Exercise 1.

Let $\{e_1, \cdots, e_n\}$ be a set of orthonormal functions, and $\{e_1, \cdots, e_n, e_{n+1}\}$ the same set augmented with the additional function e_{n+1}, orthonormal to all the others. Let f be a square–integrable function and f_n, f_{n+1} the projection of f into, respectively, $\mathrm{span}\{e_1, \cdots, e_n\}$, and $\mathrm{span}\{e_1, \cdots, e_n, e_{n+1}\}$. (In case you have forgotten, this means $f_n = \sum_{i=1}^{n} \langle f, e_i \rangle e_i$, same for f_{n+1} except that the sum extends up to $n+1$).

Prove that

$$d(f, f_n) \geq d(f, f_{n+1})$$

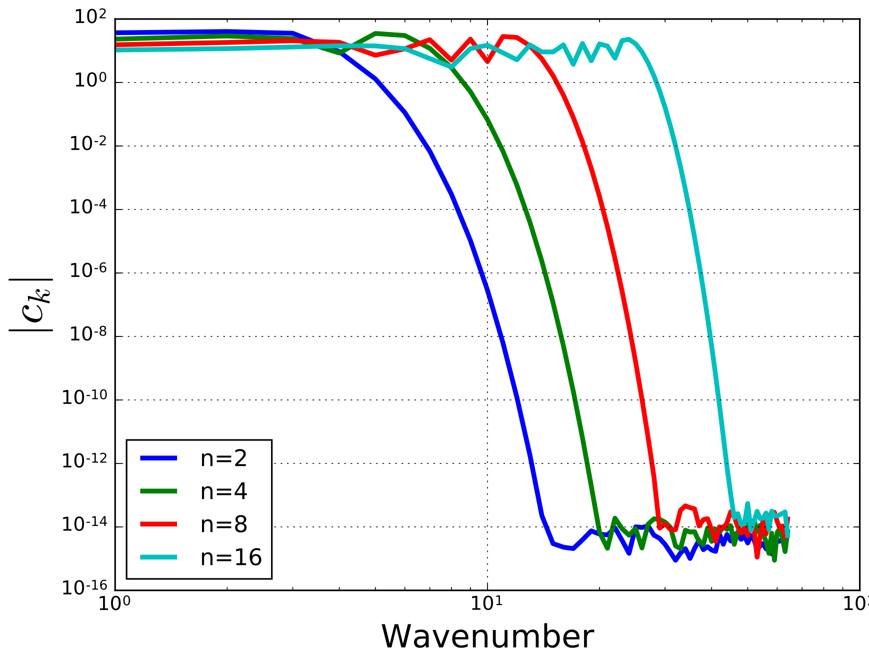
(remember that $d(a, b) = \sqrt{\langle a - b, a - b \rangle}$).

Exercise 2.

Regardless of the value of n, for $\theta \in [0, 2\pi)$, the following is a periodic function with infinitely many periodic derivatives.

$$f(\theta) = \cos\left(n \sin\left(\frac{\theta}{2}\right)\pi\right)$$

Write a program that computes the modulus of the Discrete Fourier Transform of f and plots it as a function of the wavenumber using log–log axis for $n = 2$, 4, 8, 16. The result should look something like this:



These graphs show that this function has coefficients with a magnitude roughly constant until a certain critical wavenumber k_{cut}, and, for higher values of k, in accordance with the theory of Fourier series, the magnitude of the coefficients decays faster than any power of the wavenumber k (this is easy to detect in a log–log plot, because anything proportional to a power of k would be represented as a straight line having a slope equal to the exponent of k).

Write a text (no more than 1/2 page) containing an argument that explains the dependency of the critical wavenumber k_{cut} on n. By "an argument" I mean neither a formal and rigorous proof, nor simply a fit of numerical results. I mean some believable mathematical reasoning that makes sense of what the numerics is showing to you.

Note 1: feel free to experiment with several other values of n.

Note 2: to produce the figure I have discretized the interval $[0, 2\pi)$ in 128 equally spaced grid points (in python, this means `theta=arange(0, 2*pi, 2*pi/128)`).

Note 3: in python, the DFT of real data is computed by the python function `rfft`; the element–wise modulus of a complex array is computed by the function: `absolute`.

Note 4: The flattening of the curves on the right–hand side of the plot is due to round–off errors in the computation of the DFT: because the computation of the coefficients y_k is interdependent for all values of k, when the difference between the smallest true y_k and the largest true y_k is larger than the machine accuracy, the round–off errors magnify the smallest y_k until the difference between this and the largest is of the order of the machine accuracy (≈ 16 decimal digits). So, in your argument, just disregard that flattening.

Note 5: plotting the function and looking at its graph for various values of n should help a lot in coming up with the correct idea....

Exercise 3.

Write a python function that computes the derivative of a periodic function f, of period 2π by taking the derivative of the DFT of f sampled at equally–spaced points in $[0, 2\pi)$. Use the python function to compute the derivative of

$$f(\theta) = \cos\left(4\sin\left(\frac{\theta}{2}\right)\right)$$

Compute the derivative by hand and plot the maximum absolute error as a function of the number of points. Use 8, 16, 32, 64, 128, 256, 512 points to discretize the interval $[0, 2\pi)$ (in python that's `2**arange(3,10)`). At the highest resolution, how many correct decimal digits would you expect to have with a second–order finite–differences method? And with a fourth–order one?

Exercise 4.

Repeat the exercise 3, but using the function

$$f(\theta) = \frac{1}{1 + (x - \pi)^2}$$

Look at the error. Explain why the result is so terrible.