**Assignment 3** (Turbulence and spectra).

For various applications in wind energy, time-series of wind are analyzed, treated, or simulated via spectra. In this assignment we work with measured time-series, as well as synthesizing turbulent time-series.

The assignment can be divided into four main categories, which are all somewhat related. The focus of each category can be described in terms of the data involved:

**Data 1**: Time-series of turbulent velocity measurements from a three-dimensional sonic anemometer at Høvsøre, at a height of 80 m. The file `hoevsoere1hr_uvw.csv` contains 1 hour of data sampled at 20 Hz. Its three columns are $\{u, v, w\}$, where $u$ is in the mean horizontal wind direction and $w$ is vertical.

**Data 2**: Time-series of 10-minute-mean wind speed measurements from Sprogø. Use the data in `sprogo_1.zip` (which you've also used in Assignment 1).

**Data 3**: Data you simulate, from simple linear Markov process. See below for details.

**Data 4**: Data you synthesize via Fourier Simulation. See below for details.

Do the following:

1. **Data 1**:
   (a) Calculate the mean value and variance for each of the (three) velocity components.
   (b) Calculate the friction velocity, $u_\star$.
   (c) Calculate the temporal spectrum for the three wind components, both directly (unsmoothed) and via ensemble averaging: chop the time-series up into 1 and 6 equally long sub-series (corresponding to 1 hour and 10 minutes, respectively), and ensemble average the six 10-minute spectra. Plot both the 'raw' (unaveraged) and averaged spectra as $S(f)$ vs $f$ in a log-log coordinate system.
   (d) Write a short code to make a smoothing filter that averages a spectrum over neighboring frequency bins with logarithmic separation, so that at most $n$ spectral estimates appear per frequency decade. (Note: for practical use $n$ should be between ∼10 and 20.)
   Apply the smoothing filter you just made to the 'raw' (unaveraged 1-hour) spectra calculated in the previous question, and plot the theoretical prediction for the inertial sub-range (the $^5\!/_3$ law) in the same plot; i.e., compare the power law exponent (slope in a log-log coordinate system). Also check quantitatively whether the $^4\!/_3$ law is observed.

2. **Data 2**: Calculate the frequency spectrum of the 21-year time-series of 10-minute averages from Sprogø, used in Assignment 1. For simplicity you can ignore any gaps in the data time-series, and just treat all wind speeds as one long time-series with constant sampling rate $f_s = 1/(10\text{min}) = \frac{1}{600}$ Hz.
   (a) Plot the spectrum as $fS(f)$ vs $f$, using appropriate axis scaling.
   (b) Applying your smoother to the spectrum, can you see a diurnal peak or an annual peak? I.e., what timescales have the highest peaks?

3. **Data 3** is made by you! It is the result of a simple auto-regressive process. In such a series the $(n+1)^{\text{th}}$ number is obtainable from the $n^{\text{th}}$ value via

$$X_{n+1} = c_{ar} X_n + \xi_n, \tag{2}$$

where $0 < c_{ar} < 1$, and where $\xi_1, \xi_2, \xi_3 \dots$ are independent Gaussian random variables with zero mean and unit variance. By choosing an initial value, $X_1$, you can thus generate all future values $X_n$. Make a computer program that simulates such a process, where the initial value $X_1$ is normally distributed with zero mean and variance $\sigma_X^2 = 1/(1 - c_{ar}^2)$.

(a) Simulate three *long* time-series ($N \geq 10^4$, using $c_{ar} = 0.1$, 0.4 and 0.9 respectively (you can arbitrarily pick whatever $\Delta t$ that each step corresponds to, e.g., 1 s or 0.1 s).

(b) Calculate the mean and variance of the three time-series just simulated in 3(a).

(c) For the auto-regressive process the autocorrelation function is given by $\rho(\tau) = c_{ar}^{\tau}$; plot $\rho(\tau)$ for each of the three time-series, and compare with this expression.

(d) Calculate the spectra for the three time-series and plot in log-log coordinates.

(e) What is the power-law exponent at the highest frequencies, and should it depend on $c_{ar}$? *[Hint: deriving it is more reliable than trying to get from plots]*

(*) Bonus: derive the integral timescale $T_{\text{int}}$; re-express $c_{ar}, \rho(\tau)$, and $\sigma_X$ in terms of $T_{\text{int}}$.

4. You make **Data 4**, using the Kaimal spectrum. The latter is defined by

$$S_u(f) = u_*^2 \frac{52.5z/U}{(1 + 33fz/U)^{5/3}}. \tag{3}$$

The recipe for generating a time-series $x_n$, having a given spectrum (in this case the Kaimal spectrum) is given below.

- Choose a record length, $N$.
- Make a list $X_\ell$ of random complex numbers, of length $N/2$. Using a zero-mean Gaussian distribution with unit variance, make a random number for both the real part and the imaginary part. This means that you should generate a total of $2 * N/2 = N$ random numbers.
- Multiply each member ($X_\ell$) with the correct Fourier amplitude $\sigma_\ell$; the latter is specified by the Kaimal spectrum.
- Expand $X$ using the index rules given in the course notes:

$$\overline{X} = \left\{ \overset{\boxed{1}}{a_0}, \overset{\boxed{2}}{a_1 + \imath b_1}, \overset{\boxed{3}}{a_2 + \imath b_2}, \overset{\boxed{4}}{a_3 + \imath b_3}, \dots, \overset{\boxed{\text{N/2+1}}}{a_{N/2} + 0}, \dots, \overset{\boxed{\text{N-1}}}{a_2 - \imath b_2}, \overset{\boxed{\text{N}}}{a_1 - \imath b_1} \right\} \tag{4}$$

  Use a built-in function for complex conjugation to make this easier (e.g. `conj(x)` in Matlab or `numpy.conjugate` in python).
- Inverse Fourier-transform $X$, making sure it is normalized correctly. If the resulting time-series is not real-valued (i.e., has a significant imaginary part), then you did not do the previous step correctly.

Now, do the following:

(a) Make a program that generates $u(t)$ using this 'recipe' with the Kaimal spectrum (3) at a height of $z = 80\,\text{m}$, under the conditions $U = 7.0\,\text{m}\,\text{s}^{-1}$ and $u_* = 0.25\,\text{m}\,\text{s}^{-1}$. Use it to generate a 10-minute $u(t)$, with $\Delta t = 0.05\,\text{s}$; plot $u(t)$.

(b) Calculate and plot the spectrum from the synthesized time-series to check that you get the Kaimal spectrum back.