

Tutorial 4. Fast Fourier Transforms

Phase factors

There are functions that *produce* roots-of-one as a function of time (t) or place (x). A good example is a Bloch function $\phi(x) = \exp(i kx)$ or the phase factor $\phi(t) = \exp(2\pi i vt) = \exp(i\omega t)$. The first is a function of location (x), the latter of time (t). In both cases the function **runs around on the unit circle** we have seen before.

Notice for the latter that there are two conventions for the frequency. If we use ω the factor 2π is usually considered *included* in the frequency ω . (The same holds for the *wave vector* k in the Bloch function). You probably have run into these functions before because they are used a lot in science.

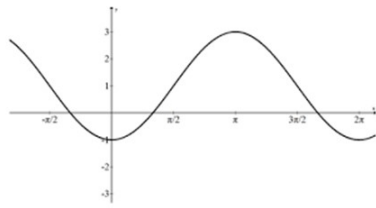
The phase factor is exactly what the name says: If I multiply by such a factor I *leave all magnitudes intact* but I impart a certain phase in the complex plane to my value. Thinking in the complex plane all I do is: rotate along the unit circle, not stretch or contract its radius. This property is the basis for the Fourier transform. If I have a measurement $f(t)$ as a function of time I can analyze it *by frequency* by multiplying with a phase factor $\phi(t; \omega) = e^{-i\omega t}$ and integrating it over all time:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt$$

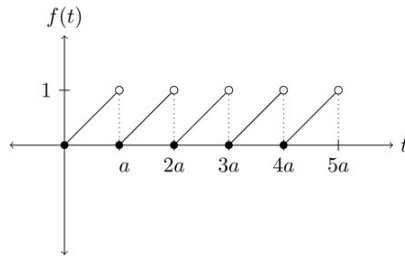
Essentially all you do here is label each measurement with a phase angle leaving all magnitudes intact and see what that gives over your whole data set for a given frequency. In quantum mechanical terms: I am computing an overlap integral to see if my function $f(t)$ **contains** $\phi(t; \omega)$. (Yes, my phase factors are an *orthogonal* set: no overlap between them ever). Another way of looking at it is to think of my data as a moving string frozen in time and now I decompose all motion in its normal modes (like you do with vibrating molecules). Each frequency represents a normal mode.

Wave forms

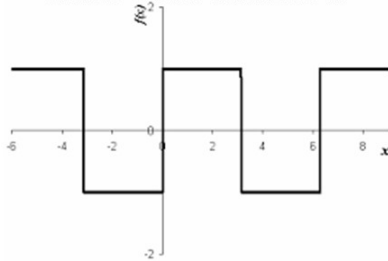
A Fourier series is a decomposition of a repeating wave in terms of sinusoidal functions. A Fourier transform (FT) is the integral representation of the type of decomposition. An FT can also be carried out on a non-repeating wave form e.g. Gaussian or Lorentzian functions. Four important types of wave forms are shown in the figure below for reference as we proceed with the lab.



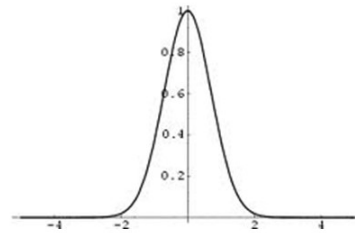
Sinusoidal function



Sawtooth function



Squarewave function



Gaussian function

Fourier transforms

This remarkable decomposition operation is known as **the Fourier transform** and it can be shown that (under certain conditions) the new function $F(\omega)$ (in the frequency domain) contains the same information as the original function $f(t)$ (in the time domain). There is also an inverse operation that brings $F(\omega)$ back to $f(t)$. It involves a very similar phase factor $\phi'(t; \omega) = \exp(-i\omega t)$. Thus Fourier transformation allows you to look at your data in a different way without altering the information content.

Discrete and Fast Fourier transforms

Mathematically the Fourier integral runs from $-\infty$ to ∞ , but data are typically more limited than that. If you take your data at regular time intervals (a constant *sampling frequency*) it is also possible to do a Discrete Fourier Transform, i.e. one where all integrals are replaced by sums:

$$F(\omega) = \sum_{t=-n}^n f_t e^{-i\omega t} dt$$

and there are algorithms to do that fast. The most famous is the Cooley-Tukay algorithm. That is also known as the Fast Fourier Transform (FFT). The crucial thing to understand about the FFT is that it uses the symmetry (think of the unit circle above) to divide the process into two processes, one with even and one with odd terms. Because of this division by two, the number of points must be a power of two. If you do not have a factor of 512, 1024, 2048 etc. you can use zero filling to get the number of points you need. You just add a bunch of points to your data that are all zero.

Computer lab

The data analysis pack of Excel contains a Fast Fourier Transform option based on a famous algorithm, that of Cooley and Tukey. In its original form these authors showed that the fastest way to do a transform on a discrete data set (a Discrete Fourier Transform DFT) is attained if the data set contains a number of points that is a binary power: $n = 2^N$. That is n should be 2, 4, 8, 16, 32, 64, 128, 256, 512, etc. If the number of data points n contains factors of 3, 5 and higher primes it was later shown that transformation is possible but takes more computer time. If n contains large prime numbers the algorithm gets pretty slow, cumbersome and complicated. Excel contains the original algorithm and thus **requires** n to be a binary power. We will use $n=512$.

- Open up spreadsheet **FFTlab** and make sure the macros are active. Excel tends to deactivate them for security reasons.
- Put a 0 in A1 and 1 in A2. Select A1:A2 and put the cursor on the bottom right corner of the selected range until a + appears.
- Now drag the + sign down until you reach A512. The cells should fill with integers and the last one should read 511. This is your time axis (say in seconds). This means that your sampling rate is 1 point per second.
- Type in cell B1: `=A1*2*PI()/512` and Enter.
- Put the cursor on the bottom right corner of B1 until it turns into a +. Then double-click. This should copy and fill the formula down to B512
- As you see from the formula the B column now runs from 0 to 2π in 512 steps. This is handy because we are going to work with sines and cosines
- The cell C1 is where you can experiment with functions. For the moment let's just put in some normally distributed noise. Type in C1: `=NORMSINV(RAND())`. Use the same copy and fill trick to spread the formula over the C range you did before for the B range.
- Now select B1:C512 (Activate B1, hit End; hit Down-arrow; hold down shift; hit Right-arrow; hit End; hit Up-arrow)
- Make a scatter-plot of these data with a solid line only. Should look pretty messy. (All you have is noise.) This is your **time domain plot**
- I made a button for you. What it does is copy the formulas in the C column and past them as values (so that they do not get recalculated all the time and slow the spreadsheet to a crawl). Then it runs the Fourier Transform option of the Analysis Pack (You can also use that directly)
- In I1 type `=A1-1` and then double click to fill I1:I512
- Scroll down till you see E257. As you see it is also a *real* value (the imaginary component is zero). This 'half-way' frequency is known as the **Nyquist frequency** and it represents the sum of all odd points *minus* the sum of all even ones (-+-+ etc.) . That is: the phase angles *alternate* between 0 and π from point to point. This is actually the highest frequency your data provide any information on. If you want to measure something that happens faster you should have measured with a higher sampling frequency than 1

point per second. In that case your data set would be larger, (say 1024 points if you sample twice as fast). In that case the Nyquist frequency would be down at row 513 not at 257.

- If what you study actually fluctuates at a frequency a bit faster than the Nyquist frequency, say at $\omega_{Ny} + \delta\omega$ (i.e. you did not sample fast enough) you will get something known as *aliasing*. A false signal will appear at a frequency $\omega_{Ny} - \delta\omega$ in your analysis. If this happens with a sound recording you get ugly distortions.
- Look at the contents of cells E256 and E258. As you can see they are each other's complex conjugates. The values below the 257th row represent *negative* frequencies and do not contain any new information. The symmetry around the Nyquist frequency comes from the fact that the original data are real numbers. (The symmetry needs to be preserved when operating on the data in the frequency domain otherwise the inverse transform will not yield real numbers).
- Change the value in I258 into -255 and in I259 into -254. Then select the two cells and double-click the bottom right corner to fill. All the way at the bottom the last number should read -1.
- Type in J1: =IMABS(E1)^2. This calculates the value of $|Z|^2 = Z*Z$ of the complex numbers that the FFT produced. This value is known as the *power spectrum* or the *intensity* of your signal.
- Use the bottom corner trick to fill the formula down to J512.
- Make a scatter plot of I1:J512 and make the Y-scale logarithmic (Click on one of the points, right click and go to the format menu). This is your **frequency domain plot**. Note that the plot looks symmetric about 0, which tells you that the information content in the last 256 points is the same as in the first 256.
- The graph should look pretty random, because the transform of random is random again: random (*white*) noise contains all frequencies equally. (Think of white light!)
- Let's do something less random: Type in C1: =COS(B1*4) and do the + trick to fill
- Push the FFT button. Graph A1:E512 (i.e. select columns A and E and then make a scatter plot selecting the line option).
- I recommend that you save this spreadsheet and a new spreadsheet where you type in the function COS(B1*4) again. The spreadsheet before you perform the FFT function can be called LAB4A. Then the spreadsheet after you press FFT you can save the sheet as LAB4A_FFT. The reason for this procedure is that the FFT function replaces the C column with values and the original function is lost. If you do this for each function you will have a record of your work that you can return to. To show that you have done the experiment you could either submit your figures pasted into a PPT document or simply submit the last step in this series of functions.

- Compare the two graphs in the time domain and in the frequency domain. As you see all the frequencies are zero now except harmonic number 4. (see cell J5, but also look at E5).
- Do the same for `=SIN(B1*4)`
- Did anything change? Not in J5, but what happened to E5?
- The problem is that by calculating Z^*Z we have thrown away the *phases* of the complex numbers in the E column (the Fourier components or harmonics) and sines and cosines only differ by a phase shift of $\pi/2=1.5708$. We will not need phase information here, but if you want to extract it you can use `(=IMARGUMENT(E1))`.
- We can add some random noise to the cosine function. To do this type in C1: `=cos(4*b1)+0.3*normsinv(rand())`. Fill the C column and recalculate the FFT.
- As you see the data generated in the C column are a pretty noisy cosine wave, but the fourth harmonic still stands out nicely above an ocean of noise. Its intensity is still about 65000, and its phase is only a little different from zero. All the other harmonics have about the same intensity and their phases are randomly scattered between $-\pi$ and $+\pi$. If we could throw away all harmonics but number four and transform back what would we get?
- Let's take another example. We can call this a linear combination of sinusoidal functions. Type in C1:`=cos(b1)+0.8*sin(2*B1)+0.7*cos(4*b1+0.1)+0.2*sin(8*b1)` and fill the C column. Then save as LAB4B. Then FFT and save as LAB4B_FFT.
- As you see the 'data' in the C column are now a pretty complicated function and you would never have guessed how many components there are just by looking at the graph. The FFT however flawlessly picks up how many components there are, how strong they are and what their phases are. This is the main use of Fourier transforms: **analyze data by frequency**. This is particularly useful if you think of colors of light or pitches of sound.
- Other periodic functions include the sawtooth function. Fourier analysis works for any periodic function. In fact, Fourier series is just a way to decompose any repeating function into a linear combination of sines and cosines. One simple function that we can create is the sawtooth function. Type in C1: `=A1/512` and fill the C column. The function looks like a straight line. However, you must remember that this is a repeating function. At end of the rise it returns to zero and rises again. Call this spreadsheet LAB4C. Study the function with FFT by clicking on the FFT button and call the spreadsheet LAB4C_FFT.
- Change both the x- and y-axes of the frequency domain plot to logarithmic. NOTE: That plot will only show the positive values since you cannot take the logarithm of a negative number. Using this representation the FFT has the appearance of a straight line. Using a Fourier series a saw-tooth function can be written as $f(t)=2[\sin(t)-\sin(2t)/2+\sin(3t)/3-\sin(4t)/4+\dots]=2\sum_{\pm}(\sin(ft)/f)$ This means that the intensities $I=f^2$ should drop off with the

square of the frequency f . Thus, the $\ln(I) = -2 \ln(f)$, which is evident in the log-log plot. The slope is negative because the intensities decrease as the frequency increases.

- Let's look at a squarewave function: `=IF(A1<256,1,-1)`. This time the slope of the double logarithmic plot is -4 (the amplitudes now drop off as the square of the frequencies). However, because the block function is an odd function (antisymmetric around the midpoint) all even harmonics are empty. Thus, the values of the log-log plot have an oscillatory appearance. You can call this pair LAB4D and LAB4D_FFT.
- You can represent a Gaussian function in Excel using C1: `=NORMDIST(A1,244,15,0)`. Fill the C column. This should give a Gaussian peak around $x=244$ with a width of 2×15 . As you see the FFT is also a Gaussian. Notice that the intensity drops to zero pretty fast at higher frequencies. You can call these LAB4E and LAB4E_FFT. Now you have a record of each wave form and the FFT.
- Change the axes of the frequency domain plot to linear-linear if they are not already. Then take the FFT. The Fourier-transformed Gaussian is also a Gaussian, but now it is in frequency space. This is a unique property. The Gaussian is the only function whose FT has the same functional form.
- Do this again but replace the standard deviation `=15` by `5`. What happens in the frequency domain? This domain is often called *reciprocal space*. Why?
- What happens if we put the peak somewhere else: `=NORMDIST(A1,380,5,0)`?
- Let's add some noise to the original Gaussian. Use C1: `=NORMDIST(A1,244,15,0)+ 0.005*NORMSINV(RAND())` and then fill.
- Do you still see the Gaussian in reciprocal space? Where does the intensity in the low frequencies come from? The peak? The noise? Both?

Some examples of FFTs:

1. Scattering of light is essentially mother nature's way of doing a Fourier transform, so all X-ray diffraction is based on it
2. Any regularly sampled 1D data set can be analyzed for its noise spectrum and operation in the frequency domain allow noise suppression and deconvolution (removal of peak broadening)
3. Interferometry is based on inducing path (and thus phase) differences. It is used in e.g. FTIR as an alternative for a grating monochromator
4. Pulsed techniques like pulsed NMR or pulsed voltametry hit a sample with a block wave, i.e. a mixture of frequencies, the response of each of which is unraveled by FFT
5. Mechanical spectroscopy hits samples with block function like deformations, again:FFT.
6. Fourier transforms are a standard trick in solving diff-eqs.

Appendix: Plot functions in Excel

Changing plot limits (also called bounds) in Excel involves finding menu options buried under several layers of menus. The way to find these is illustrated in the figures shown in this Appendix.

