FOURIER ANALYSIS using Python

(version September 2015)

This practical introduces the following:

- Fourier analysis of both periodic and non-periodic signals (Fourier series, Fourier transform, discrete Fourier transform)
- The use of Simpson's rule for numerical integration.

Fourier analysis is an extremely important tool in the investigation of signals of physical origin - essentially it decomposes a signal into constituent harmonic vibrations.   Sometimes the signals are periodic, and the period can be measured directly. But Fourier analysis is also useful for investigating signals which are not periodic. In the first part of this practical you will deal with periodic signals, while in the second part you will see that the numerical method applies with very little modification also to the case of non-periodic signals (when it is known as the Discrete Fourier Transform, or DFT.)

1   Fourier series

Any *periodic* function *f(t)*, with period $T = 2\pi/\omega$, can be represented as a *Fourier series*:

$$f(t) = a_0 + \sum_{n=1}^{\infty}(a_n \cos(n\omega t) + b_n \sin(n\omega t)) \tag{1}$$

The sine and cosine functions are **harmonic** functions, and the series (1) contains a possibly infinite set of harmonic functions with discrete frequencies $\omega_n = n\omega$, $n = 1,2 \ldots$. The frequency $\omega_1 = \omega$ is known as the **fundamental frequency**, and ω_n, $n > 1$, are the **harmonics**.

The coefficients a_n and b_n measure the 'amount' of $\cos(n\omega t)$ and $\sin(n\omega t)$ present in the function *f(t)*. The result of Fourier analysing a signal is a set of values for these coefficients for all n. Obviously in practice the coefficients will be obtained for all n up to some finite maximum *N*.

The Fourier coefficients are evaluated using the orthogonality properties of sines and cosines:

$$\frac{2}{T}\int_0^T dt \, \sin(n\omega\, t)\sin(k\omega\, t) = \delta_{nk} = \begin{cases} 0 & n \neq k \\ 1 & n = k \end{cases}, \tag{2a}$$

$$\frac{2}{T}\int_0^T dt \, \cos(n\omega\, t)\sin(k\omega\, t) = 0, \tag{2b}$$

$$\frac{2}{T}\int_0^T dt \, \cos(n\omega\, t)\cos(k\omega\, t) = \delta_{nk}, \tag{2c}$$

with period $T = 2\pi/\omega$.

If we integrate each of the functions *f(t)*, $\cos(n\omega t)$ *f(t)*, and $\sin(n\omega t)$ *f(t)* over one period, and use equations (1) and (2a)-(2c), we obtain:

$$a_0 = \frac{1}{T}\int_0^T dt \, f(t) \tag{3a}$$

$$a_k = \frac{2}{T} \int_0^T dt\ f(t)\cos(k\omega t) \qquad\qquad k=1,2,\ \dots. \qquad\qquad (3b)$$

$$b_k = \frac{2}{T} \int_0^T dt\ f(t)\sin(k\omega t) \qquad\qquad k=1,2,\ \dots. \qquad\qquad (3c)$$

1.1 Simpson's Rule

A Fourier analysis program must perform the integrations shown in equations (3), for any function f(t) of interest. Simpson's rule will be used for this. (See Kreyzig *Advanced Engineering Mathematics*, Wiley, 7th edn. 1993, chapter 18.5.)

Simpson's rule approximates the integral $\int_a^b dx\ f(x)$ by splitting the interval from x=a to x=b into n steps of equal length h=(b-a)/n where *n* is an even number, and using

$$\int_a^b f(x)\,dx \approx \tfrac{h}{3}\left[f(x_0) + 2\sum_{j=1}^{n/2-1} f(x_{2j}) + 4\sum_{j=1}^{n/2} f(x_{2j-1}) + f(x_n)\right],$$

where $x_j = a + jh$ for *j=0,1,…,n-1,n* with *h=(b-a)/n*; in particular $x_0=a$ and $x_n=b$.

1.2 Problems

1. Write Python code for the computation of the integral $I = \int_0^1 \exp(x)dx$ using Simpson's rule (and n=50 steps). Compare your numerical result with the analytical value for *I*.

2. Write Python code to compute and plot the Fourier coefficents a_k and b_k for the following functions. Note that these functions are chosen so that you can check the performance of your program, as the functions are already in the form of equation (1). (Set $\omega =1$ in the numerical computations.)

   - f(t) = sin t
   - f(t) = cos ωt + 3 cos 2ωt - 4 cos 3ωt
   - f(t) = sin ωt + 3 sin 3ωt + 5 sin 5ωt
   - f(t) = sin ωt + 2 cos 3ωt + 3 sin 5ωt

3. Analyse the square wave with period T = 2π/ω, where θ=ωt and

$$f(\theta) = \begin{cases} 1 & 0 \le \theta \le \pi \\ -1 & \pi < \theta \le 2\pi \end{cases}$$

   (a) Plot the function, compute the Fourier coefficients and compare your output with the analytic result:

$$a_k = 0 \qquad\qquad k = 1,3,5\ldots$$

$$b_k = \begin{cases} 4/\pi k & k = 1,3,5\ldots \\ 0 & k = 2,4,6\ldots \end{cases}$$

b) Plot the reconstructed function, i.e. your evaluation of eqn.(1), using 1, 2, 3, 5, 10, 20, and 30 terms.

4. Analyse the pulse train with period $T = 2\pi/\omega$ and

$$f(\omega t) = \begin{cases} 1 & 0 < \omega t < \omega\tau \\ -1 & \omega\tau < \omega t < 2\pi \end{cases}$$

The analytic results, putting $\omega\tau = 2\pi/\alpha$, are:

$a_0 = (2/\alpha) - 1$

$a_k = (2/k\pi)\sin(2k\pi/\alpha)$ $\qquad\qquad$ $k = 1,2,3\ldots$

$b_k = (2/k\pi)(1 - \cos(2k\pi/\alpha))$ $\qquad\qquad$ $k = 1,2,3\ldots$

Modify the code given for the square wave appropriately. Plot an amplitude-frequency graph, as in 2.

2   Fourier Transform for Non-Periodic Functions

A non-periodic function $f(t)$ may be expanded in terms of cosine and sine functions, but in this case the expansion is a **Fourier integral** over a **continuous range** of frequencies υ, instead of a sum over a discrete set of frequencies $k\omega$.

The Fourier integral may be viewed as the limit of a Fourier series (1) in the limit $T\to\infty$. The summation over n in equation (1) is replaced by an integration over υ,

$$f(t) = \int_0^\infty d\omega\{a(\omega)\cos \omega t + b(\omega)\sin \omega t\} \qquad\qquad (6)$$

Then equations (3a)-(3c) are replaced by

$$a(\omega) = \frac{1}{\pi}\int_{-\infty}^{+\infty} dt\, f(t)\cos \omega t \qquad\qquad (7a)$$

$$b(\omega) = \frac{1}{\pi}\int_{-\infty}^{+\infty} dt\, f(t)\sin \omega t \qquad\qquad (7b)$$

[For details see Kreyszig]

It is frequently more convenient to use complex notation, remembering $e^{i\omega t} = \cos(\omega t) + i\sin(\omega t)$, and define

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} d\omega F(v) e^{i\omega t} \tag{8}$$

$$F(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dt f(t) e^{-i\omega t} \tag{9}$$

(The factors $1/\sqrt{2\pi}$ are conventional.) Then (8) is equivalent to (6), and (9) to (7a) and (7b). The function $F(\upsilon)$ is called the **Fourier transform** of f(t).

If the signal function has the dimensions of energy, then the Fourier transform has the dimensions of power, and its magnitude $|F(\omega)|$ is a measure of the total power in the signal at frequency ω. Note that

$$| F(\omega) |= \sqrt{\Re e(F(\omega)^2) + \Im m(F(\omega)^2)} = \sqrt{\frac{\pi}{2}} \sqrt{a(\omega)^2 + b(\omega)^2} \tag{13}$$

2.1   The Discrete Fourier Transform

In practice, numerical solution of (7a) and (7b), or (9), will involve replacing the integration by a discrete summation (as will numerical solution of (6) or (8).) The exact integrals are *approximated* by a discrete Fourier transform (DFT) as defined below. DFTs are useful for analysing physical amplitude-time or intensity-time data. In such cases it is not known whether or not the signal is periodic, and even if it is, its period is unknown.

Suppose we have a time-dependent physical signal represented by a function *f(t),* and we sample it N times at intervals h, from $t = 0$ to $t = (N-1)h$. (It is usual to use $t=0$ as the lower limit when making physical measurements). Define

$$t_m = mh \qquad\qquad m = 0, 1, 2, ..., N-1 \tag{10}$$

The function is approximated by the discrete set of values at these instants, and time $\tau = Nh$ becomes the period of the approximate function. We need τ to be the longest time over which we are interested in the behaviour of *f(t)*, and we assume

$$f(t + \tau) = f(t), \text{ i.e. } f(t_m) = f(t_{m+N}) \text{ or in shorthand form, } f_m = f_{m+N}. \tag{11}$$

In this case the lowest frequency in the DFT will be $\upsilon_1 = 1/\tau = 1/(Nh)$. This is the fundamental in the case where the function *f(t)* is periodic and $\tau = T$. The frequency spectrum is the discrete spectrum

$$\upsilon_n = n(1/(Nh)) = n\upsilon_1, \text{ for } n = 1, 2, ..., N. \tag{12}$$

The DFT evaluates eqn(8a) and (8b) as

$$F_n = \sum_{m=0}^{N-1} f_m e^{2\pi i \upsilon_n t_m} = \sum_{m=0}^{N-1} f_m e^{\frac{2\pi i m n}{N}}$$

$$\tag{13a}$$

and

$$f_m = \frac{1}{N} \sum_{n=0}^{N-1} F_n e^{-2\pi i \upsilon_n t_m} = \frac{1}{N} \sum_{n=0}^{N-1} F_n e^{-2\pi i m n / N}$$

$$\tag{13b}.$$

[Note that there exists an orthogonality relation for sums that leads to an identity when inserting f_m from (13b) into (13a) and reverse.] F_n are generally complex numbers.

Not all the Fourier components F_n are independent of each other and one can show that $F_{N/2-n} = F^*_{N/2+n}$ where F^*_n is the complex conjugate of F_n. The highest frequency component is thus $F_{N/2-1}$, corresponding to (see eqn.(12)) a frequency $v_n=(N/2-1)/(Nh)) =1/(2h) - 1/(Nh) \approx 1/(2h)$ for large N. This is called the *Nyquist frequency* $v_{Nyquist}$.

If the signal has a component with frequency $v > v_{Nyquist}$ there are less than two sample points per period. In this case there will be one or more frequencies less than $v_{Nyquist}$ for which the amplitude equals the true amplitude at the sample points, and these lower (incorrect) frequencies will appear in the calculated spectrum. This is known as **aliasing**.

The power spectrum (see eqn(13)) of the DFT is given by a plot of all the values $P_n = F_{n,real}^2 + F_{n,imaginary}^2$ as a function of n, where $F_{n,real}$ is the real part of F_n and $F_{n,imaginary}$ its imaginary part.

2.2   Exercises

Modify your Python script for the Fourier series (and save the new version under a different name!) program to calculate the DFT in the following examples, using the following relations for $F_{n,real}$ and $F_{n,imaginary}$,

$$F_{n,real} = \sum_{m=0}^{N-1} f_m \cos(2\pi mn/N)$$

(14a)

$$F_{n,imaginary} = \sum_{m=0}^{N-1} f_m \sin(2\pi mn/N)$$

(14b).

The original signal is then reconstructed via eqn(13b), resulting in

$$f_m = \frac{1}{N} \sum_{n=0}^{N-1} (F_{n,real} \cos(2\pi mn/N) + F_{n,real} \sin(2\pi mn/N))$$

(15)

Remember that the effective fundamental frequency, v_1, to be used in each case is determined by the total sampling time τ. Make your program evaluate and print out the sampling rate v_s as well as v_1 so that you can see how many Fourier components you would expect to be able to calculate accurately.

1. Derive eqn. (15) from eqn. (13b).

2. Consider the function $f(t)=\sin(0.45\pi t)$.
   - Set $N=128$ and $h=0.1$ and plot the function for a total time $\tau=Nh$, together with the points where it is sampled.
   - Write code that returns the Fourier components $F_{n,real}$ and $F_{n,imaginary}$ using eqns (14a) and (14b) and plot them as a function of n.
   - You should find one dominant Fourier component $f_n=n/(Nh)$. How does its value compare with what you expect for the frequency from the given function $f(t)$ above?
   - An ideal sampling time would contain a multiple of the period of the periodic signal being probed. Use this criterion to pick an "ideal value" for h, keeping $N=128$ fixed and perform your Fourier calculations. What do you see?

- Compute the Fourier components $f_{n,real}$ and $f_{n,imaginary}$ and plot these as a function of time $t_m = mh$ to reconstruct the initial function $f(t)$. Include $f(t)$ in this figure to see how well the reconstruction works.
- An ideal sampling time would contain a multiple of the period of the periodic signal being probed. Use this criterion to pick an "ideal value" for h, keeping $N=128$ fixed and perform your Fourier calculations. What do you see?

3. Treat the function $f(t)=cos(6\pi t)$ as one of unknown period, and sample it every second for time $\tau=Nh$ (N=32, h=0.6). Record the sampling frequency, and plot the spectrum (real and imaginary part of F_n) and power spectrum as a function of n (ranging from 0 to N-1). Repeat the exercise with $h=0.5$, $h=0.4$, and $h=0.1$, keeping $N=32$ fixed. Explain your results in the three cases. (Hint: in each case, calculate the function at each of the sampling instants.) Choose a sampling rate and total sampling time which will produce an exact result for the Fourier transform, i.e. in this case only one Fourier component will be finite.

4. Consider the function $f(t)=sin(24\pi t)$. Set $N=64$ and $h=0.1$ and perform the Fourier transform. What frequency component do you obtain? Now set $N=32$ and $h=0.2$ and compute the resulting frequency component? Explain your result by computing the Nyquist frequency $f_{Nyquist} = 1/(2h)$.

5. Treat the function $f(t)=cos(3t)$ as one of unknown period, and sample it every second for 8 seconds ($h=1$, $\tau=8$). Plot the coefficients as a function of frequency, and the power spectrum. Repeat the calculation, sampling the function every second for a total sampling time τ of 16, 32 and 64 seconds. Compare the spectra obtained. What went wrong in the first calculation? (This is known as leakage - the amplitude leaks into nearby frequency bins.)

6. Evaluate the DFT for the damped sine wave: $f(t) = sin\, t\, e^{-kt}$. For a suitable value of k (e.g. 0.2) choose τ long enough for the signal to have decreased to at least $1/e$ of its initial value.

- Make a plot of the function $f(t)$ which also shows the sampled values.
- Plot real and imaginary parts of F_n as functions of n, and also the power spectrum.
- Reconstruct the function from the DFT, using equation (15), to check how good an approximation it is.

7. Evaluate the DFT for a single pulse.

$$f(t) = 1 \quad 0 < t < \tau$$
$$f(t) = 0 \quad t < 0, t > \tau$$

Compare the power spectrum with that obtained in problem 4 of section 1.2. Notice that increasing N for a fixed total time interval τ (i.e. decreasing h and increasing N so that Nh is unchanged) increases the number of lines in the frequency spectrum, so that in the limit $N \to \infty$, $h \to 0$, the spectrum would become continuous.

8. Calculate and plot the power spectrum of the signal represented by the data file *sampledata.dat*. Can you guess what this data file represents?