa_mc.html`

Thumbnails:



Related to: §2.2.3. Material: Images and code (matlab, sagemath).

Content: There are three examples of constructing an optimal low-pass filter using the Parks-McClellan algorithm as implemented in matlab. It is plotted the comparison with the ideal filter and the error.

Title: **Pencil line drawing effect**. `d_pencil.html`

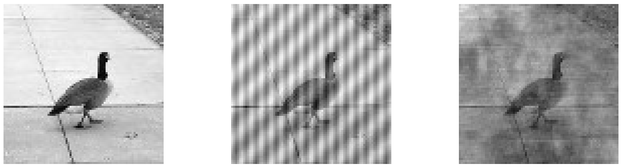
Thumbnails:

Related to: §2.2.4. Material: Images and code (matlab).

Content: The Laplacian filter is successively modified to get pencil drawing effect on a photo. The challenge here is to use only linear filters and clamping. The result is not bad taking into account the simplicity of the filter.

Title: **Periodic noise**. `d_periodic.html`

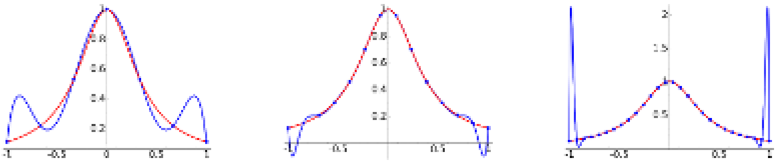
Thumbnails:



Related to: §2.2.4. Material: Images and code (octave).

Content: It is shown how to reconstruct an image affected by periodic noise substituting the isolated peaks in the DCT by values inferred from the behavior in a neighborhood. To detect the peaks it is compared each value of the DCT with the median of the surrounding values.

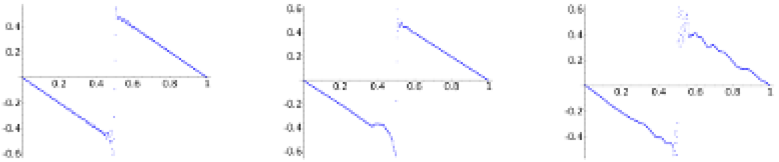
Title: **Runge's phenomenon**. `d_runge.html`
Thumbnails:



Related to: §2.1.3. Material: Graphs and code (sagemath).

Content: It is plotted the interpolation polynomial with evenly spaced nodes in $[-1,1]$ for the functions $\sin(\pi x)$, $\sin(5\pi x)$ and $(8x^2+1)^{-1}$. The interpolation error for latter function increases when the number of nodes is large due to some peaks in the interpolation polynomial. This phenomenon does not apply to $\sin(\pi x)$ while $\sin(5\pi x)$ only produces the peaks in a transitory range.

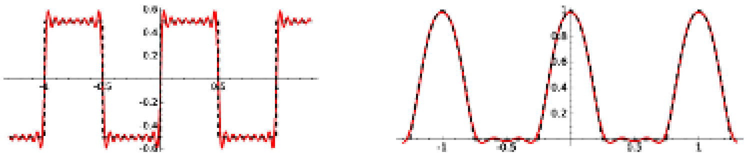
Title: **Signal compression**. `d_scomp.html`
Thumbnails:



Related to: §3.2.2. Material: Graphs and code (octave, sagemath).

Content: A method to compress a signal is to store its transform setting to zero the values under certain threshold. This threshold is chosen in such a way that the energy loss is very small. The reconstruction is achieved applying the inverse transform. This procedure is completed here for a function with a single jump singularity using the DCT and the DWT with the filter corresponding to D4 and D8. The DWT produces more zeros than the DCT allowing more compression.

Title: **Simple Fourier series**. `d_sim_fou.html`
Thumbnails:



Related to: §1.2.1. Material: Graphs and code (sagemath).

Content: The square wave, the sawtooth wave, the triangle wave and the rectified cosine are very simple waves having explicit Fourier series. Here some partial sums are plotted.

The triangle wave and the rectified cosine are more regular (continuous and piecewise differentiable) and even very few terms give approximations coinciding with the function to the naked eye. The error term is plotted in some of theses cases.

Title: **Simple image filters**. `d_simimf.html`

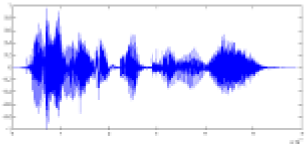
Thumbnails:

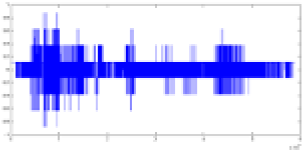
Related to: §2.2.4. Material: Images and code (octave).

Content: This is to illustrate several linear filters given by a finite convolution represented by a matrix. The instances discussed are the sharpen, Laplace and emboss filters and different forms of blurring.

Title: **Sound quantization**. `d_sound.html`

Thumbnails:



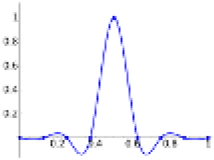


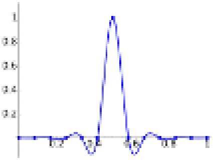
Related to: §2.1.2, §1.1.3. Material: Graphs, audio (flac) and code (matlab/octave).

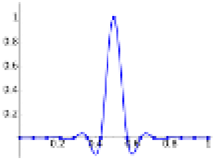
Content: It is shown the quantization of a pure sine tone and of a sample of voice. Something a little surprising is that even with only two levels of quantization, the voice is still understandable. To put it in other terms, it seems that the most of the voice information lies in the frequencies.

Title: **Splines are nearly local**. `d_lsplines.html`

Thumbnails:





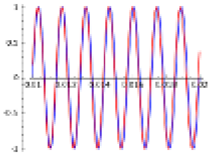


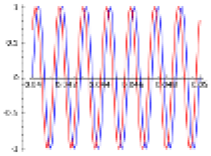
Related to: §2.1.3. Material: Graphs and code (sagemath).

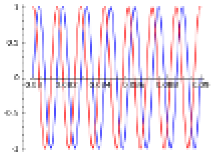
Content: It is plotted the cubic spline interpolation for a functions that takes the value 1 at the central node and zero at the rest. The result is quite localized, which can be rephrased saying that cubic splines do not differ that much from B-splines.

Title: **The strange beat**. `d_stra_beat.html`

Thumbnails:





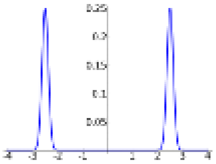


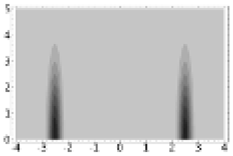
Related to: §1.3 (homonymous challenge). Material: Graphs and code (sagemath).

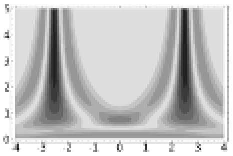
Content: In an audio file it is shown the effect of the superposition of two sine waves of very close frequencies. The result may result a little strange and deserves a mathematical explanation.

Title: **Windowed transforms**. `d_wt.html`

Thumbnails:







Related to: §3.1.1. Material: Images and code (sagemath).

Content: The function $f(x) = \big(e^{-32(x-5/2)^2} + e^{-32(x+5/2)^2}\big)/4$ represents two peaks of height $1/4$ at $x = \pm 5/2$. Two windowed transforms are considered: the Gabor transform and the wavelet transform with the Mexican hat wavelet. In this last case it is also shown the contribution of different ranges of a.

Bibliography

[AF67] M. Alonso and E. J. Finn. *Fundamental University Physics: Fields and waves.* Addison-Wesley series in physics. Addison-Wesley, 1967.

[Ahl78] L. V. Ahlfors. *Complex analysis.* McGraw-Hill Book Co., New York, third edition, 1978. An introduction to the theory of analytic functions of one complex variable, International Series in Pure and Applied Mathematics.

[ANR74] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Trans. Computers*, C-23:90–93, 1974.

[Atk89] K. E. Atkinson. *An introduction to numerical analysis.* John Wiley & Sons, Inc., New York, second edition, 1989.

[Aus08] D. Austin. What is... JPEG? *Notices Amer. Math. Soc.*, 55(2):226–229, 2008.

[Bae05] J. Baez. The pendulum, elliptic functions and imaginary time. `http://math.ucr.edu/home/baez/classical/pendulum.pdf`, 2005.

[BB05] G. L. Baker and J. A. Blackburn. *The pendulum.* Oxford University Press, Oxford, 2005. A case study in physics.

[BCT91] U. Baum, M. Clausen, and B. Tietz. Improved upper complexity bounds for the discrete Fourier transform. *Appl. Algebra Engrg. Comm. Comput.*, 2(1):35–43, 1991.

[Bla10] R. E. Blahut. *Fast algorithms for signal processing.* Cambridge University Press, Cambridge, 2010.

[Blu70] L. I. Bluestein. A linear filtering approach to the computation of discrete Fourier transform. *IEEE Transactions on Audio and Electroacoustics*, 18(4):451–455, 1970.

[Bov09] A. C. Bovik. *The Essential Guide to Image Processing.* Academic Press, Inc., 2009.

[Bra86] D. Braess. *Nonlinear approximation theory*, volume 7 of *Springer Series in Computational Mathematics.* Springer-Verlag, Berlin, 1986.

[Bré02] P. Brémaud. *Mathematical principles of signal processing.* Springer-Verlag, New York, 2002. Fourier and wavelet analysis.

[Bus03] S. R. Buss. *3-D computer graphics.* Cambridge University Press, Cambridge, 2003. A mathematical introduction with OpenGL.

[Byr04] C. Byrne. A unified treatment of some iterative algorithms in signal processing and image reconstruction. *Inverse Problems*, 20(1):103–120, 2004.

[Byr15] C. L. Byrne. *Signal processing.* Monographs and Research Notes in Mathematics. CRC Press, Boca Raton, FL, second edition, 2015. A mathematical approach.

[Can86] J. F. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell*, PAMI-8:679–698, 12 1986.

[Car66] L. Carleson. On convergence and growth of partial sums of Fourier series. *Acta Math.*, 116:135–157, 1966.

[CCI91] CCITT. *Digital Compression and Coding of Continuous-Tone Still Images, Part 1, Requirements and Guidelines.* ISO/IEC JTC1 Committee Draft 10918-1, 1991.

[CH53] R. Courant and D. Hilbert. *Methods of mathematical physics. Vol. I.* Interscience Publishers, Inc., New York, N.Y., 1953.

[Cha11] L. Chaparro. *Signals and Systems using MATLAB.* Elsevier Science, 2011.

[Chu92] C. K. Chui. *An introduction to wavelets*, volume 1 of *Wavelet Analysis and its Applications.* Academic Press, Inc., Boston, MA, 1992.

[Coh17] H. Cohn. A conceptual breakthrough in sphere packing. *Notices Amer. Math. Soc.*, 64(2):102–115, 2017.

[Cór08] A. Córdoba. La tesis de riemann sobre las series trigonométricas. In *Conferències FME, Volum V: Curs Riemann (2007/2008)*, pages 223–246. Facultat de Matemàtiques i Estadística, UPC, 2008.

[CR17] F. Chamizo and D. Raboso. La fórmula de sumación de Poisson y parientes cercanos. *Materials matemàtics*, pages 1–27, 2017.

[CT65] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Comp.*, 19:297–301, 1965.

[Dau88] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 41(7):909–996, 1988.

[Dau92] I. Daubechies. *Ten lectures on wavelets*, volume 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.

[Dav75] P. J. Davis. *Interpolation and approximation.* Dover Publications, Inc., New York, 1975. Republication, with minor corrections, of the 1963 original, with a new preface and bibliography.

[DJ94] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[DL92] I. Daubechies and J. C. Lagarias. Two-scale difference equations. II. Local regularity, infinite products of matrices and fractals. *SIAM J. Math. Anal.*, 23(4):1031–1079, 1992.

[DM72] H. Dym and H. P. McKean. *Fourier series and integrals.* Academic Press, New York-London, 1972. Probability and Mathematical Statistics, No. 14.

[DS89] D. L. Donoho and P. B. Stark. Uncertainty principles and signal recovery. *SIAM J. Appl. Math.*, 49(3):906–931, 1989.

[Ein05] A. Einstein. Zur Elektrodynamik bewegter Körper. *Annalen der Physik*, 17:891–921, 1905.

[Fie03] S. Q. Field. *Gonzo Gizmos.* Chicago Review Press, Incorporated, 2003. Projects and Devices to Channel Your Inner Geek.

[FLS64] R. P. Feynman, R. B. Leighton, and M. Sands. *The Feynman lectures on physics. Vol. 2: Mainly electromagnetism and matter.* Addison-Wesley Publishing Co., Inc., Reading, Mass.-London, 1964.

[Fol92] G. B. Folland. *Fourier analysis and its applications.* The Wadsworth & Brooks/Cole Mathematics Series. Wadsworth & Brooks/Cole Advanced Books & Software, Pacific Grove, CA, 1992.

[Fol08] G. B. Folland. *Quantum field theory*, volume 149 of *Mathematical Surveys and Monographs.* American Mathematical Society, Providence, RI, 2008. A tourist guide for mathematicians.

[Fou88] J. Fourier. *Théorie analytique de la chaleur.* Éditions Jacques Gabay, Paris, 1988. Reprint of the 1822 original.

[Fra28] P. Franklin. A set of continuous orthogonal functions. *Math. Ann.*, 100(1):522–529, 1928.

[FS76] R. W. Floyd and L. Steinberg. An adaptive algorithm for spatial grey scale. *Proc. Soc. Inf. Display*, 17:75–77, 1976.

[Gal06] R. G. Gallager. Principles of digital communications I. MIT OpenCourseWare 6.450 `https://ocw.mit.edu/`, Fall 2006.

[Gar15] T. A. Garrity. *Electricity and magnetism for mathematicians.* Cambridge University Press, New York, 2015. A guided path from Maxwell's equations to Yang-Mills.

[Ger99] N. Gershenfeld. *The nature of mathematical modeling.* Cambridge University Press, Cambridge, 1999.

[GG12] A. García García. *Bases en espacios de Hilbert: teoría de muestreo y wavelets.* Sanz y Torres, 2012.

[GP90] A. Galindo and P. Pascual. *Quantum mechanics. I.* Texts and Monographs in Physics. Springer-Verlag, Berlin, 1990. Translated from the Spanish by J. D. García and L. Alvarez-Gaumé.

[GP91] C. M. Goldie and R. G. E. Pinch. *Communication theory*, volume 20 of *London Mathematical Society Student Texts.* Cambridge University Press, Cambridge, 1991.

[Gra08] L. Grafakos. *Classical Fourier analysis*, volume 249 of *Graduate Texts in Mathematics.* Springer, New York, second edition, 2008.

[Gui41] A. P. Guinand. On Poisson's summation formula. *Ann. of Math. (2)*, 42:591–603, 1941.

[GW08] R. C. Gonzalez and R. E. Woods. *Digital image processing.* Prentice-Hall, Inc., Upper Saddle River, NJ., third edition, 2008.

[GWE03] R. C. Gonzalez, R. E. Woods, and S. L. Eddins. *Digital Image Processing Using MATLAB.* Prentice-Hall, Inc., Upper Saddle River, NJ., 2003.

[Haa10] A. Haar. Zur Theorie der orthogonalen Funktionensysteme. *Math. Ann.*, 69(3):331–371, 1910.

[Ham80] R. W. Hamming. *Coding and information theory.* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1980.

[Ham89] R. W. Hamming. *Digital Filters.* Prentice Hall International (UK) Ltd., third edition, 1989.

[Har33] G. H. Hardy. A Theorem Concerning Fourier Transforms. *J. London Math. Soc.*, 8(3):227–231, 1933.

[Har78] F. J. Harris. On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 66(2):51–83, 1978.

[Hei27] W. Heisenberg. Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik. *Zeitschrift für Physik*, 43(3):172–198, 1927.

[Hel91] H. Helson. *Harmonic analysis.* The Wadsworth & Brooks/Cole Mathematics Series. Wadsworth & Brooks/Cole Advanced Books & Software, Pacific Grove, CA, 1991. Reprint of the 1983 original.

[Her90] H. Hertz. *Las ondas electromagnéticas.* Servei de Publicacions de la Universitat Autònoma de Barcelona, 1990.

[Her18] E. Hernández. Ondículas: historia, teoría y aplicación. *Gac. R. Soc. Mat. Esp.*, 21(2):275–299, 2018.

[HJB85] M. T. Heideman, D. H. Johnson, and C. S. Burrus. Gauss and the history of the fast Fourier transform. *Arch. Hist. Exact Sci.*, 34(3):265–277, 1985.

[Huf52] D. A. Huffman. A method for the construction of minimum-redundancy codes. *Proc. I.R.E.*, 40(9):1098–1101, 1952.

[Hun68] R. A. Hunt. On the convergence of Fourier series. In *Orthogonal Expansions and their Continuous Analogues (Proc. Conf., Edwardsville, Ill., 1967)*, pages 235–255. Southern Illinois Univ. Press, Carbondale, Ill., 1968.

[HW96] E. Hernández and G. Weiss. *A first course on wavelets.* Studies in Advanced Mathematics. CRC Press, Boca Raton, FL, 1996. With a foreword by Yves Meyer.

[HW06] C. Heil and D. F. Walnut, editors. *Fundamental papers in wavelet theory.* Princeton University Press, Princeton, NJ, 2006.

[Kai74] J. F. Kaiser. Nonrecursive digital filter design using the i_0-sinh window function. In *Proc. 1974 IEEE International Symposium on Circuits & Systems, San Francisco DA, April*, pages 20–23, 1974.

[Kai94] G. Kaiser. *A friendly guide to wavelets.* Birkhäuser Boston, Inc., Boston, MA, 1994.

[Kam07] D. W. Kammler. *A first course in Fourier analysis.* Cambridge University Press, Cambridge, second edition, 2007.

[Kat68] Y. Katznelson. *An introduction to harmonic analysis.* John Wiley & Sons, Inc., New York-London-Sydney, 1968.

[KC96] D. Kincaid and W. Cheney. *Numerical analysis.* Brooks/Cole Publishing Co., Pacific Grove, CA, second edition, 1996. Mathematics of scientific computing.

[KK66] J.-P. Kahane and Y. Katznelson. Sur les ensembles de divergence des séries trigonométriques. *Studia Math.*, 26:305–306, 1966.

[Kör88] T. W. Körner. *Fourier analysis.* Cambridge University Press, Cambridge, 1988.

[LL58] L. D. Landau and E. M. Lifshitz. *Quantum mechanics: non-relativistic theory. Course of Theoretical Physics, Vol. 3.* Addison-Wesley Series in Advanced Physics. Pergamon Press Ltd., London-Paris, 1958.

[Llo82] S. P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inform. Theory*, 28(2):129–137, 1982.

[LM86] P. G. Lemarié and Y. Meyer. Ondelettes et bases hilbertiennes. *Rev. Mat. Iberoamericana*, 2(1-2):1–18, 1986.

[LT00] M. Lacey and C. Thiele. A proof of boundedness of the Carleson operator. *Math. Res. Lett.*, 7(4):361–370, 2000.

[LW18] K. H. Lee and G. Weiss. Constructing an orthonormal wavelet from an MRA. arXiv:1503.04874 [math.CA], 2018.

[Mac03] D. J. C. MacKay. *Information theory, inference and learning algorithms.* Cambridge University Press, New York, 2003.

[Mal09] S. Mallat. *A wavelet tour of signal processing.* Elsevier/Academic Press, Amsterdam, third edition, 2009. The sparse way, With contributions from Gabriel Peyré.

[Mar91] R. J. Marks, II. *Introduction to Shannon sampling and interpolation theory.* Springer Texts in Electrical Engineering. Springer-Verlag, New York, 1991.

[Max65] J. C. Maxwell. A dynamical theory of the electromagnetic field. *Philosophical Transactions of the Royal Society of London*, 155:459–513, 1865.

[Max54] J. C. Maxwell. *A treatise on electricity and magnetism.* Dover Publications Inc., New York, 1954. 3d ed, Two volumes bound as one.

[Max60] J. Max. Quantization for minimum distortion. *IRE Trans. Inform. Theory*, 6(2):7–12, 1960.

[McM56] B. McMillan. Two inequalities implied by unique decipherability. *IEEE Trans. Information Theory*, 2(4):115–116, 1956.

[Mey87] Y. Meyer. Principe d'incertitude, bases hilbertiennes et algèbres d'opérateurs. *Astérisque*, 145-146:4, 209–223, 1987. Séminaire Bourbaki, Vol. 1985/86.

[MI11] D. G. Manolakis and V. K. Ingle. *Applied Digital Signal Processing.* Cambridge University Press, 2011. Theory and Practice.

[MR97] D. K. Maslen and D. N. Rockmore. Generalized FFTs—a survey of some recent results. In *Groups and computation, II (New Brunswick, NJ, 1995)*, volume 28 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 183–237. Amer. Math. Soc., Providence, RI, 1997.

[Mui60] T. Muir. *A treatise on the theory of determinants.* Revised and enlarged by W. H. Metzler. Dover Publications, Inc., New York, 1960.

[MVG16] A. del Mazo, S. Velasco, and R. García. *Oír y Ver: 61 experimentos de acústica y óptica.* Ediciones de la universidad de Murcia, 2016.

[NA08] M. Nixon and A. S. Aguado. *Feature Extraction & Image Processing.* Academic Press, Inc., second edition, 2008.

[New74] D. J. Newman. Fourier uniqueness via complex variables. *Amer. Math. Monthly*, 81:379–380, 1974.

[OSS68] A.V. Oppenheim, R.W. Schafer, and T.G. Jr. Stockham. Nonlinear filtering of multiplied and convolved signals. *Proceedings of the IEEE*, 56(8):1264–1291, 1968.

[Pap75] A. Papoulis. A new algorithm in spectral analysis and band-limited extrapolation. *IEEE Trans. Circuits and Systems*, CAS-22(9):735–742, 1975.

[PB72] A. Papoulis and M. S. Bertran. Digital filtering and prolate functions. *IEEE Trans. Circuit Theory*, CT-19:674–681, 1972.

[Pin02] M. A. Pinsky. *Introduction to Fourier analysis and wavelets.* Brooks/Cole Series in Advanced Mathematics. Brooks/Cole, Pacific Grove, CA, 2002.

[PM72] T. W. Parks and J. H. McClellan. Chebyshev approximation for nonrecursive digital filters with linear phase. *IEEE Trans. Circuit Theory*, 19:189–194, 1972.

[PM96] J. G. Proakis and D. G. Manolakis. *Digital Signal Processing: Principles, Algorithms, and Applications.* Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1996.

[PP10] M. Petrou and C. Petrou. *Image Processing: The Fundamentals.* John Wiley & Sons, Ltd, second edition, 2010.

[Rad68] C. M. Rader. Discrete fourier transforms when the number of data samples is prime. *Proceedings of the IEEE*, 56(6):1107–1108, 1968.

[RD05] P. Renteln and A. Dundes. Foolproof: A sampling of mathematical folk humor. *Notices Amer. Math. Soc.*, 52(1):24–34, 2005.

[Rén84] A. Rényi. *A diary on information theory.* Akadémiai Kiadó (Publishing House of the Hungarian Academy of Sciences), Budapest, 1984.

[Rom92] S. Roman. *Coding and information theory*, volume 134 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, 1992.

[RSR69] L. R. Rabiner, R. W. Schafer, and C. M. Rader. The chirp z-transform algorithm and its application. *Bell System Tech. J.*, 48(5):1249–1292, 1969.

[Rud74] W. Rudin. *Real and complex analysis.* McGraw-Hill Book Co., New York-Düsseldorf-Johannesburg, second edition, 1974. McGraw-Hill Series in Higher Mathematics.

[Rus07] J. C. Russ. *The image processing handbook.* CRC Press, Boca Raton, FL, sixth edition, 2007.

[RVF09] D. K. Ruch and P. J. Van Fleet. *Wavelet theory.* John Wiley & Sons, Inc., Hoboken, NJ, 2009. An elementary approach with applications.

[RW98] H. L. Resnikoff and R. O. Wells, Jr. *Wavelet analysis.* Springer-Verlag, New York, 1998. The scalable structure of information.

[Sal02] D. Salomon. *A guide to data compression methods.* Springer-Verlag, 2002.

[Sal07] D. Salomon. *Data compression.* Springer-Verlag London, Ltd., London, fourth edition, 2007. The complete reference, With contributions by G. Motta and D. Bryant.

[SB02] J. Stoer and R. Bulirsch. *Introduction to numerical analysis*, volume 12 of *Texts in Applied Mathematics.* Springer-Verlag, New York, third edition, 2002. Translated from the German by R. Bartels, W. Gautschi and C. Witzgall.

[Sha48] C. E. Shannon. A mathematical theory of communication. *Bell System Tech. J.*, 27:379–423, 623–656, 1948.

[Sha49] C. E. Shannon. Communication in the presence of noise. *Proc. I.R.E.*, 37:10–21, 1949.

[Sha51] C. E. Shannon. Prediction and entropy of printed English. *Bell System Tech. J.*, 30(1):50–64, 1951.

[Sim17] G. F. Simmons. *Differential equations with applications and historical notes.* Textbooks in Mathematics. CRC Press, Boca Raton, FL, 2017. Third edition.

[Sle78] D. Slepian. Prolate spheroidal wave functions, Fourier analysis and uncertainity. V. The discrete case. *Bell System Tech. J.*, 57(5):1371–1430, 1978.

[Sle83] D. Slepian. Some comments on Fourier analysis, uncertainty and modeling. *SIAM Rev.*, 25(3):379–393, 1983.

[Soi03] P. Soille. *Morphological Image Analysis: Principles and Applications.* Springer-Verlag, Berlin, Heidelberg, second edition, 2003.

[SR01] J. M. Sánchez-Ron. *Historia de la física cuántica: el periodo fundacional (1860–1926).* Crítica, 2001.

[SS03] E. M. Stein and R. Shakarchi. *Fourier analysis*, volume 1 of *Princeton Lectures in Analysis.* Princeton University Press, Princeton, NJ, 2003. An introduction.

[Ste16] I. Stewart. *Calculating the cosmos.* Basic Books, New York, 2016. How mathematics unveils the universe.

[Sto88] J. A. Storer. *Data Compression: Methods and Theory*, volume 13 of *Principles of Computer Science Series.* Computer Science Press, 1988.

[Str86] G. Strang. *Introduction to applied mathematics.* Wellesley-Cambridge Press, Wellesley, MA, 1986.

[Str99] G. Strang. The discrete cosine transform. *SIAM Rev.*, 41(1):135–147, 1999.

[Tao] T. Tao. Hardy's uncertainty principle. `https://terrytao.wordpress.com/2009/02/18/hardys-uncertainty-principle/`.

[Ter99] A. Terras. *Fourier analysis on finite groups and applications*, volume 43 of *London Mathematical Society Student Texts.* Cambridge University Press, Cambridge, 1999.

[TG15] A. Tomás and R. García. *Experimentos de física y química en tiempos de crisis.* Ediciones de la universidad de Murcia, 2015.

[Uli87] R. Ulichney. *Digital Halftoning.* MIT Press, Cambridge, MA, USA, 1987.

[VBM96] A. Verma, S. Bilbao, and T.H.Y. Meng. The digital prolate spheroidal window. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 3, pages 1351–1354, 1996.

[Vya09] A. Vyas. Construction of non-MSF non-MRA wavelets for $L^2(\mathbb{R})$ and $H^2(\mathbb{R})$ from MSF wavelets. *Bull. Pol. Acad. Sci. Math.*, 57(1):33–40, 2009.

[Wal91] G. K. Wallace. The JPEG still picture compression standard. *Commun. ACM*, 34(4):30–44, apr 1991.

[Wal96] J. S. Walker. *Fast Fourier transforms.* Studies in Advanced Mathematics. CRC Press, Boca Raton, FL, second edition, 1996.

[Wal08] J. S. Walker. *A primer on wavelets and their scientific applications.* Studies in Advanced Mathematics. Chapman & Hall/CRC, Boca Raton, FL, second edition, 2008.

[Whi15] E. T. Whittaker. XVIII.-on the functions which are represented by the expansions of the interpolation-theory. *Proc. Roy. Soc. Edinburgh Sect. A*, 35:181–194, 1915.

[Win78] S. Winograd. On computing the discrete Fourier transform. *Math. Comp.*, 32(141):175–199, 1978.

[Win02] S. Winder. *Analog and Digital Filter Design.* EDN Series for Design Engineers. Elsevier Science, 2002.

[Woj97] P. Wojtaszczyk. *A mathematical introduction to wavelets*, volume 37 of *London Mathematical Society Student Texts.* Cambridge University Press, Cambridge, 1997.

[Xia93] X. G. Xia. Extensions of the Papoulis-Gerchberg algorithm for analytic functions. *J. Math. Anal. Appl.*, 179(1):187–202, 1993.

[Yav68] R. Yavne. An economical method for calculating the discrete Fourier transform. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, pages 115–125, New York, NY, USA, 1968.

[You14] Y. You. *Audio Coding: Theory and Applications.* Springer Publishing Company, Incorporated, 2014.

[Zee16] A. Zee. *Group theory in a nutshell for physicists.* Princeton University Press, Princeton, NJ, 2016.

[ZL77] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Trans. Information Theory*, IT-23(3):337–343, 1977.

[Zwi13] B. Zwiebach. Quantum physics ii. `https://ocw.mit.edu/courses/physics/8-05-quantum-physics-ii-fall-2013/lecture-notes/`, 2013.

[Zyg88] A. Zygmund. *Trigonometric series. Vol. I, II.* Cambridge Mathematical Library. Cambridge University Press, Cambridge, 1988. Reprint of the 1979 edition.

Index