Data Processing Problems

2023-2024

5N-027-SCI / DPP_NT

DPP_Pb_S5_01 (10%)

Keywords

Data Processing ;

Sessions

0 Course(s) - 1h30
0 Directed Session(s) - 1h30
1 Computer Session(s) - 2h00
0 Practical Session(s) - 4h30

Working by **team of 2**

Institut d'Optique
Graduate School, *France*
https://www.institutoptique.fr

Displaying scientific data

Producing **scientific reports** with **smooth and clear graphics** is a large part of the life of an engineer or a researcher.

In this session, you will have to :

- generate discrete sinewaves at different frequencies - from a period (or frequency), a number of periods and a sampling frequency
- plot those signals depending on a time vector (or different time vectors)
- fill the legend, the axis labels of the graphics

Intended Learning Outcomes

By solving this problem, students will be able to :

1. **translate a mathematical function** into a programming function
2. **produce scientific smooth and clear graphics** from a mathematical function - including titles, legends and axis labels - with *Matploblib.pyplot* Python library

Deliverables

At the end of the session, you have to produce :

1. **commented functions** to generate specific signals (such as a sinewave) from a time vector or other parameters (sampling frequency, frequency of the signal, number of periods)
2. graphics with titles, legends and axis labels

Covered Concepts

This session refers to :

- mathematical trigonometric functions
- vectors for data storage
- Python libraries for data displaying

Step by step

During this session, you can follow the steps below :

1. open a new Jupyter Notebook script (or basic python script)
2.

Ressources

This session is based on Python programming language.

You can use the **JupyterHub@Paris-Saclay** environment - https://jupyterhub.ijclab.in2p3.fr/ or **Spyder** development environment from *Anaconda*.

You will find tutorials on Python (and basics library as Numpy, Matplotlib or Scipy) at : http://lense.institutoptique.fr/python/, we suggest to you to read the following ones :

- How to create a function properly (with documented comments)
- How to create a vector (linear or logarithm evolution)
- How to plot data from vectors

Numerical Tools

Functions and libraries to use :

- **Numpy** for mathematical functions and vectors generation :
  - **linspace**
  - **logspace**
- **Matplotlib** for plotting data
  - **plotly** sublibrary
  - **figure**
  - **plot**
  - **legend**
  - **xlabel, ylabel**
  - **title**
  - **show**
- **Scipy** for scientific functions
  - **fftpack** sublibrary
  - **fft**

Advanced tools :

- **rcParams** from MatPlotLib.pyplot
- **Scipy** for scientific functions
  - **special** sublibrary
  - **jv** Bessel function