Python G3 - DAD

TP2

Useful packages include those of `SciPy` (say "saïe-paï"):

- **NumPy**'s array type augments the Python language with an efficient data structure useful for numerical work, e.g., manipulating matrices. NumPy also provides basic numerical routines, such as tools for finding eigenvectors.
- **SciPy**: additional routines needed in scientific work for computing integrals numerically, solving differential equations, optimization, and sparse matrices... `https://docs.scipy.org/doc/scipy/reference/tutorial/index.html`
- **matplotlib** module: to produce high quality plots. With it you can turn your data or your models into figures for presentations or articles. No need to do the numerical work in one program, save the data, and plot it with another program.

Numerics:

- Variables of type `string` (within "" or '...') and `tuple` (within parentheses) are not modifiable.
- Variables of type `list` (within [...]) are modifiable.
- In practice, we will mostly work with matrices and vectors **using `numpy` arrays**.
  $\longrightarrow$ `https://docs.scipy.org/doc/numpy-dev/user/quickstart.html`
- For a tutorial on `SciPy` go to
  $\longrightarrow$ `https://docs.scipy.org/doc/scipy/reference/tutorial/index.html`

Graphics:

- `matplotlib` is my friend ! it is 'matlab like'...

  Have a look at the 1st tutorial:

  $\longrightarrow$ `http://matplotlib.org/users/tutorials.html`

Graphics & Numerics:

- `import matplotlib as plt` # with prefix `plt.` like in `plt.show()`
- `import numpy as np` # with prefix `np.` like in `np.array(...)`

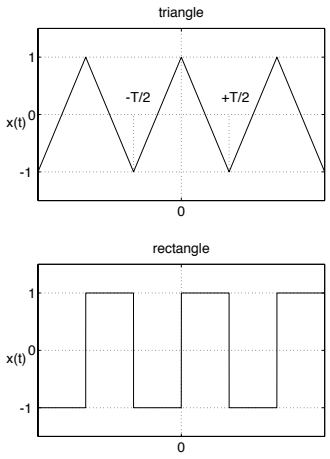
or within a notebook, you can invoke either:

- `from pylab import *`
- the magic command: `%pylab`

which imports `matplotlib` and `numpy` and `scipy` with no prefix (neither `plt.` nor `np.`) in front of commands. **Be careful:** this last method can generate conflicts !

Exercise 1 *Fourier series reconstruction*
We study the Fourier series decomposition of two signals.

1. Generate the reference signals and reproduce the following figures :



Useful : `plt.plot, plt.xlabel, plt.ylabel, plt.title, plt.text, plt.subplot, plt.show()...`

2. The Fourier series of these signals are given by:
- for the triangle :

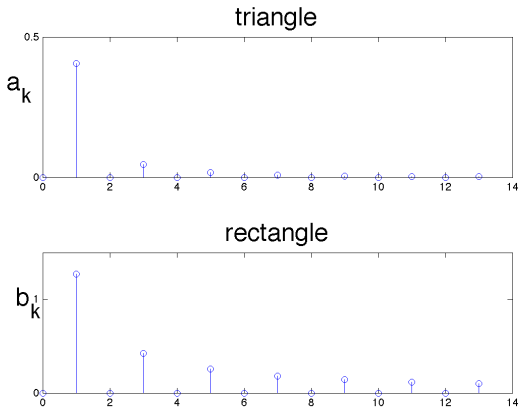
$$x_{\Delta}(t) = \sum_{k=0}^{\infty} \frac{8}{[\pi(2k+1)]^2} \cos((2k+1)\omega_0 t)$$

- pour le rectangle :

$$x_{rect}(t) = \sum_{k=0}^{\infty} \frac{4}{\pi(2k+1)} \sin((2k+1)\omega_0 t)$$

Compute the 14 first coefficients as below: [use *comprehensions* or `def` and `map`]

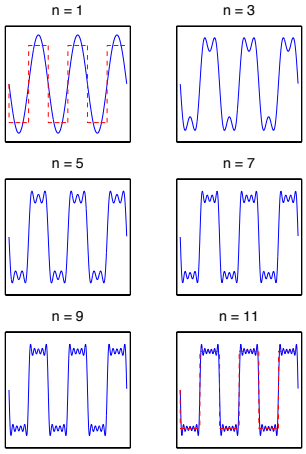
3. Display these coefficients as below:



Useful : `plt.stem, plt.tight_layout()...`

4. Save this figure in a file by using `savefig(fillename)...`

5. Illustrate the convergence of Fourier series by showing the evolution of 6 partial sums as subplots of the same figure for the rectangular signal as below:



Useful : `for, while, plt.subplot, plt.plot(x,y,'r--',linewidth=2)...`

Exercise 2 *Basic statistics*
The following table gives the infant mortality (X) and the gross national product per inhabitant (Y) of 12 european countries:

X	190	128	180	212	56	192	68	98	110	197	181	233
Y	24	28	24	19	37	22	34	25	36	24	20	18

1. For X and Y, compute the median, mean, variance and standard deviation (`median, mean, var, std`)

2. Give the equation of the regression line of Y as a function of X (`np.polyfit, np.polyval`).

3. Display the cloud of points and the regression line on the same figure.
Remark: explore how to represent points by colored stars, squares, circles...

Exercise 3 *Histograms & distributions*
We will check whether the function `randn` actually throws random numbers according to the normalized Gaussian probability density function:

$$p(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

1. Generate a large number of samples using `np.random.randn(...)`.

2. Estimate a histogram of this distribution for a well chosen set of bins and represent it (`np.histogram, plt.hist,...`).

3. Compare this histogram to the theoretical distribution $p(x)$.
Indication: you should think about what a histogram is and how it relates to the probability density function.