Discrete Fourier Transform

V. CROQUETTE

January 2017

Abstract

The Fourier transformation is at the heart of many signal treatment methods. We shall present here its principle of a projection on an orthogonal base, some specific signals where it may be applied and finally its specific properties that make it a most wanted method in signal treatment.

1 Signal decomposition on a base of orthogonal polynomials.

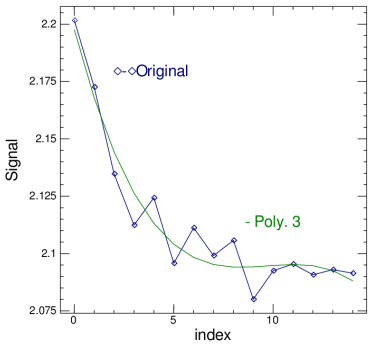

Figure 1: *Experimental signal (in blue) over 15 points, that we wish to adjust with a smooth curve. In green, an adjustment with a third order polynomial obtained by projecting the signal on an orthogonal base.*

On many occasions, we wish to describe experimental data $S(x_i)$ by a curve; a line or a simple polynomial as in Fig. 1. The recipe to achieve such an adjustment is rather simple for a line (linear regression) but becomes more involve for a polynomial of order higher than one. As a matter of fact, to fit data to a polynomial $P_n(x_i)$ the best is to minimize the chi-square which takes the following expression: $\chi^2 = \sum_0^m \frac{(S(x_i)-P_n(x_i))^2}{\sigma_i^2}$ where x_i are the abscissa of the measurement points in ($i \in [0, m]$ and where σ_i represents the error of the measure at point i). The χ^2 expression represents the ÃĆÂń distance ÃĆÂż between the model and the experimental points. The minimization process is not straightforward and often requires the knowledge of the derivatives of $P_n(x_i)$, may take some time as soon as the number of adjusted parameters is high. Since its principle is to find a minimum of a surface in a space having as many dimensions as we have parameters. Finding this minimum is challenging if the surface presents many local minima, this explains the lengthy process which is not always working. If the experimental points are sampled with the same interval along the x axis

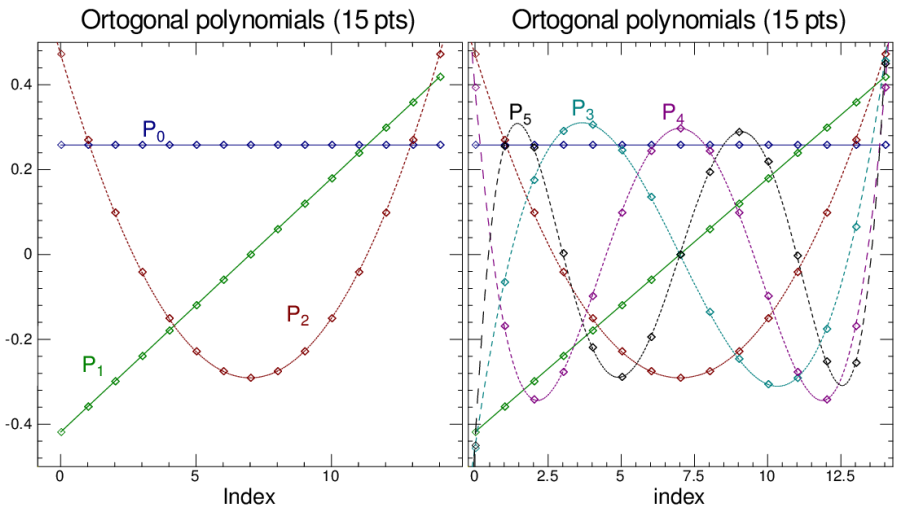

Figure 2: *Build-up of a base of orthogonal polynomials over 15 points. On the left, the first one P_0 corresponds to a constant value, the second P_1 is a line which mean value is null. The third P_2 is a parabola which is by definition orthogonal to P_1 and which mean value was set to zero so that it is orthogonal to P_0. On the right, we have iterated the process to define P_3, P_4 and P_5*

and if the error σ_i is the same for all points, there is a method simpler and faster than the previous one: it consists in projecting the data on a set of orthogonal polynomials. The projection consists in computing a scalar product between the signal $S(x_i)$ and the polynomials $P_n(x_i)$, that is $a_n = \sum_0^m S(x_i).P_n(x_i)$. This computation is simple and never leads to any exception. It is possible to define a polynomial base with P_n of increasing order n that are normalized, that is $\sum_0^m P_p(x_i).P_n(x_i) = \delta(p-n)$. The polynomial definition actually depends on the number n of points used, see for instance "Introduction to numerical analysis" by Hildebrand p. 350. Once the polynomial base established, we can proceed to the projection of the signal $S(x_i)$and then use a_n to work on the signal. On figure 1, we have only kept the first four values of a_i, this is equivalent to adjust the signal with a polynomial of order 3 which is a form of filtering. The projection on the orthogonal polynomial base is a change of base done on our signal. The value a_0 corresponds to the mean value of our signal, a_1 provides the average slope, a_2 is the curvature... The useful coefficients a_i are characteristic of the signal. The coefficients corresponding to higher order polynomials are often less interesting. On the other hand, coefficients associated with n big may have precision issues. The orthogonal polynomial base is thus useful mainly for its first components, we shall see in the following that the Fourier transform is also a projection on a base of sinuses and cosines. But this time all the components will present a physical interest.

2 Fourier Transform

Mathematical the Fourier transform is defined for continuous functions from $-\infty$ to $+\infty$, all numerical signals that we shall be discussed will concern discrete Fourier transform over a finite time interval spanning N samples. Of course when N becomes big one may think that we approach the continuous limit, but one should keep in mind that discrete Fourier transform assumes that the signal is in fact periodic over N.

2.1 Discrete Fourier transform

Discrete Fourier transform is easier to describe for complex variables, even if it also applies to a real signal. It corresponds to a change of base of the signal from real space (or of positions) S_r towards the base of frequencies (or wave vectors) associated with the functions $e^{-2i\pi.k.r}$ with the variable k. As in the polynomial example, if the signal is sampled at regular intervals and if its statistical error is constant for all samples, one can use the scalar product method to obtain the Fourier coefficients. The signal can be written $S_r \in C$ with $r \in [0, N[$ in the direct space while the Fourier component writes:

$$a_k = \frac{1}{N} \sum_{r=0}^{N-1} S_r.exp(-2.i\pi.k.r/N) \tag{1}$$

with $k \in [-N/2, N/2[$ and $1/N$ the normalization coefficient. If we have N complex points for the signal (that is $2N$ variable), We obtain N Fourier modes (also complex, that is again $2N$ variables). This is clearly a base change. One can obviously define the inverse Fourier transform which leads back to the original signal starting from its frequency components:

$$S_s = \sum_{k=-N/2}^{k<N/2} a_k.exp(2i\pi.k.s/N)$$

All the Fourier components content the same information as the signal in the direct space. The energy is, of course, conserved this is the **Parceval theorem**. The Fourier base has its components all orthogonal and one easily demonstrate that:

$$\frac{1}{N} \sum_{r=0}^{N-1} exp(-2i\pi.p.r/N).exp(-2.i\pi.q.r/N) = \delta(p-q)$$

. It is useful to expand a_k in the S_s expression:

$$S_s = \sum_{k=-N/2}^{k<N/2} \frac{1}{N} \sum_{r=0}^{N-1} S_r.exp(-2.i\pi.k.r/N).exp(2.i\pi.k.s/N)$$

One can put together and inverse the summation order:

$$S_s = \frac{1}{N} \sum_{r=0}^{N-1} S_r \sum_{k=-N/2}^{k<N/2} exp(-2.i\pi.k.(r-s)/N)$$

The sum of k, on the right, is more easily visualized in the complex plane, it corresponds to the sum of N vectors with a module 1 and which angles are at the vertex of a polygon with N sides if $r \neq s$. In this situation, all the vectors add-up to zero. When $r = s$ this sum is equal to N. And thus recovers the expected result.

2.2 Case of a real signal.

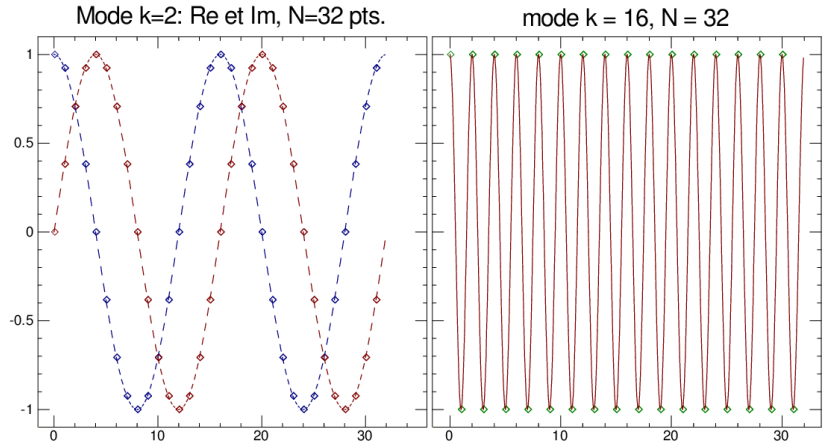

Figure 3: *On the left, k = 2 mode for N = 32, only the points are meaningful in view of the data sampling, the dotted lines are a guide to the eyes. In red, the cosine, in blue, the sinus. On the right, k = 16 mode for N = 32. For this particular mode $k = N/2$ the real part in the cosine takes only the alternating values +1 and -1, we have drawn the cosine in dashed lines as a guide for the eyes. The imaginary component in sinus (not represented) is null since the signal points are sampled exactly when the sinus is equal to zero.*

Many signals are real and thus is no imaginary parts $Im(S_i) = 0$, the signal only contents N variables; Its Fourier transform still contains N complex modes that is $2N$ variables. In fact one easily demonstrate that if the variable S_i is real, the Fourier coefficients a_k and a_{-k} are complex conjugates. The knowledge of a_k is sufficient to determine a_{-k}. We only need $N/2$ Fourier modes to describe a real signal. This corresponds indeed to N independent variables ($N/2$ times 2 for the real and imaginary parts). The signal may be written $S_r \in R$ with $r \in [0, N[$ in real space, the Fourier components write:

$$Re(a_k) = \alpha \sum_{r=0}^{N-1} S_r.cos(-2.i\pi.k.r/N) \text{ and } Im(a_k) = \alpha \sum_{r=0}^{N-1} S_r.sin(-2.i\pi.k.r/N)$$

with $k \in [0, N/2]$. The $k = 0$ mode is special, since its imaginary part is null. The $k = 1$ mode corresponds to a single arch oscillation of a cosine or a sinus covering the entire signal that is $[0, N]$. All modes are strictly periodic in the signal window. The $k = N/2$ mode is again very special. As one can see on Fig. 3, the $k = N/2$ mode corresponds really to an extreme case in sampling: for the cosine component, each sample is either equal to +1 or -1 of the cosine. There is no imaginary part since all samples correspond to the zero of the sinus. This mode corresponds exactly to the sampling rule of Shanon that we shall discuss later. If you decide to filter a real signal, be careful that the $N/2$ mode cannot be phase shifted since its imaginary part is always zero.

2.3 The example of a square wave signal.

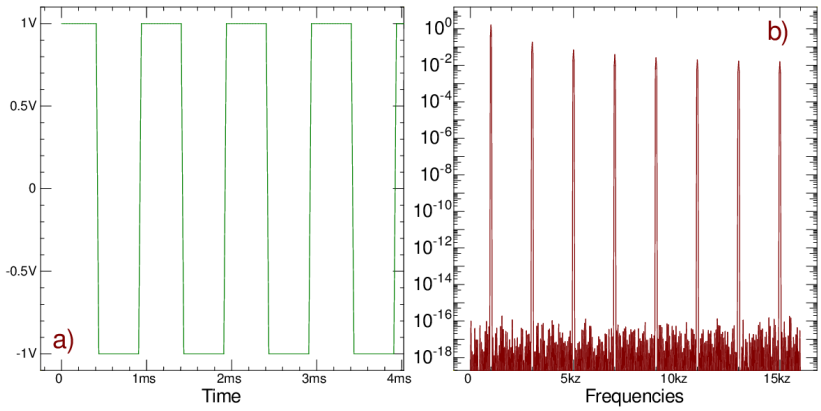

Figure 4: *On the left, a square wave signal in real space with a period of 32 pts over a total N = 1024. On the right, the power spectrum in log scale. One notices that this spectrum contains only sharp peaks corresponding to odd harmonics with $k = (2.n+1).32$. All the other modes just contain numerical noise.*

A sinus signal having exactly the frequency of the Fourier mode leads to a spectrum where only one mode has a strong amplitude. All the other modes are null or just contain numerical noise. The Fourier transform of a sinus in real space is a Dirac function in frequency space. For a square wave signal, its power spectrum leads to a series of Dirac peaks corresponding to the odd harmonics.

2.4 Case of the Dirac signal

A signal corresponding to a Dirac peak in real space is drawn in Fig. 6. This is a signal where all samples are zero except one in r where the signal takes a finite value (let's say one). The Fourier coefficients $d_r(k)$ are easy to compute :

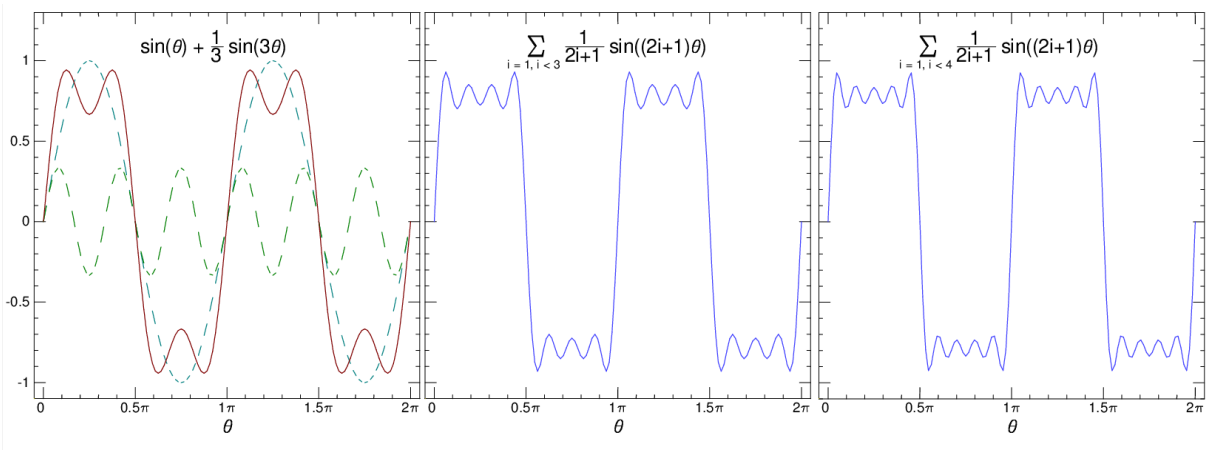

Figure 5: *It is possible to reconstruct a square wave signal by adding different odd harmonics of its fundamental frequency. We present here reconstructions at various stages.*

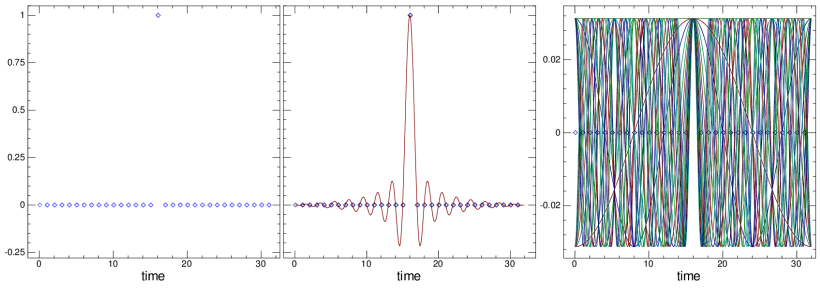

Figure 6: *On the left, Dirac signal in the time space with $N = 32$. On the right, ensemble of all Fourier components of this Dirac peak. In the middle, peak reconstruction using these components drawn with interpolation between the 32 initial points. In fact the Dirac peak is a cardinal sinus which maximum is located on the peak. On the other hand, the fact that the 31 remaining points in real space are null, is related to the fact that these points are located precisely on the zero of the sinus cardinal.*

$d_r(k) = (1/N)exp(-2i.\pi.k.r/N)$. All modes have equal amplitudes but different phases. The phase of each k is such that the signal has a maximum in real space in j. Since all modes present a maximum at this point, as soon as one escapes these points, the mode oscillations cancel each other as one can see in the figure Fig. 6 at the center. If we consider a

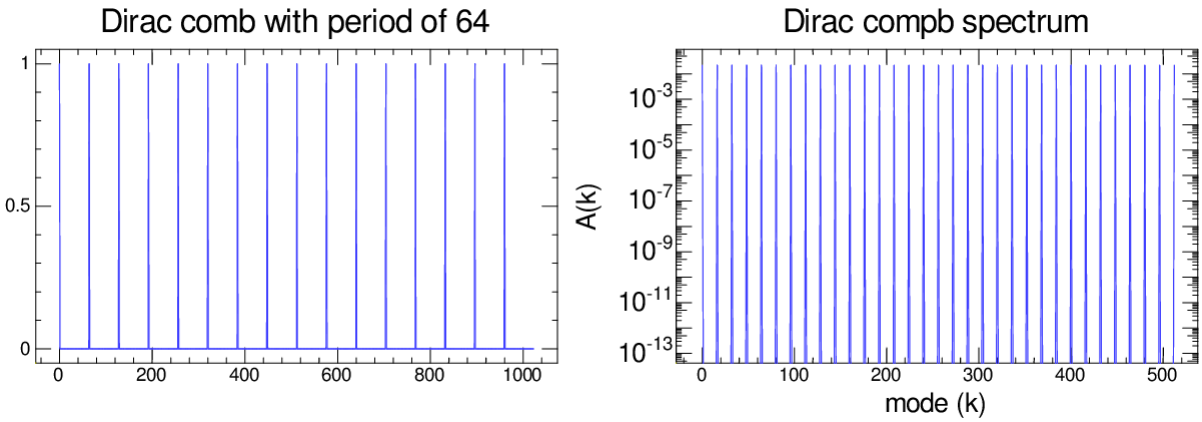

Figure 7: *The Fourier transform of a Dirac comb is also a Dirac comb. On the left, Dirac comb in real space with a peak every 64 points over 1024 total points. On the right, Power spectrum of the signal on the left with one peak every 16 modes.*

periodic signal with τ as the period, it may be decomposed as a Fourier series with all elements occurring at harmonics of this fundamental frequency $f_1 = N/\tau$. This means that the Fourier transform of a periodic signal is a Dirac comb. In

the special case where the signal is also a Dirac Comb, its Fourier transform is also a Dirac comb with all modes of equal amplitude (see Fig. 7). For the discrete transform we have the following relation $\tau * f_1 = N$.

2.5 Interpolation between sampling points.

The previous discussion allows us to better appreciate several points: the signal representation is limited by the small number of samples per period for modes having large k values. If we want to interpolate between the original points without altering its content, one may use the sum of the Fourier modes expressed in between the sampling positions. $S(x) = \beta \sum_{k=-N/2}^{k<N/2} a_k.exp(2\pi.k.x/N)$ Where x is a real number in between 0 and N, this is what we have done on figure Fig. 6, at the center, we notice that the Dirac peak is in fact a sinus cardinal or a sinc function. This is normal since the Fourier components are defined over a rectangular window in frequency with maximum value $k = \pm N/2$ and equal zero above. The bandwidth of our signal is thus a rectangular window in frequency space. The impulse response corresponds to a sinc.

2.6 Relation between phase and offset (delay theorem)

We have just suggested that the phase of a mode allows adjusting the position of its maxima. If we alter the phases of all the Fourier components of a Dirac peak in such a way that $\phi(k) = -2.\pi.k.\delta x/N$, we shift the point where these components coherently add by δx; Thus we have shifted the Dirac peak by the quantity δx. This process does not alter the mode amplitudes, more over this process may be applied to all signals (the Dirac peak better illustrates the phase coherence). This method of signals shifting is possible for non-integer values of δx, it is reversible and does not alter the signal. As we have just seen that a Dirac peak is in fact a sinc function, if we shift the sinc function of figure Fig. 6 at the center by $dx = 0.5$, all the extrema of the sinc will coincide with the sampling points thus decorating the signal. This

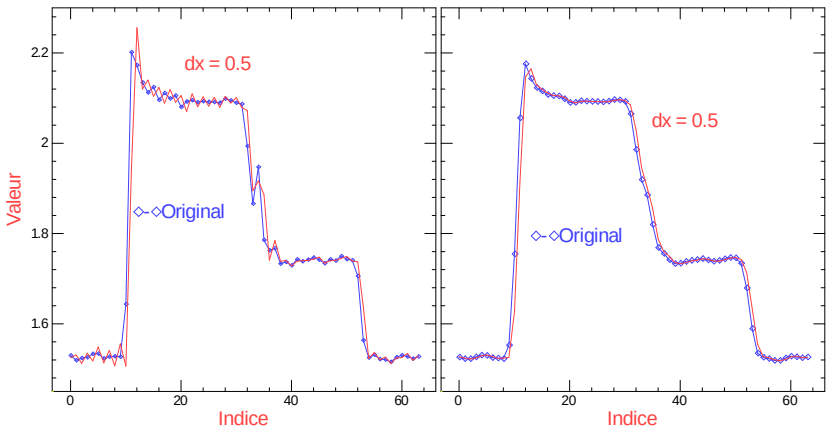

Figure 8: *On the left, the original signal shifted by half a pixel using the Fourier transform method. As this signal contains high frequency components, the shifted signal exhibits oscillations. On the right, the shifted signal has first been low-pass filtered to suppress high frequencies components, now the shifted is smoother.*

effect is perturbing but it is possible to avoid it: the slow decrease of the sinc oscillations is in fact caused by the sharp transition occurring $k = N/2$ in Fourier space, if we damp the amplitude of the Fourier component by filtering out the high frequency components so that the transition in Fourier space is smooth, far less oscillations will be seen in real space. If our signal is real, translation in real space made by phase shifting the Fourier components will lead to a small problem with the mode $k = N/2$: any phase shift of this mode will lead to an imaginary part that may be conserved only if the signal is made complex. One way to avoid this issue is to have filtered the high frequency component so that the $k = N/2$ mode is set to zero.

3 The Fourier base: the ideal base for linear differential equations.

3.1 Estimate of the derivative of a signal.

Taking the derivative of a signal is a classical operation, as in the case of the signal shift, the approximation consisting in taking the difference between two consecutive points is far from convincing. The first obvious problem is that this method evaluates the derivative just at the mid distance between two sampled points thus the derivative is in fact shifted by half a sample. The Fourier transform provides an interesting method, it consists of multiplying each mode by $i.k$ which is simple. (Again the $k = N/2$ mode will lead to the problem expressed previously). The method is easily expanded to higher derivatives, etc. The great interest of the Fourier method is that it does not alter the original signal: integrating the derivative of the signal will bring back the same signal (excepted for the continuous component).

3.2 Spectral methods.

To integrate a differential equation with spatial derivatives like a heat diffusion equation, wave propagation or flow velocity, the computation of spatial derivatives is fundamental. Let us imagine that we want to describe waves propagating on a rope without friction. $\partial^2 F/\partial^2 t = a\partial^2 F/\partial^2 x$, The wave energy is time invariant, if we compute partial derivatives based on two consecutive points difference separated by h, we make a linear approximation of the derivative with the accuracy of $1/h$, we also need to integrate in time with a step time dt. At each time step we need to evaluate the spatial derivative, if we have a small error in these evaluations (let us say that we make a small error in this process losing or gaining a small amount of energy), this will grow exponentially in time! If we lose energy, our wave will disappear sooner than expected, but if we gain energy, the wave will diverge: this is again really problematic. By taking the spatial derivatives in Fourier space, we avoid this issue and we are only limited by the numerical accuracy. This is the spectral methods. These methods are highly efficient but suffer some limitations: First, they require passing in Fourier space to compute the spatial derivatives, but some computations cannot be done in Fourier space (nonlinear terms for instance), leading to transformation back and forth between real and Fourier spaces. Second, Fourier transform works for periodic data, which means that when the wave packet reaches the end of the cord at point $N-1$ it then comes back to the origin. Periodic conditions are often interesting in Physics but they are not always acceptable.

3.3 Fast Fourier Transform FFT

The discrete Fourier transform is done by directly applying the equation 1 requires N multiplications to compute the Fourier coefficient a_k, as there is also N equations, the computation time of the simple Fourier transform grows like N^2 multiplications. This is a heavy computation burden. Luckily Cooley and Tukey have noticed that if N is a power of 2, the symmetry of the equations allows computing a_k much faster (see the article of R.J. Higgins AJP **44**, 8, (1976)). By using the symmetry even and odd it is possible to reduce the number of operations to $N.log_2 N$. If we need to achieve a Fourier transform over 1024 points (2^{10}) one switch from 10^6 operations to 10^4 operations that is one hundred gain! (On a Apple II in 1980, the FFT over 1024 points needed 1 second, nowadays GPU perform million FFT in one second). The algorithm is simple and nice looking if $N = 2^n$ (it covers one page), If N is not a power of two, other algorithms have been proposed which achieve the same speed gain, but are far more complex in their implementation. Many packages are free and provide excellent libraries for FFT for most N using advances fast processor operations (SMID): http://www.fftw.org.

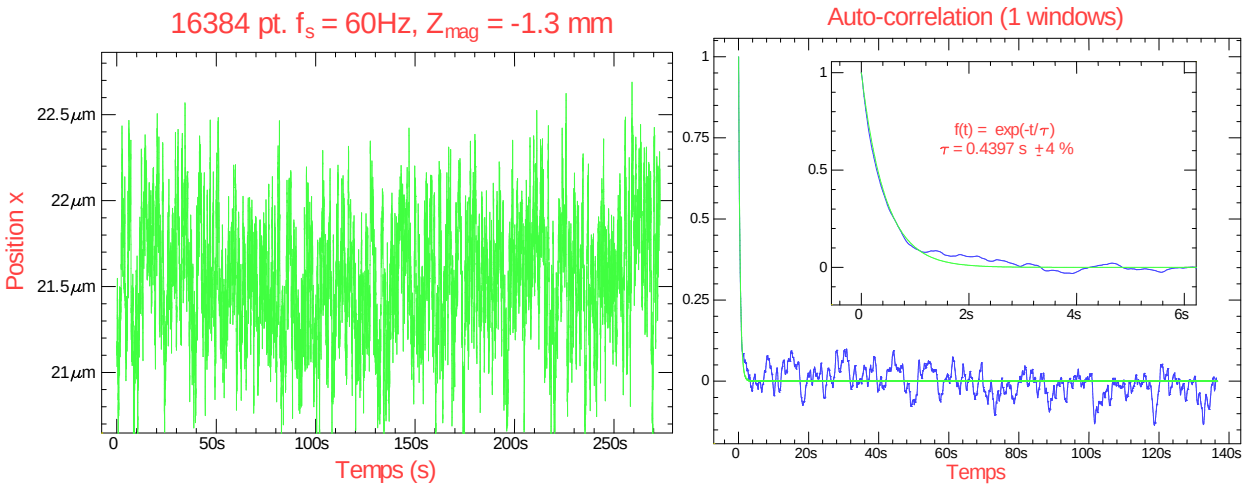

Figure 9: *On the left, experimental signal of the Brownian motion of a bead attached to a DNA molecule. On the right, autocorrelation function of the signal on the left (in blue), this demonstrates that the fluctuations have some memory with a characteristic time of half a second (see insert). At a longer time, the autocorrelation function presents only noise.*

3.4 Fourier transform and correlation function.

We often have two signals of different nature that we think correspond to the same phenomenon buried in some noise. One want to look for a correlation between these two signals. The principle to look for the correlation is simply to multiply the two signals, when the correlated parts are in phase they will lead to a positive value while noise will not build any positive sum. Of course, this works if there is no continuous bias in the signal that one might need to remove before. The problem is usually a little more complex since the interesting signal often presents some delay between the

two representations. One needs to use the correlation product which basically repeat what we have sketched, but for all possible delays described by the variable τ. For two complex signals, this product writes:

$$C(\tau) = \sum_{0}^{N-1} X(t).\tilde{Y}(t-\tau)$$

where X is the first signal and $\tilde{Y}(t-\tau)$ corresponds to the second signal shifted in time. In this expression we assume periodic boundary conditions. As we have to evaluate this function for N possible delays over N samples, this product requires N^2 multiplication operations. If we write $x(t)$ and $\tilde{y}(t-\tau)$ in Fourier's space using the delay theorem, we obtain the interesting relation:

$$C(\tau) = \sum_{t=0}^{N-1} \sum_{k=-N/2}^{N/2-1} X_k e^{2i.\pi.k.t/N} . \sum_{k'=-N/2}^{N/2-1} \tilde{Y_k} e^{2i.\pi.k'.t/N} . e^{-2i.\pi.k'.\tau/N}$$

Thus

$$C(\tau) = \sum_{k=-N/2}^{N/2-1} X_k . \sum_{k'=-N/2}^{N/2-1} \tilde{Y_k'} \sum_{t=0}^{N-1} e^{2i.\pi.(k+k').t/N} . e^{-2i.\pi.k'.\tau/N}$$

Where the sum over t differs from zero only if $k = -k'$, which leads to:

$$C(\tau) = \sum_{k=-N/2}^{N/2-1} X_k \tilde{Y_{-k}} e^{2i.\pi.k.\tau/N}$$

One notices that it is possible to express the correlation function using the Fourier transform of each signal. In the case where $y(t)$ is the signal $x(t)$ one speaks of signal autocorrelation, we then obtain the **Wiener Kintchine theorem** stating that the correlation function and the power spectrum are Fourier transform of each other. The interest of this relation is that since there exists a Fast Fourier Transform algorithm, computing the correlation is done faster in Fourier space. This is true even if we need to take three Fourier transforms: one need to take the TF of $X(t)$ and $Y(t)$, we then compute the cross product in Fourier space and come back in the real world by taking the inverse TF of this product. Again the assumption of a periodic signal is made when using the FT. The correlation function generally contains useful physical information, if we compute the correlation function of the temperature of the air versus time, we will obtain the characteristic stability time of the atmosphere (its Lyapounov exponent) of order two days. In the example of the Brownian motion of a bead shown in Fig. 9, the characteristic time is equal to the response time of our system (here ≈ 0.5s). If we now compute the correlation function of a numerical noise generator, we will obtain a Dirac function since all samples are completely uncorrelated except with themselves for $\tau = 0$.

3.5 Convolution product and Fourier transform.

Many physical phenomena involve the convolution product: periodic signals may be described as the convolution product of one signal corresponding to one period by a Dirac comb. All physical systems measured through a device are convoluted by an instrument response function. When this instrument is good, its response function is sharp and the convolution is barely visible. But often this response function is not ideal, one may wish to de-convolute the experimental signal. It is easy to convolute a signal in real space but it is more difficult to realize the opposite operation. The Fourier transform brings us to a solution. For continuous signal, the convolution product writes:

$$f \otimes g(\tau) = \int_{-\infty}^{\infty} f(t).g(\tau - t)dt \text{ for a discret signal } = \sum_{0}^{N-1} f_r.g_{\tau - r}$$

Writing $f(r)$ and $g(\tau - r)$ in Fourier and using the delay theorem, one obtains the interesting relation:

$$f \otimes g(\tau) = \sum_{r=0}^{N-1} \sum_{k=-N/2}^{N/2-1} F_k e^{2i.\pi.k.r/N} . \sum_{k'=-N/2}^{N/2-1} G_{k'} e^{-2i.\pi.k'.r/N} . e^{2i.\pi.k'.\tau/N}$$

thus

$$f \otimes g(\tau) = \sum_{k=-N/2}^{N/2-1} F_k . \sum_{k'=-N/2}^{N/2-1} G_k' \sum_{r=0}^{N-1} e^{2i.\pi.(k-k').r/N} . e^{2i.\pi.k'.\tau/N}$$

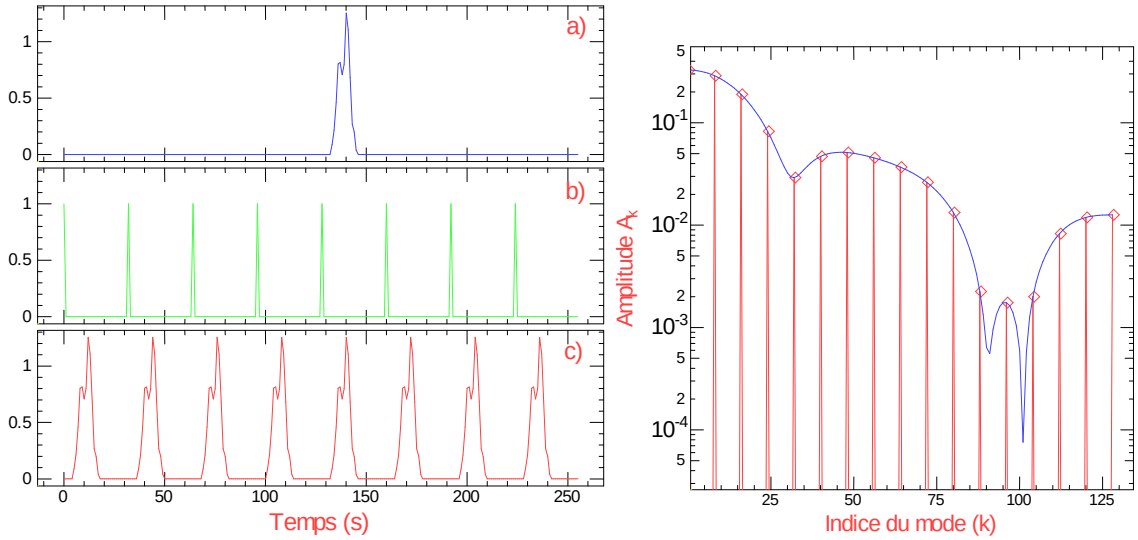

Figure 10: *On the left, construction of a one dimension ÃĆÂń crystal ÃĆÂż : a) experimental signal, b) Dirac comb, c) convolution product of a) by b) leading to a periodic signal. On the right, power spectrum of the signal a) (green continuous line), and of the periodic signal c). This signal appears as the product of the Dirac comb (not shown) by the one of the original signal. For the illustration we have scaled the spectrum of a) by a factor 8 since in c) there is 8 peaks. The analogy with a crystal allows better understanding X-ray pictures of crystals: the interesting information is inside one cell, one get it by inverse TF of the diffraction peaks.*

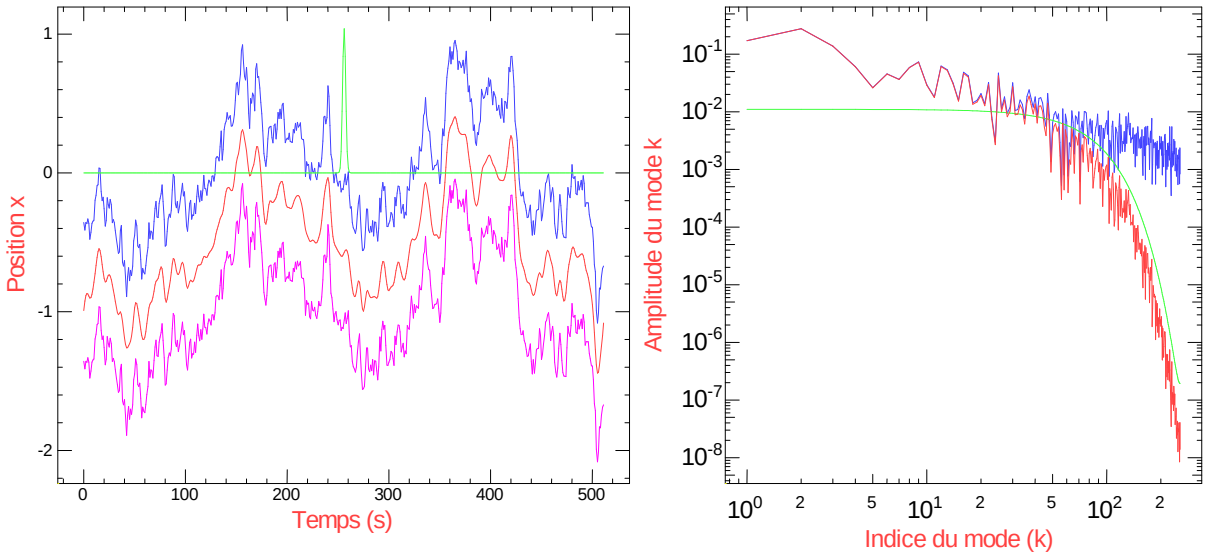

Figure 11: *On the left, in blue the original signal over 512 points, in green Gaussian response function with a width of two points (we have shifted this signal which is normally centered for clarity). In red signal obtained by convoluting the original blue signal by the green response function. Here the effect of the convolution is to low-pass filter (we have shifted the red signal by -0.5 for clarity). In magenta, we have de-convolved the red signal by that of the Gaussian response using the Fourier transform (the magenta signal is shifted by -1 for clarity). On the right, we have represented the Fourier mode amplitude of the signal shown on the left with the same color coding (we have skipped the TF of the magenta signal which is identical to that of the original signal). This spectrum is drawn in log scale since the power magnitude varies over a wide range.*

Where the sum over r is not null only if $k = k'$, which leads to

$$f \otimes g(\tau) = \sum_{k=-N/2}^{N/2-1} F_k G_k e^{2i.\pi.k.\tau/N}$$

One deduces that the Fourier transform of a convolution product of two functions becomes the product of the Fourier transform of these two functions. As for the correlation product, the Fourier transform allows speeding up the computation

of the convolution product. But it is also a very simple way to achieve de-convolution.

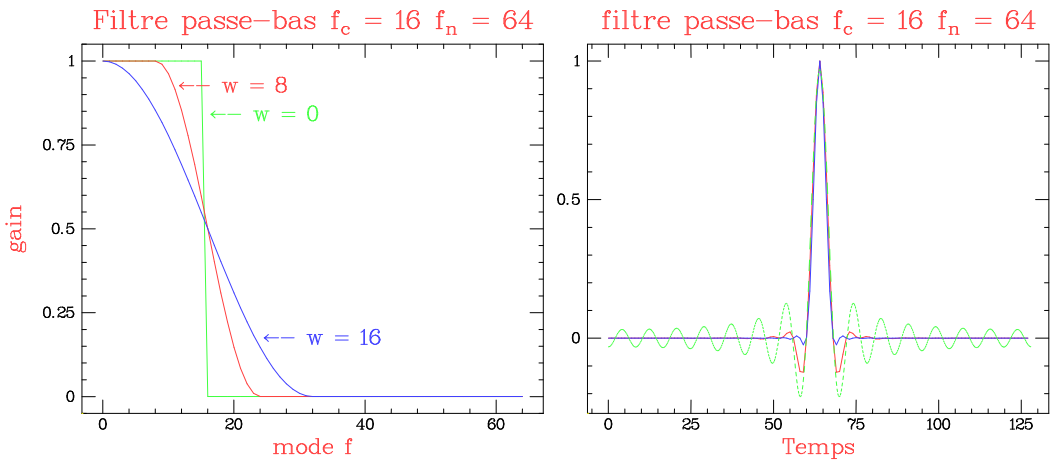

Figure 12: *Comparison of three low-pass filters having the same cutoff frequency $f_c = 16$ but with different shapes (they operate over 128 real points, that is 64 frequencies). On the left, transmission coefficients of these filters in Fourier space, they have transmission coefficients of 1 for modes such tha $f < f_c - w$, null when $f > f_c + w$ and in the medium area $T = (1 + sin(\pi.(f_c - f)/2.w))/2$. On the right, impulse response of these filters in real space. The pulse width is equivalent since all filters have the same cutoff frequency at mid-height, but the decay of the ringing strongly depends on the width w of the filter (These impulse response has been shifted by 64 points for clarity).*

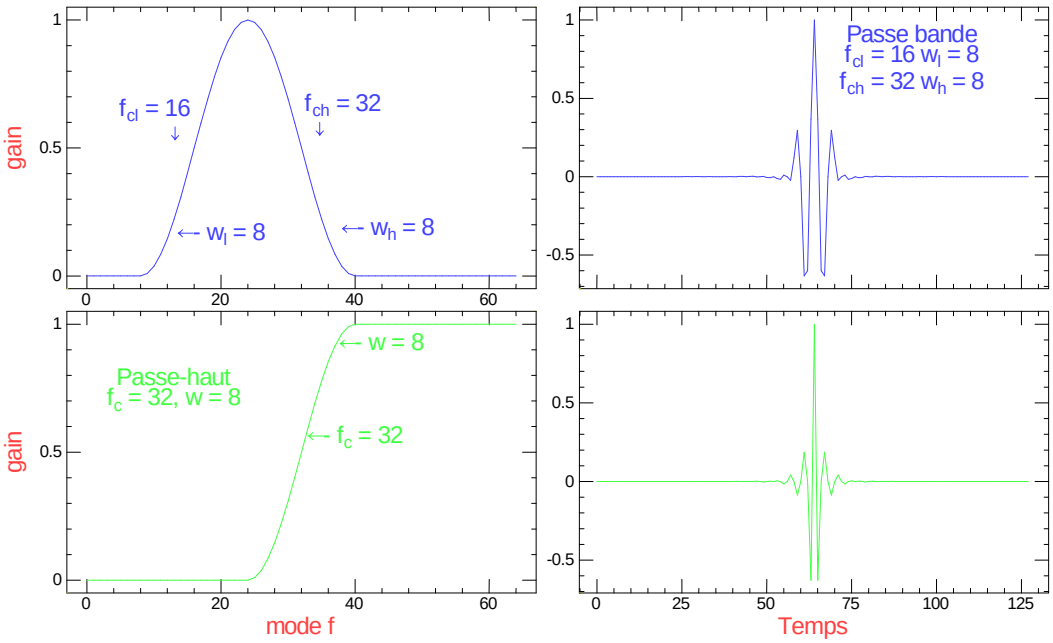

Figure 13: *Pass-band and high-pass filters (working over 128 points in real space, thus 64 frequencies). On the left, Fourier coefficients of transmission of these filters, pass-band on the top, high-pass on the bottom. On the right, real space impulse response of those filters (traces have been shifted by 64 points for clarity).*

3.6 Frequency filtering

Performing Fourier transform allows separating the various components of a signal. It is frequent that the useful part of a signal covers only a small number of modes in Fourier space while noise spreads all over the modes. Suppressing modes that only contains noise improves the signal as shown in the figure 10. There are many ways to attenuate modes in TF, the most common is the low-pass filtering where high frequencies are suppressed and high-pass where the low frequencies are removed and finally pass-band filters which allows selecting a set of components close to a central frequency. As we have illustrated in the figure 11, filtering is in fact convoluting data by an impulse function. In this peculiar case, we have

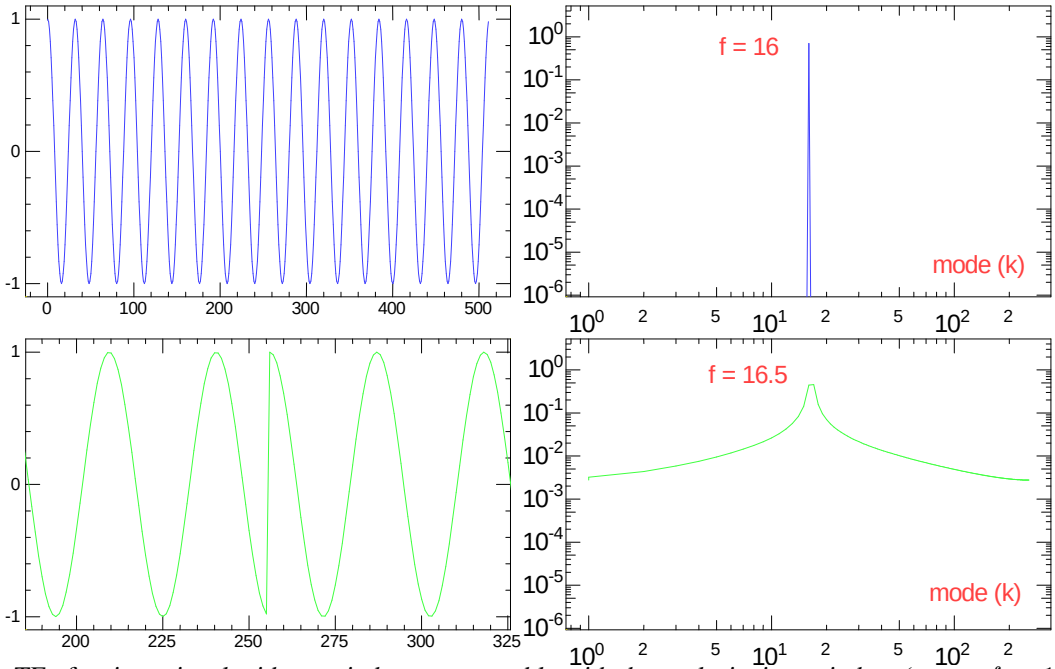

Figure 14: *TF of a sinus signal with a period commensurable with the analysis time window (on top $f = 16$) and of a sinus incommensurable with the time window ($f = 16.5$). On the left, the signal in real space shifted by $N/2$, for the top signal, the periodicity is perfect, for the bottom signal the shift leads to a phase jump. On the right, TF of each signal, the occurrence of a phase jump leads to strong perturbations in the spectrum.*

used a Gaussian impulse response which also leads to Gaussian frequency shape in Fourier space. Taking the FT of the impulse response of a filter in real space, leads to the attenuation of the filter in the frequency domain. The choice of this attenuation profile depends on the wanted usage. If we are mainly interested to visualize the frequency components of the spectrum, it is possible to use stiff filters in frequency domain as a rectangular function where $T(f) = 1$ if $f < f_c$ and 0 elsewhere (this filter is called "brick-wall"). In principle this filter provides the best signal to noise. But if one wants to consider the signal in real space, the impulse response of the brick-wall filter is sinc which has a lot of ringing which is not very good. Finding a filter that does not alter frequency components and shows little ringing is not simple it corresponds to the principle of Heisenberg in quantum mechanics.

3.7 Non-periodic signals

Of course most signals that we are interested to study are not periodic or more exactly not periodic with the analysis window. If we want to use the discrete FT it implies such a periodicity of the signal over the N points of the analysis window. If no special treatment is applied strange effects will occur in our FT and the validity of the previous theorems will not be true. Effect of the absence of periodicity on a sinus signal: since the FT assumes that the signal is exactly periodic over the analysis window, that is over the interval $[0, N]$, the point $N - 1$ must connect perfectly with the point 0. Let us compare this case to the one where there is no commensurable relation with the analysis window as in figure 14. The problem arises at the reconnection between the points $N - 1$ and 0; to visualize the issue, it is best to shift the signal by $N/2$ which brings back points $N - 1$ and 0 respectively $N/2 - 1$ and $N/2$. As one can see in the figure 14 on left bottom. The effect of this discontinuity is strong on the Fourier spectrum (compare the spectrum of the figure 14). This phenomenon is particularly annoying since if we sweep continuously the frequency of a generator during an analysis, the peak corresponding to the frequency will alternate between commensurate and non-commensurate with the analysis window and the shape of the peaks observed will strongly change during this action. To avoid this issue, it is best to anneal the strength of the signal at both ends, thus erasing the phase jump. The solution consists in multiplying the input signal with a window reaching zero at both edges in 0 and N. To improve even further the windowing effect, one can choose a window where the first derivative also cancel on the edge. The most common window shape is called the Hanning window which writes:

$$F(r) = (1 - cos(2.\pi * r/N))/2$$

The Hanning window is a compromise, there exist many other windows addressing this issue; they will either try to have a flat response in frequency but at the expense of their frequency resolution or the opposite. The Hanning window is simple and should always be tried first. The Hanning window will correct issues when observing Fourier's spectrum but FT may

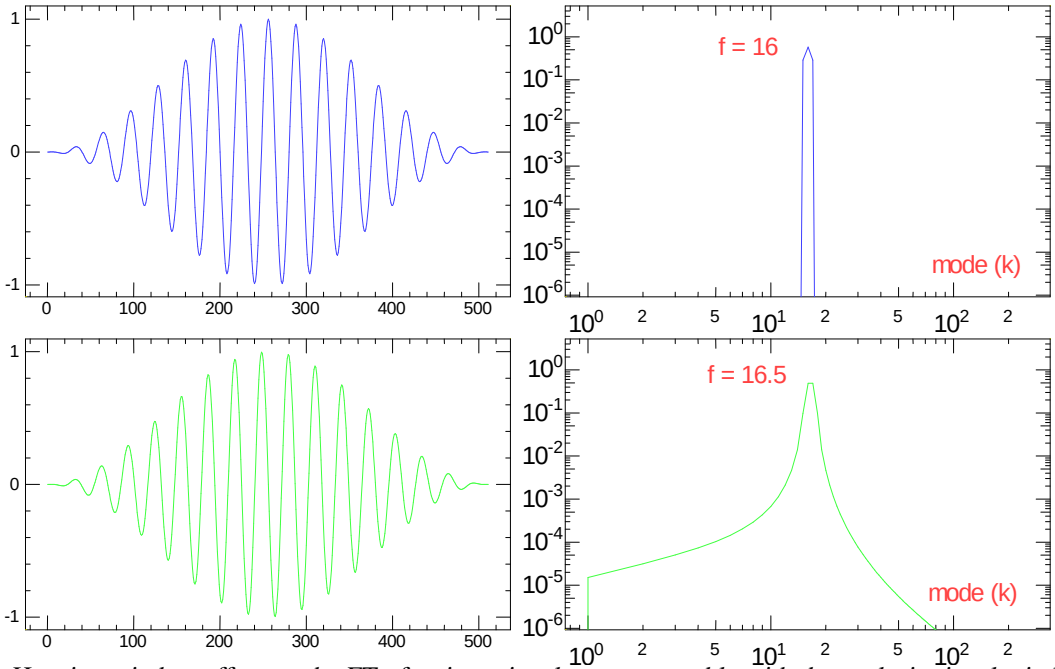

Figure 15: *Hanning window effect on the FT of a sinus signal commensurable with the analysis signal window (on top $f = 16$) and of an incommensurable sinus with the window ($f = 16.5$). On the left, the signal in real space is multiplied by a Hanning window. On the right, FT of each signal, for the top signal, the window leads to a wider peak, for the bottom one, the phase jump effect is strongly reduced.*

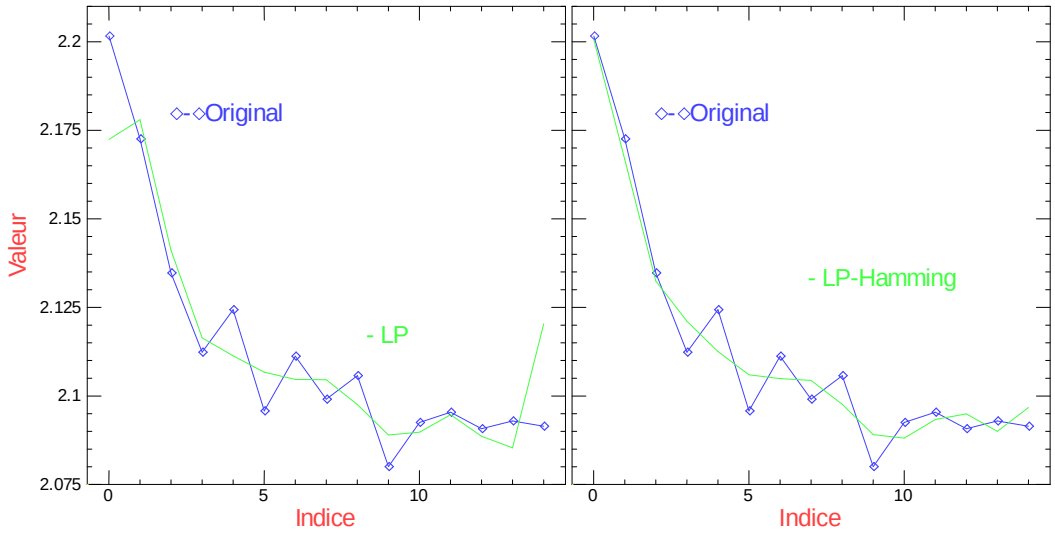

Figure 16: *Effect of Hamming window on filtering a signal non-periodic with the analysis window. On the left, the signal in real space treated directly with FFT. Has the signal has a discontinuity, the filtered signal presents a strong perturbation at its extremities. On the right, the signal was first multiplied by a Hamming window before low-pass filtered in Fourier space, in real space the signal was multiplied by the inverse of the Hamming window. The discontinuity at the edge has now gone.*

also be used to filter signal thus the signal is finally observed in real space. The same problem is also true for a correlation, a convolution and a de-convolution. If one uses the Hanning window before taking the FT to filter and comes back in the real world by inverse FT, the filtered signal will still be multiplied by the Hanning window function, thus it is depressed at its edges. Owing to its mathematical form, Hanning function cannot be inverted (since it equals zero on two points). A variant of the Hanning function solves this issue this is the Hamming window that very much looks like the Hanning one except that it remains non-zero on the edges, thus allowing it to be inverted. Of course being non-zero at the edges lead to evidence the phase jump related to non-periodic signals. However, in the Hamming window the signal is still strongly reduced at the edge avoiding most unwanted effect. Again the Hamming window is a compromise that you might adjust

according to your application. The Hamming windows equation is:

$$F(r) = 0.54 - 0.46.cos(2.\pi * r/N)$$

3.8 Spectrum and correlation functions of some classical systems

3.8.1 Low-pass filter of order 1, case of the RC circuit

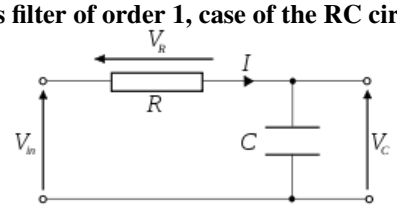

Figure 17: *Low-pass filter made with a RC circuit.*

The most classical filter is made with a resistor and a capacity, Fig. 17. The signal V_{in} is applied to one pole of the resistor R while it second plle is connected to the capacity C, the second part of the capacity is connected to the ground. The impulse response is a decreasing exponential $v(t) = V_0 exp(-t/\tau)$ with $\tau = R.C$. It is easy to show that the frequency response of this filter in Fourier's space is a Lorenzian $V^2(\omega) = 1/(1 + \omega^2/\omega_0^2)$ with $\omega_0 = 1/R.C$. This

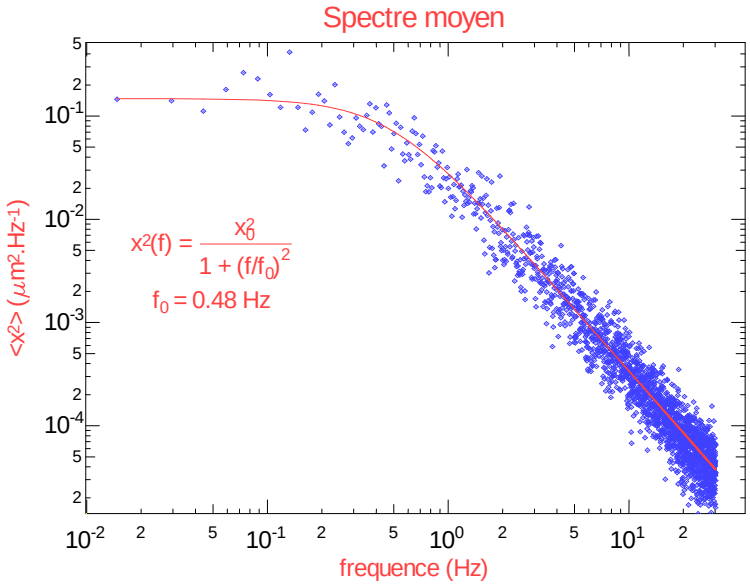

Figure 18: *Spectrum of the bead Brownian fluctuations tethered to a DNA molecule. This spectrum averaged four times is fitted to a Lorenzian. On needs to compare with the correlation function of the figure 9since it applies to the same signal.*

filter is the simplest that one can build, it let pass low frequencies as long as they are smaller than $f_c = \omega_0/2.\pi$. Above that, it attenuates their amplitude which a decay like $1/\omega$. The response is in phase at low frequency, at high frequency it reaches asymptotically $90°$ and is $45°$ at f_c. When we measure transmission in Decibels the attenuation is 6 dB per octave or 20 dB per decade. This moderate attenuation is often insufficient. Many systems are described by this type of spectral response. For instance, the micron size bead tethered by a DNA molecule that we have presented previously. As a matter of fact, the DNA molecule acts as a spring while the viscous drag of the bead leads to the damping. At this low scale the inertia of the bead is completely negligible and the motion equation writes: $6\pi\eta r\dot{x} + kx = 0$.

3.8.2 Low pass filter of second order

This class of filter describes for instance, the motion of the loudspeaker membrane as a function of the frequency. The membrane and the driving coil which position is described by x, have a mass m, they are attached by a spring of stiffness k. The friction terms and the dissipation in the coil induce dissipation or viscous drag terms proportional to the membrane speed. The equation governing x takes the form: $m\ddot{x} + \gamma\dot{x} + kx = F_o(t)$ where $F_o(t)$ is the force generated by the current circulating in the coil and γ is the friction coefficient. Si $F_o(t) = Aexp(i\omega t)$ we obtain $\tilde{x}(\omega) = \frac{x_0}{k + i\gamma\omega - m\omega^2}$ This system behaves like a second order filter, it is a low-pass filter with a cutoff frequency $f_c = \omega_0/2.\pi$ with small phase shift at low frequency with $\omega_0 = \sqrt{k/m}$. At this frequency it may present a strong or weak resonance, phase shift change rapidly passing by $90°$ at the resonance. Beyond this frequency, the filter attenuates the amplitude of the signal which decays as $1/\omega^2$. The transmission measured in Decibel, present a slope of 12 dB per octave or of 40 dB per decade.

3.9 Hilbert transform.

So far, we have mostly discussed the case of real signals. But we are often interested in temporal oscillations. It is very common to try measuring the period, or the frequency, or the phase of oscillations. One method consists of measuring the

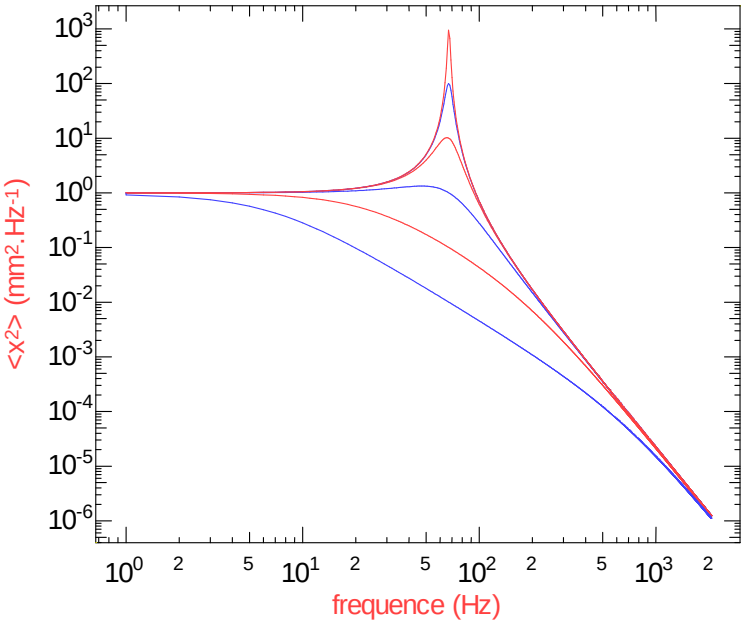

Figure 19: *Power spectrum of the displacement of the membrane of a loudspeaker as a function of the excitation frequency, in log-log scale. We have made the friction coefficient variable in a wide range so as to span the extreme cases over the high-quality factor to the over-damped situation.*

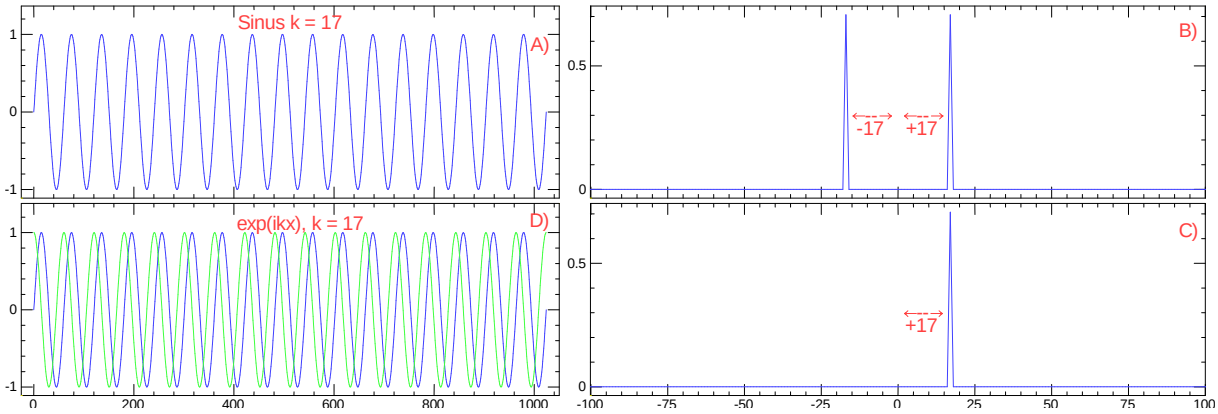

Figure 20: *Principle of the Hilbert transform.*

distance separating the place where the signal passes by zero. However, this method suffers from multiple issues. In this case, it is often more common to use a complex representation of the signal. Let us compare the function $cos(\omega t)$ to its complex equivalent $e^{i\omega t}$. As this last one keeps a constant modulus, this allows to easily extract the phase $\phi = \omega t$ using real and imaginary part. The instantaneous frequency of the signal ω is obtained in derivating $\phi(t)$. Thus, it is tempting when we consider a signal centered on a specific frequency and to render it complex, this is precisely what the Hilbert transform allows. The Hilbert transform consists in changing a cosine or a sinus in a function $e^{i\omega t}$. If we the Fourier transform of a real signal, the modes at the frequencies $+k$ and $-k$ are both simply complex conjugates. In Fourier space, if we suppress negative modes, we do not loose information, but if we convert the signal in real space by invert FT, we construct a complex signal, this is the Hilbert transform. This is a simple operation starting from the Fourier Transform. In the case of the cosine or the sinus, the transform is straightforward since the modes in $+k$ and $-k$ are well defined. For a more general signal centered on a frequency, as long as the width of the packet of Fourier modes is narrow, there is no issue in the process. Things more complex arise when the width of the frequency peaks increase and reach $k = 0$. In this case, the two frequency peaks positive and negative mix over and render their separation problematic. In practice, the Hilbert transform should not be used when strong low frequency modes exist. In fact, it is often used after a band-pass filtering removing modes around 0.

4 Signal sampling.

All measuring instruments use numerical data, the music, the pictures, now the video and the television, have now turned to numeric data, while they were analogous before. With numerical approaches, we may have the temptation to forget that all signals are originally analogous. Microphones measure tiny pressure changes associated with sound waves generally

by measuring the small displacement of a thin membrane forming a condenser of a few picofarads. This leads to an electrical signal of a few millivolts. Image capture devices collect incoming photons on a small area semiconductor diode (a pixel), converting ideally each photon in an electron, each pixel integrates these electrons in a small capacity which voltage reflects the total number of electrons that have impacted the pixel during a lapse of time. After some signal amplification, these signals are digitized leading to a numerical information that is finally stored or transmitted without loss.

4.1 Principle of Analog Conversion

Digitizing an analog signal (Analog to Digital conversion) is not completely trivial even though the integrated circuits that realize this operation are nowadays extremely miniaturized and of outstanding quality. There exist several types of AD converter and we shall just sketch the principle of operation of one of them. First, we need to realize that even though the signals are now numeric, all the instruments used to restore sound and images are analogous (see the loudspeaker for instance). Thus we also need Digital to Analog converters, it turns out that it is easier to make a DA converter than the opposite the reason being that you often use a DA converters to make an AD converter. One classical DA converter uses a R/2R array associated to a series of analog switches as shown in the figure 21. In this circuit, the analog switches do not alter the voltage value at their level whatever their state is. Thus the voltage at each cell R/2R is divided by 2 and the analog switches direct or not the current which scales like 2^{-n} at cell n in a summing current amplifier. By applying a binary number to analog switches, one obtains a current at the summing node which precisely reflects the binary number times V_{reff}/R, with V_{reff} a reference voltage . To achieve the analog to digital conversion, one way consists of comparing

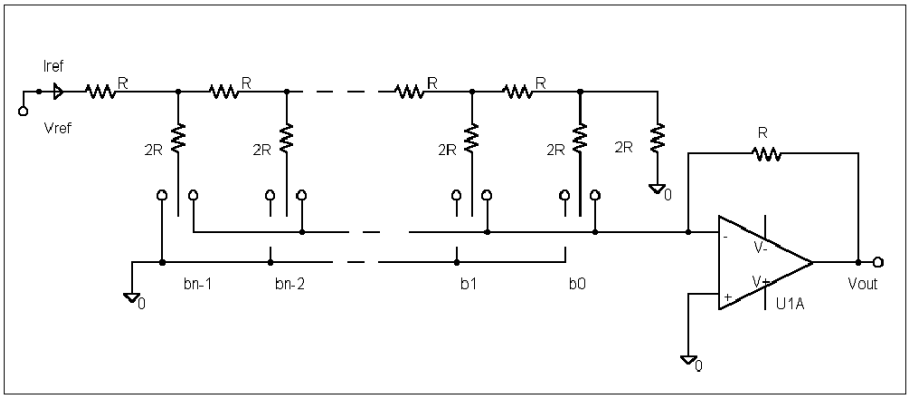

Figure 21: *Principle of a digital to-analog converter relying on an R/2R resistance network.*

the unknown voltage to that produces by a digital to-analog converter. By adjusting each bit one at a time starting from the most significant bit, one converges towards the voltage value represented by a binary number over N bits. Images use mostly $N = 8$ (but are now leading to $N = 12$) with sampling rate in the MHz range, sounds require more dynamics, typically $N = 16$ (and now $N = 24$) with a sampling rate of 44 kHz increasing to 96 or 192 kHz in good equipment. During AD conversion, it is fundamental that the signal that is highly dynamic remains constant during the N comparisons leading to the final number, otherwise the convergence of the process is compromised. AD converters usually imply a dedicate circuit called a sampled and hold circuit, which achieves this task. This circuit latches the incoming signal during the conversion. This sample and lock principle convolves the input signal by a Dirac comb having the sampling frequency. In this way it is possible to freeze fast variation of the input signal due to frequency components that may be higher than the sampling rate. If the signal frequency components are two high and indeed reaches the sampling frequencies, this leads to unwanted artefacts that will be impossible to correct later on. We have seen in the FT that the mode having the maximum frequency that may be used is $N/2$ and corresponds to two sampling points per signal period. This is also known as the Shanon theorem stating that the maximum frequency that may be reproduced with a sampling rate of f_s is $f_s/2$.

4.2 Absolute requirement of filtering before digitizing a signal, Shanon theorem

What happens if we sample frequencies above the limit specified by the Shanon theorem? Let us take the case of a signal at frequency f_s exactly equal to the sampling frequency: $f_d = f_s$, one period of the signal separate exactly two samples. Thus samples are identical our data signal frequency f_d lead to a constant signal as if its effective frequency was just null $f_{ds} = 0$ (this approach is also true for all the harmonics of the sampling frequency). If the frequency f_d is just a little larger than f_s, one will see a beating of the two frequencies leading to $f_d + f_s$ and $f_d - f_s$. As one can see, the component smaller than $f_s/2$ is $f_s - f_e$ which is a very low frequency that we cannot distinguish from a low frequency

really content in the signal. This issue is called spectrum aliasing. One can describe this phenomenon of the spectrum folding more accurately by using the convolution product that we have seen in the figure 22. Sampling is equivalent to multiply the signal by a Dirac comb. The product in direct space becomes a convolution product in Fourier space where the spectrum of the signal is convoluted by the spectrum of a Dirac comb. This convolution leads to a signal spectrum shifted by all the harmonics of the sampling frequency. If the spectrum of the signal is wider than $f_s/2$ (the Shanon criterion also called the Nyquist criterion), an overlapping occurs of these shifted spectra. In practice we can only access to the domain in $-f_s/2$ and $f_s/2$. The aliasing will lead to change in frequency $f_a = f_d - f_s$. Pour $f_d \in [f_s/2, f_s]$, the frequency of the signal grows when the frequency f_d rises. One must absolutely avoid spectrum refolding since this

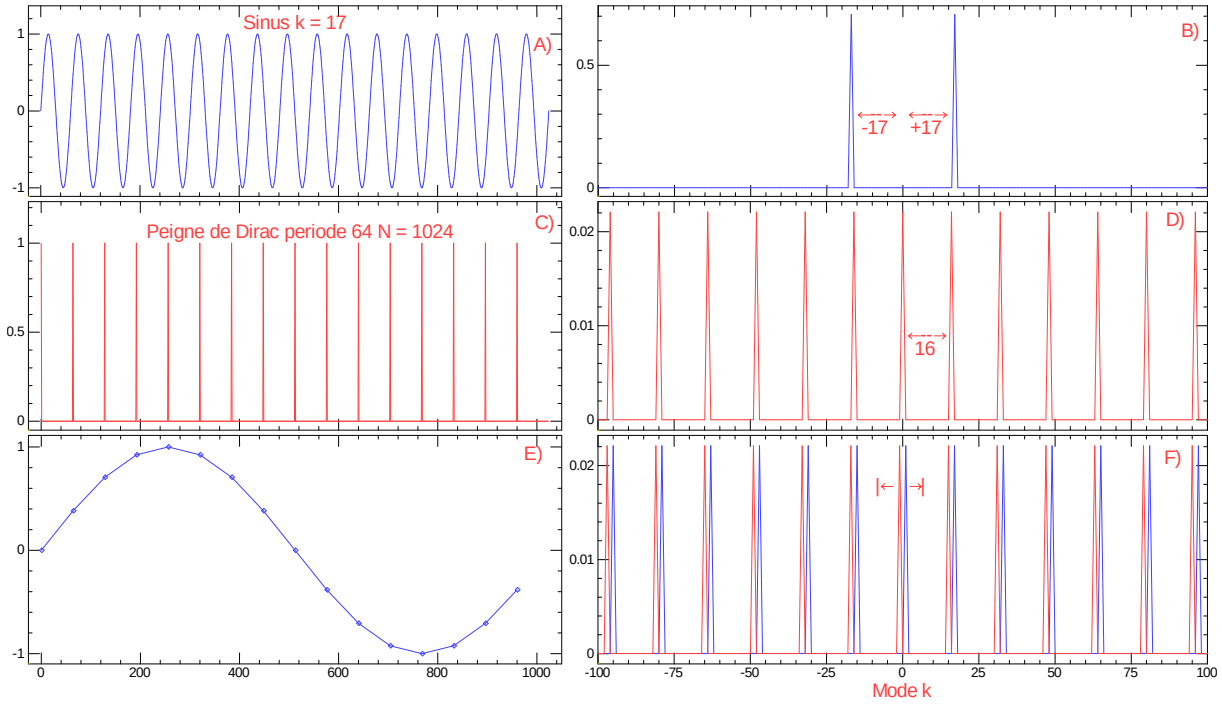

Figure 22: *Spectrum folding in aliasing mechanism: On the left A) a signal of frequency $k = 17$ displayed over 1024 points. B) Fourier transform of A). C) Dirac comb corresponding to the digitalization with 64 points period leading to 16 samples. D) FT of the Dirac comb, presenting peaks at $k = n.1024/64 = n.f_s$ with $n \in Z$ b). E) The sampled signal appears as a sinus with $k = 1$. To understand the refolding principle of this spectrum, one can describe the sampling as the direct product of A) by C) which corresponds to a convolution product of B) by D). As B) is made of two Diracs, this convolution product is simple to compute, it corresponds to the sum of two Dirac combs with peaks at $k = n.16$ shifted by +17 and the other by -17. As the signal frequency of the sampled signal must ly in the interval $k \in [-f_E/2, f_e/2]$, only the modes at $k = \pm 1$ are visible.*

leads to non-reversible alteration of the signal. The only way to avoid this issue is to filter by an analogous filter the signal prior to its digitalization. The signal treatment chain will amplify and filter the signal prior to achieve AD conversion. The filter required characteristics are severe and not always easy to fulfill. Let us discuss the CD recording: the sampling rate (is) was 44 kHz for a 16 bits AD converter while we want to record signal having a bandwidth of 18 kHz. 16 bits dynamic corresponds to 1 over 65536 thus nearly 10^5 that is 5 decades in amplitude or 10 in power. The anti-aliasing filter should leave the 18 kHz frequency pass without attenuation, but should remove all the frequencies above. The most dangerous frequency that should be reduced by a factor 10^{-5} is 26KHz, since the Shanon frequency is 22 KHz, 26 KHz will refold to 18 kHz by aliasing. Thus the filter must be very sharp leaving 18 kHz pass and damping by 5 decades a 26 kHz. Building such a filter is a challenge especially at low price. This constrain has been chosen originally to limit the data rate of the CD which was also a challenge when the CD norm was designed. It turned out that the filtering issue has been identified as a quality issue and thus DVD has relaxed the constrain by pushing the sampling frequency to 48 kHz and new audio equipments have pushed this sampling frequency even further. Filtering is clearly the weak point in many acquisition devices, digital scopes, for instance, allow acquiring signals with widely variable sampling frequency. In principle one would require a special filter for each sampling frequency. Most cheap scopes do not provide this feature, they prefer implementing a software filter performing the anti-aliasing filtering with moderate success. One often sees surprising signals on such instrument. A very sharp filter in Fourier's space implies ringing of the impulse response in the real space. This is an annoying property to observe real signal. In fact the best solution is clearly to oversample a signal, if you want a final bandwidth of f_c, you select a sampling frequency of $4f_c$ or even $8f_c$. Then the constrain on the

anti-aliasing filter is relaxed and the impulse response has very little ringing. Notice that this choice is seldom chosen by instrument manufacturers, since it diminishes the bandwidth of their instrument by a factor 2 or 4.

4.3 The special case of image sensors: CMOS and CCD cameras

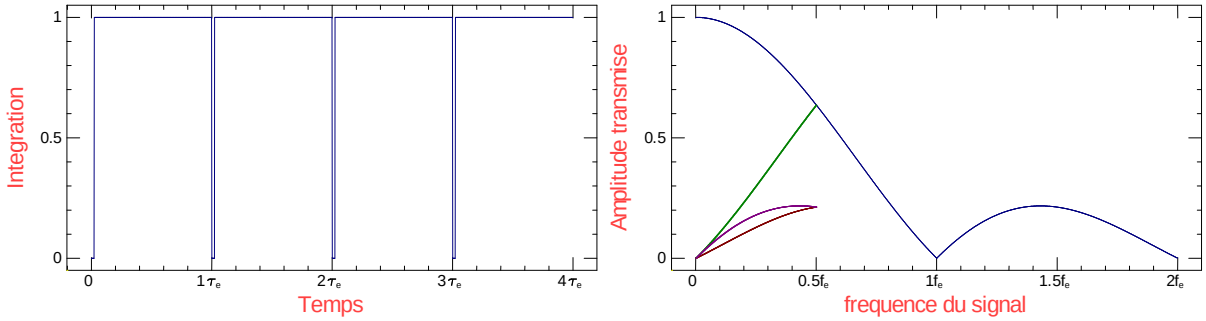

Figure 23: *Time response function and spectrum folding for the case of on a camera with a sampling frequency $f_s = 1/\tau_s$. On the left, digital signal indicating when light integration occurs (integration is done when the signal equal to 1). In fact, the camera integrates light most of the time, the small blind period occurs during the switch between two samples. On the right, the frequency response of the camera (in blue without taking in account refolding). Only frequencies belonging to the interval 0 and $f_s/2$ are accessible. We have drawn the refolding of the blue curve as they occur. In fact there exists other smaller sinc lobs that we have skipped for simplicity.*

Filtering prior digitizing is a complex issue as we have just seen. Cameras are becoming measuring instruments in many situations, they now have millions of pixels that are independent sensors. All these sensors are read at the camera sampling rate which is not very high but keep on increasing with newer equipment. But cameras face two issues in terms of digitization: 1) it is impossible to place a filter after each pixel before converting the charge of the pixel capacity in a number, thus cameras miss completely anti-aliasing time filter. 2) As the image sensor, they spatially digitize the spatial information with a binning corresponding to the pixel size. What is true for time space is also true for spatial one, and spatial aliasing is possible on a camera: if the image contains high spatial frequencies like the strips of the shirt of the speaker on the TV, for instance, this spatial frequency may exceed the spatial sampling frequency of cameras leading to strange ghost images on the TV screen. Cameras are sometimes equipped with a spatial filtering reducing this issue, the idea is simple: the filter low-passes the image thus it blurs the image. Such a filter is placed just above the pixel ship. However, the camera clients are not always pleased by having a filter that blurs the fine details of the image which seems counterintuitive to people eager to obtain images with more and more pixels. Unfortunately this is the only way to avoid this aliasing issues. In practice, a compromise is taken having a small amount of filtering avoid strong aliasing issues but keeping some high resolution. It is easy to observe this aliasing effect even on high quality 4K videos: when the image slowly moves along one direction, the aliasing is very sensitive to the phase difference between the details and the pixel and leads to fast modulations arising in sharp contours features. The time anti-aliasing filter is thus completely absent in all cameras and aliasing must be expected if you measure time dynamics on a movie. The issue is strongly related to the image time integration: if the shutter of the camera is very short, you will freeze fast moving elements but will reduce low-pass filtering, and you are likely to see their effects. The classical example is the wheels of the stagecoach in westerns that often turn in the wrong direction or appear immobile. If you select a shutter exposure time close to the period of the sampling frequency, you average light over this period which minimized aliasing issues but damp the amplitude of an oscillation. In all cases aliasing issues must be considered. There is a case that is worth noting where spatial aliasing is not a problem on a camera: if you observe some elements with an optical microscope at maximum magnification (with the 100x objective for instance). Typically one pixel of your camera will cover 0.1 μm of the real image. As the Abbe relation limits the microscope resolution to $\approx 300nm$, the microscope provides a very good anti-aliasing filter removing all fast spatial frequencies which completely avoid any moiret effect.

5 Some Fourier transform typical applications

5.0.1 Deconvolution of a blurred image

The convolution operation may look weird but it is a very common phenomenon. A simple example is given by a camera picture. If the focus is not perfectly sharp, the light ray coming from a point that are passing through the objective is restricted by its iris leading to a light cone that collapses to point at perfect focus but leading to a small circular dot when

slightly out of focus. A picture slightly out of focus is thus the convolution of the sharp picture by the dot that we have just defined. If we have taken a blurred image and if we know very well the spot shape, that has led to this blurring by a

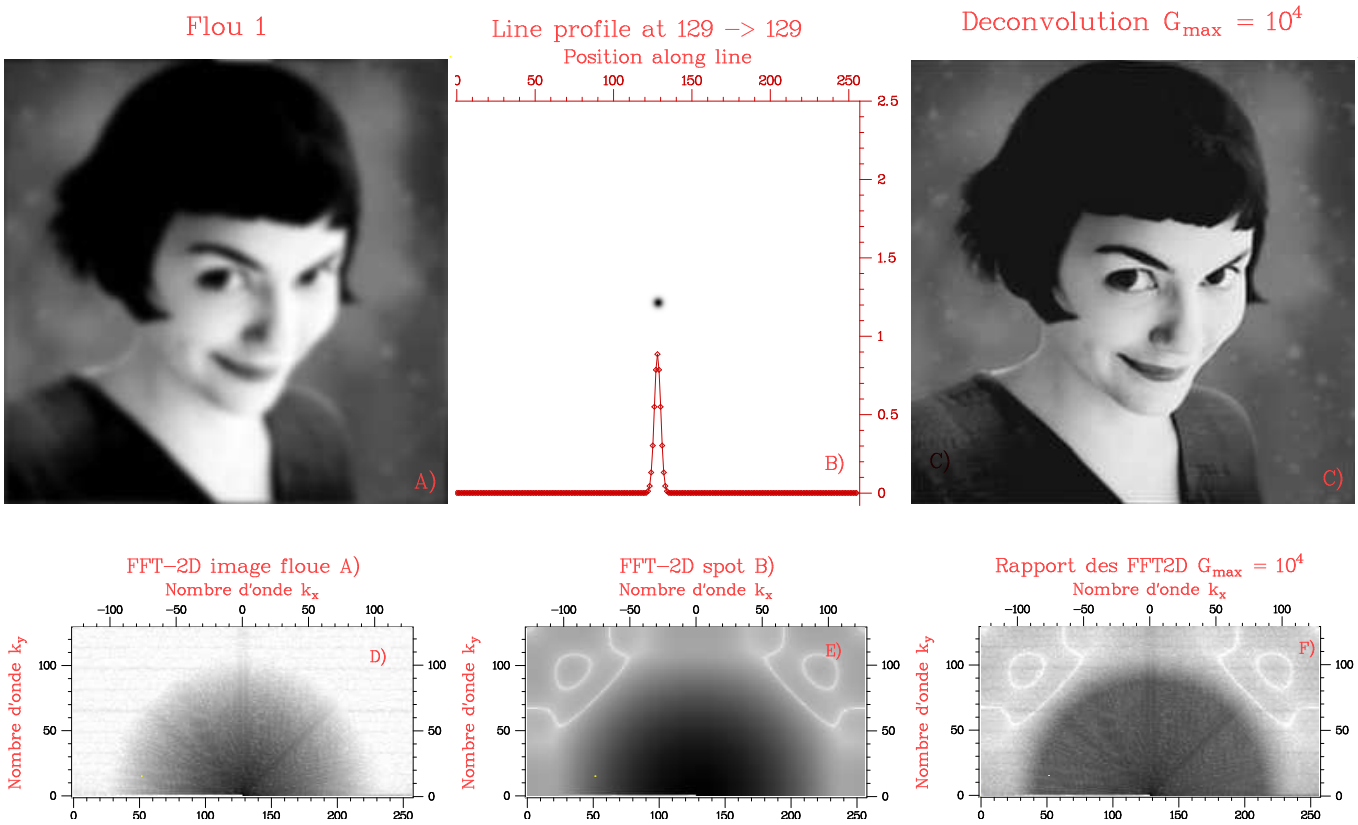

Figure 24: *Principle of deconvolution. On top left, blurred image obtained through the convolution of a sharp image by a Gaussian spot (at the center). To de-convolve, we need to perform the FT of both images. We then divided the FT of the picture by the FT of the spot (bottom right) we then apply the inverse FT to obtain the sharp image (bottom right). During the division, we will reinforce the modes having a high frequency since the amplitude of these modes is low in the FT of the spot. One needs to limit the maximum by which a mode amplitude is multiplied, since the mode may contain only noise that will then become very large. Here we have limited gain too* 10^4.

convolution mechanism (this is the impulse response of our camera picture), we can in principle, de-convolve our blurred image to recover a sharp one. The measure of the impulse response is tricky, to illustrate the method we have made the blurred image by convoluting a sharp one with a Gaussian known spot (on top middle of the figure 24). One obtains the image on the top left of the figure. To de-convolve, one needs to compute the FT of both images (bottom of the figure 24) and to divide the first by the second to obtain the FT of the de-convolved image. It just remains to perform an inverse FT to obtain the de-convolve image which is sharp (on top right of the figure 24). As we have chosen a Gaussian profile to blur the image, its FT is also a Gaussian symmetrical and dividing by this FT is equivalent to amplify the modes having a high frequency of the FT of the blurred image. The convolution by a Gaussian spot is equivalent to have low-pass the image, de-convolution is just applying the inverse filter that is a high-pass filter. This process suffers a limitation, the original image contains signals and noise, and, moreover, the pixels have been digitized using a limited dynamics. Thus, the filtered image may disappear in noise and amplifying the mode too much will not always restore the attenuated signal but reinforce the noise. This means that one should limit the amount of magnification applied in the de-convolution as one can see in the figure 25, the signal of the blurred image is limited at high frequencies by the noise. It is good to define the maximum acceptable gain to avoid noise blowing up. Here we have chosen to keep gain below 10^4.

5.1 Shooting a picture of an object without a lens.

We all have seen a hologram allowing visualizing an object in 3D. To obtain such an image, one proceed in a completely different approach without using any lens to actually image the object of interest. For the hologram, we send monochromatic light on the object and directly on the sensor so that an interference pattern appears between the direct illumination and the light diffused by the object. This demonstrates that the diffused light actually contains some photographic information of the object. If the object of interest just slightly diffuses light, so that the intensity of the incoming light may be considered as unchanged, it si easy to demonstrate that the intensity of the diffused light in the direction $\vec{k_d}$ is nothing

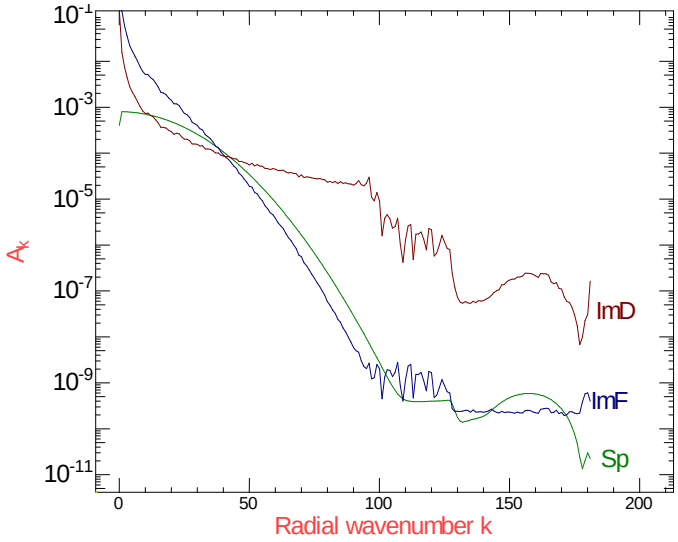

Figure 25: *Radial spectral density of FT of the images of the figure 24. In blue, spectral density of the blurred image, in green spectral density of the Gaussian spot used to blur the image. In red, spectral density of the de-convolved image. We can notice that the de-convolution strongly amplifies the high frequency modes.*

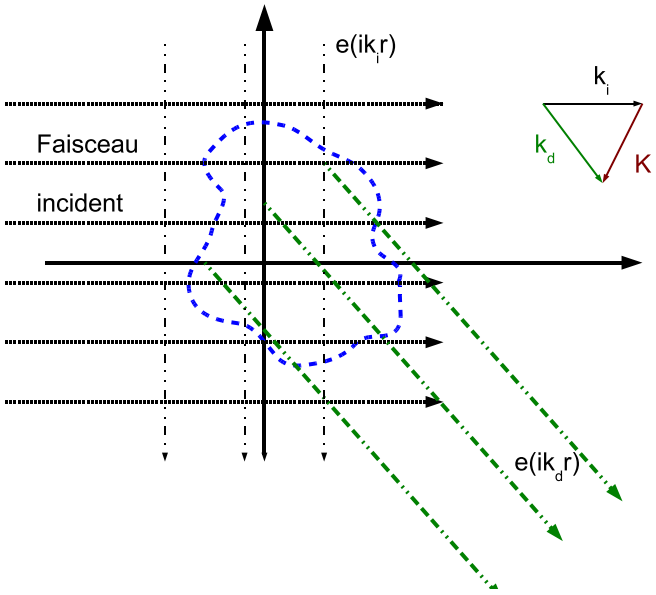

Figure 26: *Diffusion of a monochromatic wave by a semi-transparent object. A plane wave of wave vector $\vec{k_i}$ impact a transparent object which slightly diffuses light (in blue). The wave is diffused in different directions of space. If we consider the direction $\vec{k_d}$, the intensity of the diffused light by each point by $\vec{r}$ of the object suffers a phase shift $\phi = (\vec{k_i} - \vec{k_d}).\vec{r}$. The expression of the diffused light is nothing but the FT of the object with the wave vector $\vec{k_i} - \vec{k_d}$.*

but the Fourier transform of the object with the wave vector $\vec{k} = \vec{k_i} - \vec{k_d}$ where $\vec{k_i}$ is the incoming wave vector. To reconstruct the image, we need to record phase and amplitude of the diffused light within a conic angular direction around the incoming light direction so that we can perform the inverse FT. The aperture of the conic domain of vector $\vec{k_d}$ defines the number of Fourier's mode used. The bigger it is the higher the resolution of the reconstruction. In practice, the method is not used with visible light since it only applies to objects having refractive index very close to 1 and which are fairly transparent (like a smoke volute). On the other hand, this is exactly the situation that occurs for X-ray. For these waves the electromagnetic field has such a high frequency that only the electrons are able to move a little, thus the refractive index is close to 1 and slightly smaller and the object is transparent to X-ray.

5.1.1 Protein structure obtained by the diffraction of X-ray on a crystal.

A very important application of what we have presented on the convolution product corresponds to the protein structure determination using crystal X-ray diffraction. Proteins are polymers of amino acids which are folded in a compact 3D structure owing to the amino-acid interactions. As their size is typically 5 to 10 nm, it is impossible to distinguish these

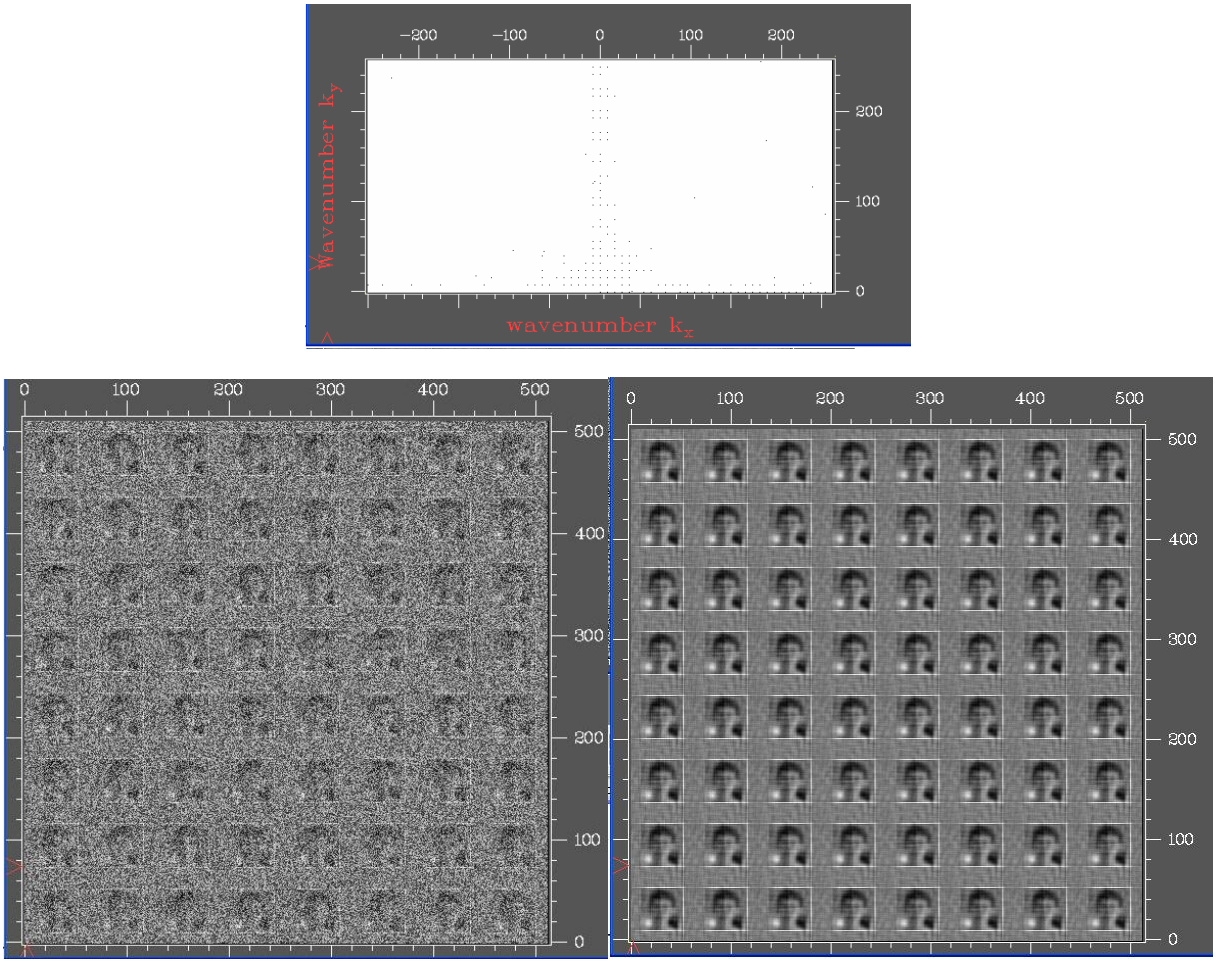

Figure 27: *On the left, the build-up of a two-dimension crystal: a small image is replicated in a periodic fashion in two directions x et y to form a picture of 512 x 512 pixels. We have added numerical noise (to simulate an experiment). On top the FT of this image leads to 512x256 complex modes represented par the log of their amplitude. Many modes are very weak and appear white a few strong modes show up as black dots. These black peaks are organized following a Dirac comb in two dimensions. If we suppress all weak modes and keep only the strong modes, we filter out the noise of the original image. The inverse FT of this filtered image is shown at the bottom right where the original person is nearly recognizable.*

structures using an optical microscope. On the other hand, X-ray has a wavelength smaller than the distance between the atoms of the protein. They are well adapted to build the protein image. In principle, one would just need to place a protein in a monochromatic X-ray beam and measure the diffusion at different angles. In fact, this does not work so simply for several reasons: i) it is not possible to place a single protein in the X-ray without using some material to hold it, this material will also diffuse X-ray, and the diffuse signal will be extremely weak. Ii) if one place a solution of protein in the X-ray beam, all the proteins move in time by Brownian motion and are all in different orientations leading to "powder spectrum" which does not contain enough information to recover the protein structure. On the other hand, in some peculiar conditions, proteins do crystallize forming a periodic structure where all the proteins have exactly the same orientation. If we place such a crystal in a X-ray beam, this time we will observe coherent diffusion occurring along some special directions, according to the angle of the crystal with the beam. These directions are named Bragg direction of the reciprocal space, the crystal periodicity focused diffusion along the Bragg direction like a grating as a sketched on the figure 10. The structure information corresponds to the modulation of the intensity (and the phase) of the peaks. The 3D arrangement of the peaks are dictated by the periodicity of the crystals. If we inverse FT the peak intensity and phase, we obtain the structure of the electronic density of the protein. This allows distinguishing the position of the atoms and finding the protein structure. To achieve the inverse Fourier transform, one needs to have the phase of the Bragg peaks, this information is not simple to get, since one usually observes the modulus of the amplitude of the peak but not its phase. There are tricks to measure this phase which, for example, slightly alter the frequency of the X-ray leads to interference between the two situations which provides a proxy to the phase. To illustrate the principle of this method we have built up a dummy crystal in two dimensions by replicating a small image in both directions and by adding numerical noise.

5.2 Position measurement with subpixel resolution

Abbe has demonstrated that the resolution of a microscope was limited by the wave nature of the light to a fraction of the wavelength. However, this fundamental law is often applied in an unappropriated manner, suggesting that all measurements made with a microscope will suffer this limitation? This is not true, we shall describe a method allowing to measure the position of a micron size object with a nanometric resolution. Our example is the one of the bead tethered by a

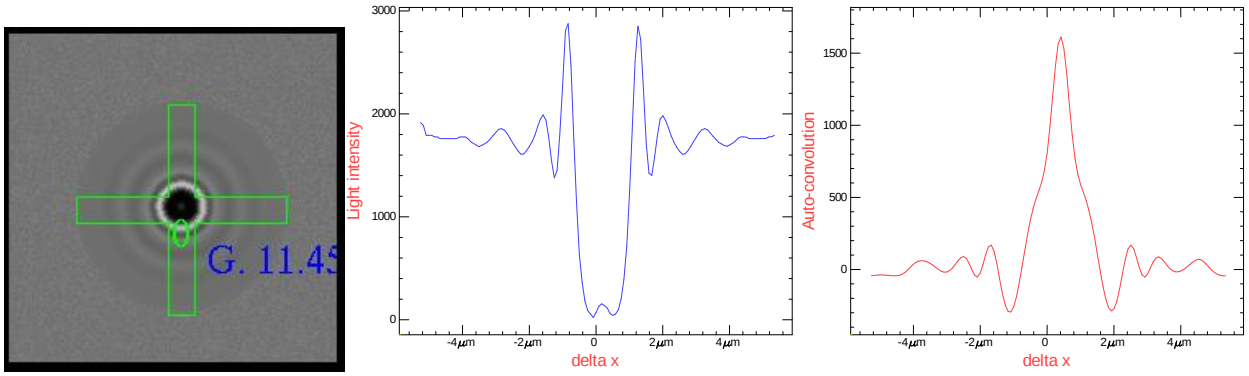

Figure 28: *The algorithm allowing to measure a micron size bead. On the left, the image of the bead and of a superimposed cross allowing the bead tracking. In the middle, intensity profile along X along the horizontal arm of the cross. This profile presents complex intensity modulations variable in time. The bead profile is symmetrical and we notice that here, it is slightly off centered on the right. On the right, auto-convolution of this profile (without its continuous part). This function presents a positive maximum when δx is twice the offset of the profile with its center. By interpolating this maximum by a polynomial we can evaluate its position with an accuracy reaching the nanometer.*

DNA molecule that we have already mentioned. The one micron in diameter bead is clearly visible under the microscope, its image is decorated by diffraction rings and the finite resolution of the microscope limits the visible details on the image. But we are interested by tracking the bead position versus time to record its Brownian motion. But since we are far less courageous than Langevin, we want to avoid doing this tracking manually and prefer to write a program achieving this task. An obvious approach consists in measuring the barycenter of the bead image, as a matter of fact, the gray level of the bead is different from the image background, by defining a threshold it is possible to define a boundary of the bead and thus to compute the barycenter of the pixel inside this boundary. These works, but it is not very robust, changing slightly the microscope focus alters the bead image and strongly affect the result. Unfortunately, the bead own Brownian motion also exists in Z, so that the threshold issue occurs spontaneously. The definition of the threshold is clearly the weak point of this method. The tracking algorithm that we propose to use a cross that will follow the bead as it moves. This cross defines two strips of pixels along X and Y along which we compute profiles averaged in their short direction, as sketched on la figure 28. If we start from a situation where the cross is well centered on the bead, the two profiles present intensity profiles symmetrical around their center. The algorithm measure the off-centering δx of the bead in the cross at image n and correct the cross position for the image $n+1$. To determine this off-centering we use the profile of the figure 28, at the center and we compute the auto-convolution product (done to the profile with its mean value subtracted). This is equivalent to compute the correlation between this profile and the inverted profile (where the first point becomes the last one the inverse). As the bead is symmetrical, this product always displays a positive maximum which position relative to the center of the cross equals twice the off-centering δx as one can see on the figure 28 on the right. By measuring the position of the maximum with subpixel interpolation one determines the bead position with high accuracy. The auto-convolution product has the nice feature of presenting a well-defined maximum whatever the shape of the bead intensity modulation looks like. The algorithm does not depend on an arbitrary threshold and is stable. On the other hand, the FFT provides a fast implementation of the auto-convolution.

5.3 Particle Image Velocimetry PIV

Measuring a velocity field is very useful in fluid mechanics. One classical method is the PIV, it uses small marker particles that follow the velocity field smoothly. The principle consists in following the displacement of those particles from image to the next. The idea is not to track individually each particle, but more to track the motion of small groups of particles. We cut the mage is small square images so that their size is small in comparison of the scale of variation of the flow field but large enough to cover several particles. Then, we correlate each small image with the same small image recorded a faction of time before, in order to determine the direction and shift between the two images. To achieve this process, one computes the correlation between the two images differing by δt. This correlation function presents a positive maximum shifted by $\delta\vec{r}$. The local speed is equal to $\vec{v} = \delta\vec{r}/\delta t$. In practice, one needs to study the contrast modulations of the

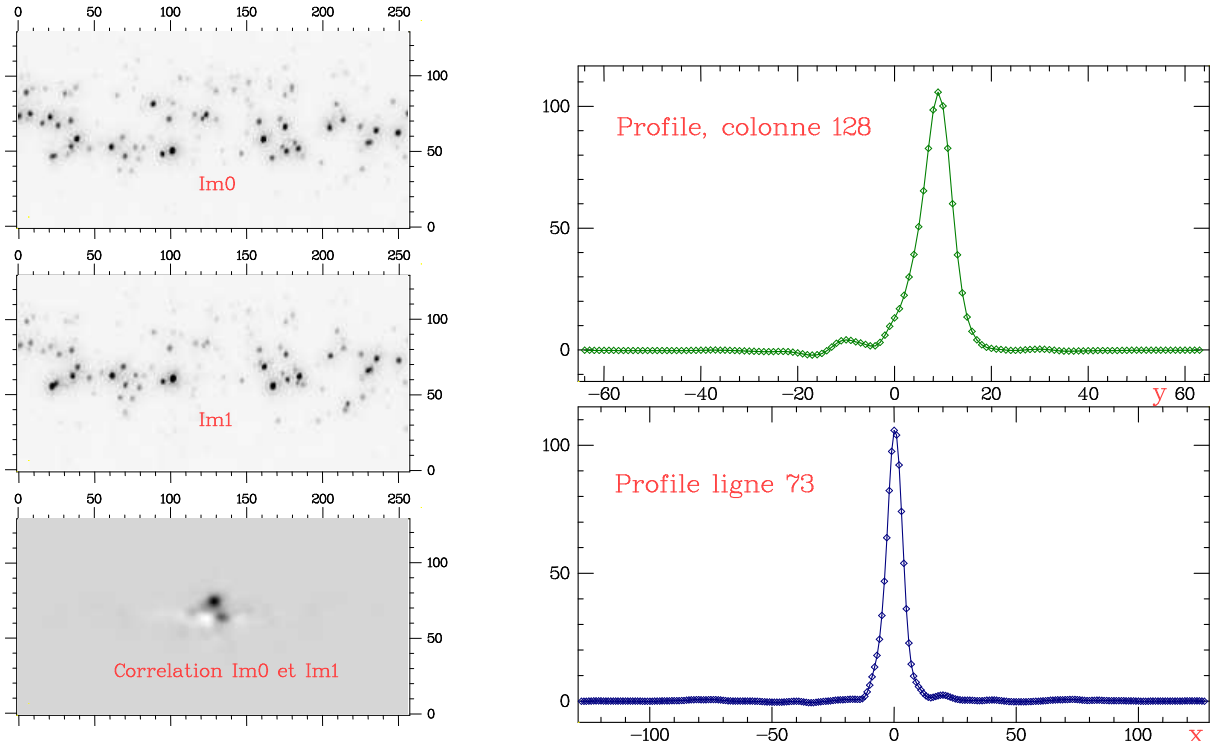

Figure 29: *Principle of PIV. We see here two consecutive images Im0 and Im1 (in inverted contrast) of one micron particles which diffuse the light of a laser beam. The two images have been taken 100 ms apart showing the displacement of the cloud of particles. Here all the particles have the same downward translation speed. If we compute the correlation function of these two images, we obtain the image at the bottom left (shifted by $nx/2$ and $ny/2$). The correlation image presents a positive peak. We show sections of this peak on the curves on the right. This peak is centered along x but presents a shift of 9 pixels along y. Thus the flow speed is essentially along y and equals to 90 pixels/s.*

image which is achieved by suppressing the continuous component before computing the correlation. The position of the maximum is subpixel interpolated.

5.4 Tomography

For medical diagnosis, images have become fundamental, Magnetic Resonance Imaging (MRI) and Y-ray scanner allows obtaining 3D images of the human body. Both methods actually use the Fourier transform. In the case of the MRI (that we shall not discuss), the nuclear magnetization actually detected is directly the Fourier transform of the proton density in Fourier's space following the direction along which the magnetic field gradient is applied. The real image is obtained by inverse Fourier transform of the signal. The X-ray scanner is less obvious, but also interesting. An X-ray picture of the human body corresponds to a projection of the electronic density of the body along one direction on a perpendicular plane. For the sake of simplicity, we assume that the X-ray beam suffers very small attenuation while traveling in the body. The scanner principle is to reconstruct a 3D image starting from a series of projections obtained while turning around the patient with a peculiar axis. Taking one plane perpendicular to the rotation axis, we in fact, reconstruct a 2D image from a set of projection lines. The 3D reconstruction is just the composition of all this 2D images. To illustrate this principle, we shall reconstruct a 2D photographic image starting from a set of projections in 1D. Each point of this projection is the sum of the image pixels in the perpendicular direction. Thus each projection averages the information in the perpendicular direction. If $(\vec{x}, \vec{y})$ is the system in the Cartesian coordinate, it is convenient to describe our projection in the new referential $(\vec{u}, \vec{v})$ where $\vec{v}$ is the projection direction. One can write the projection intensity as:

$$P(u) = \int_{-\infty}^{+\infty} \rho(u, v) dv$$

where $\rho(x, y)$ is the image density. Its FT can be written:

$$\tilde{P}(k_u) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \rho(u, v) e^{-ik_u.u} du.dv = \tilde{P}(k_u, k_v = 0) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \rho(u, v) e^{-i(k_u.u + (0.v))} du.dv$$

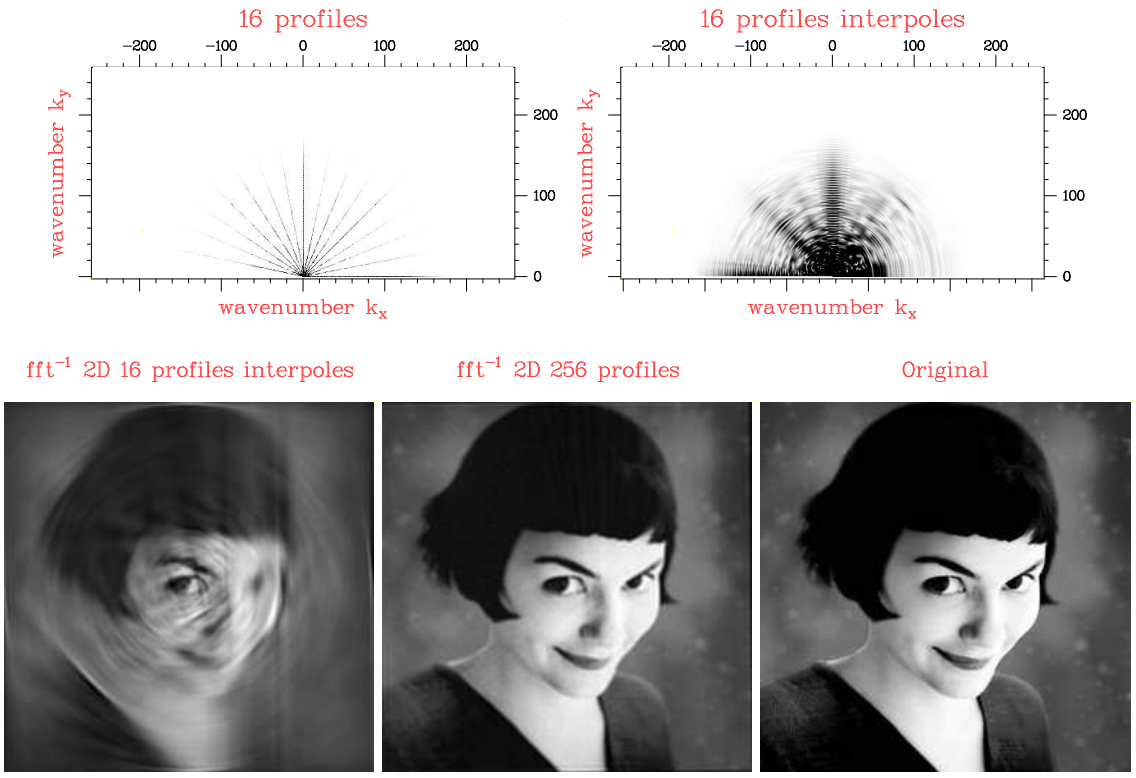

Figure 30: *Principle of the tomography applied to a picture. Here, we shall reconstruct an image (the original one is at the bottom right) by having only access to projections of this image in a set of angular directions. It is possible to show that a projection in one direction leads in Fourier's space to a line passing by the origin and perpendicular to the projection direction. Using each projection, one reconstitutes in Fourier space a set of lines passing by the origin. Of course, the line value is obtained by taking the 1D FT of the profile that is inserted in a 2D Fourier space with the angle perpendicular to the projection direction. We thus obtain an evaluation of the FT as shown in the image on the top left. This evaluation is actually missing many Fourier modes (located outside the lines). A better process consists in performing a change of variables, switching from polar to Cartesian coordinates as shown on top right. If we use only 16 equidistant projection angles, we obtain a limited resolution (bottom center). By increasing the number of projections to 256, the resolution becomes nearly perfect (bottom right) that may be compared with the original.*

One notices that the FT of the profile can be express of the 2D-FT of the image density with $k_v = 0$ and k_u taking whatever value. In short, the profile in Fourier's space passing by the origin $\tilde{P}_{\vec{k_u}}$ is nothing but the Fourier transform of the projection in the real space in the direction $\vec{r_v}$. To reconstruct the image we shall construct the 2D Fourier space by adding these profiles in the 2D Fourier space with the correct angle. Then we perform an inverse Fourier transform to recover the image. Starting from angular projections it is more convenient to use the polar coordinate system to describe the Fourier space. The angular resolution is given by the number of sections. To come back in real space with Cartesian coordinate, we convert from the polar Fourier space choosing a Cartesian resolution higher than the polar one. Then we evaluate each Cartesian Fourier mode by interpolating the polar coordinates in Fourier. We then perform the inverse Fourier transform. The quality of the final image depends of the number of angular projections recorded. Since our image is in fact obtained in polar coordinate, its angular resolution is the same independent of the radius and the image resolution is best around the rotation center. With 16 sections, the resolution is just fine near the center but degrades fast with the radius. Increasing the number of sections to 256 leads to a better image.

5.5 Compression d'image type jpeg

If we save images or video clips in raw format, the size of the generated files becomes very fast prohibitive. A HD frame contains 2 million pixels, in color mode you need 6 megabytes to save a raw picture leading to 150 megabytes per second of HD video. Image compression has been fundamental in the development of numerical imaging. The classical compression algorithms are lossless (GIF, PNG, TIFF...) and use the redundancy of the image information to reduce their weight. If you copy your computer screen displaying graphical elements, many pixels in the same area are alike and PNG compression will replace patches of similar pixels by a single copy and the number of their repeat, this turns out to be very efficient reducing size easily by more than a factor 10. However, if you try to apply this process to a picture the result is

very bad, you gain at most a factor two. The reason behind is just that your picture also contains noise which alters each pixel from their neighbors killing the redundancy feature. To further compress, one must accept to lose some information, the Jpeg's like algorithms allows compressing with ratios 10 to 20 keeping the image quality reasonable. It also offers to choose the quality factor. How Jpeg compression works? The short response is using Fourier space by eliminating modes

Figure 31: *Principle of image compression of a picture with loss using here pure FT process. The original image on the left is Fourier transform, we then remove the Fourier modes having an amplitude smaller than a threshold. If we suppress 90% of the modes having the smallest amplitude, and then inverse FT the result leads to a very acceptable image. This is this property that is exploited in the Jpeg compression.*

having a very small amplitude, and then using a classical lossless compression method to compress the mode having a significant amplitude. In practice, things are a little more involved. The image is cut in small images having 8x8 pixels, these small images are converted in the frequency domain using cosine base (a variant of the FT). The way to eliminate low amplitude modes uses the statistics of all small images choosing the most common pattern. These details are not fundamental. The reason behind the success of the Jpeg compression is that the Fourier components are more specific than the real space, thus by suppressing low amplitude mode in Fourier the algorithm mainly removes noise and little signal. To illustrate this principle of Jpeg compression, we have selected a picture and have Fourier transform the entire image. Before coming back in real space, we have just suppressed (set to zero) modes which amplitude is smaller than a define threshold, thus keeping only strong modes that are in many cases located at low frequency. Upon inverse Fourier transform we see that the image is very good even though we have kept only 10% of the original modes. The process described here is not per se a compression algorithm since in the present form it is missing the way to omit transmitting the mode with zero amplitude.