PHYS 501: Mathematical Physics I

Fall 2020, Homework #7

(Due December 12, 2020)

1. The functions $f(t_k)$ and $F(\omega_n)$ are discrete Fourier transforms of one another, where $t_k = k\Delta, \omega_n = 2\pi n/N\Delta$, for $k, n = 0, \ldots, N-1$. Show that

   (a) if f is real, then $F(\omega_n) = F^*(4\pi f_c - \omega_n)$,

   (b) if f is pure imaginary, then $F(\omega_n) = -F^*(4\pi f_c - \omega_n)$, where $f_c = 1/2\Delta$ is the Nyquist frequency.

2. (a) Let $\{r_k\}$ be a random sequence of real numbers, with r_k distributed uniformly between -1 and 1. For the N-point discrete Fourier transform of $\{r_k\}$

$$R_n = \sum_{k=0}^{N-1} r_k \, e^{2\pi ikn/N},$$

   calculate analytically the expectation value and variance of the "periodogram estimate" of the (unnormalized) power spectrum

$$P_n = \frac{|R_n|^2 + |R_{N-n}|^2}{N^2},$$

   for $n = 1, \ldots, N/2$.

   (b) Generate a sequence of random numbers (using, for example, the `numpy.random.random()` function in Python), with properties as in part (a), and compute $\{P_n\}$ numerically using a fast Fourier transform with $N = 65536$. Plot both P_n and $\log_{10} P_n$ against $\log_{10} n$, for $n = 1, \ldots, N/2$.

   (c) For both plots in part (b), average the plotted data over an interval of width 65 in n centered on each data point, and plot the averaged result on top of the raw transformed data.

   How do your results compare with analytic expectations?

   (d) Repeat the computation in parts (b) and (c) for a *random walk* $\{w_k\}$ defined by $w_0 = 0$, $w_{k+1} = w_k + r_k$. Can you account for the differences in appearance between this graph and one obtained previously?

3. A "corrupted" real-valued dataset may be found in the file `corrupt.dat` on the Learn page. It is a time sequence consisting of two columns of data, k and c_k, for $k = 0, \ldots, N-1$. The original data have been convolved with a Gaussian transfer function of the form $g_k \sim \exp(-k^2/a^2)$ (normalized so that $\sum_k g_k = 1$), with $a = 2048$, and are subject to random noise of some sort at some level.

   Find a filter to apply to the data, and plot your best-guess reconstruction of the original uncorrupted dataset. Rather than attempting to characterize the noise in detail and applying an optimal filter as in *Numerical Recipes*, it will be sufficient simply to truncate the data in

Fourier space at some frequency dictated by the form of the power spectrum you obtain. Turn in (i) your program, (ii) a graph of the untruncated power spectrum, clearly indicating where you chose to truncate (and why — hint: create a log-log plot of the power spectrum), and (iii) a plot of your final "uncorrupted" time sequence. Don't forget to truncate the negative frequencies!

Based on what you found in problem 2, can you characterize the type of noise in the data?

4. The file `pendulum2.dat` on the Learn page contains a chaotic data set generated as a solution to the equations of motion of the damped, driven, nonlinear pendulum:

$$\frac{d^2\theta}{dt^2} + \frac{1}{q}\frac{d\theta}{dt} + \sin\theta = g\cos\omega_d t.$$

The four columns contain t, θ, $\omega = d\theta/dt$, and $\phi = \omega_d t$. In this system, the natural frequency (for small θ) is 1, and ω_d is the driving frequency. (For definiteness, we have taken $q = 2$, $\omega_d = 2/3$, $g = 1.35$, with initial conditions $\theta(0) = 0$, $\omega(0) = 1$.) The time sequence starts at $t = 500$, by which time any transients should have damped away.

(a) Plot the time sequence $\theta(t)$ for $1000 \le t \le 1250$.

(b) Use an FFT to compute the power spectrum $P(f)$ of $\theta(t)$, where f is frequency. Use the entire dataset, with a Bartlett data window, and plot $P(f)$ with log–log axes for $0.01 \le f \le 2$. Can you identify any features in the plot?

(c) As in Problem 2, smooth the data by averaging over an interval of width 129 centered on each frequency data point, and plot the results as in part (b).

(d) Implement the alternative smoothing strategy of dividing the input dataset into a series of slices each of length 32768, computing the power spectra of each, and then averaging all the individual power spectra. Again plot the results as in part (b). Don't forget the Bartlett windows!