



PYTHON UE3

matplotlib, numpy

Vous devez rendre des programmes commentés et expliquer vos choix de conception : choix des structures de données, enchaînement des fonctions,....

1. Prise en main de matplotlib

Références : **numpy** : <http://docs.scipy.org>
http://wiki.scipy.org/Tentative_NumPy_Tutorial
matplotlib : <http://matplotlib.org/>

Depuis l'interpréteur python afficher la courbe représentative de la fonction :

$$y(x) = 4 \cdot e^{-0.5x} \cdot \sin(2x)$$

sur l'intervalle [-4;4] avec 1000 points.

On utilisera les modules *numpy* et *matplotlib.pyplot* et notamment : *arange* ou *linspace*, *plot*, *show* et *savefig*.

Utilisez les fonctions mathématiques de *numpy* de préférence aux fonctions de même nom de math car elles peuvent prendre directement des vecteurs en paramètres (array de numpy ou liste de python).

Ajoutez votre nom en titre et modifier les couleurs pour obtenir ceci :





PYTHON UE3

matplotlib, numpy

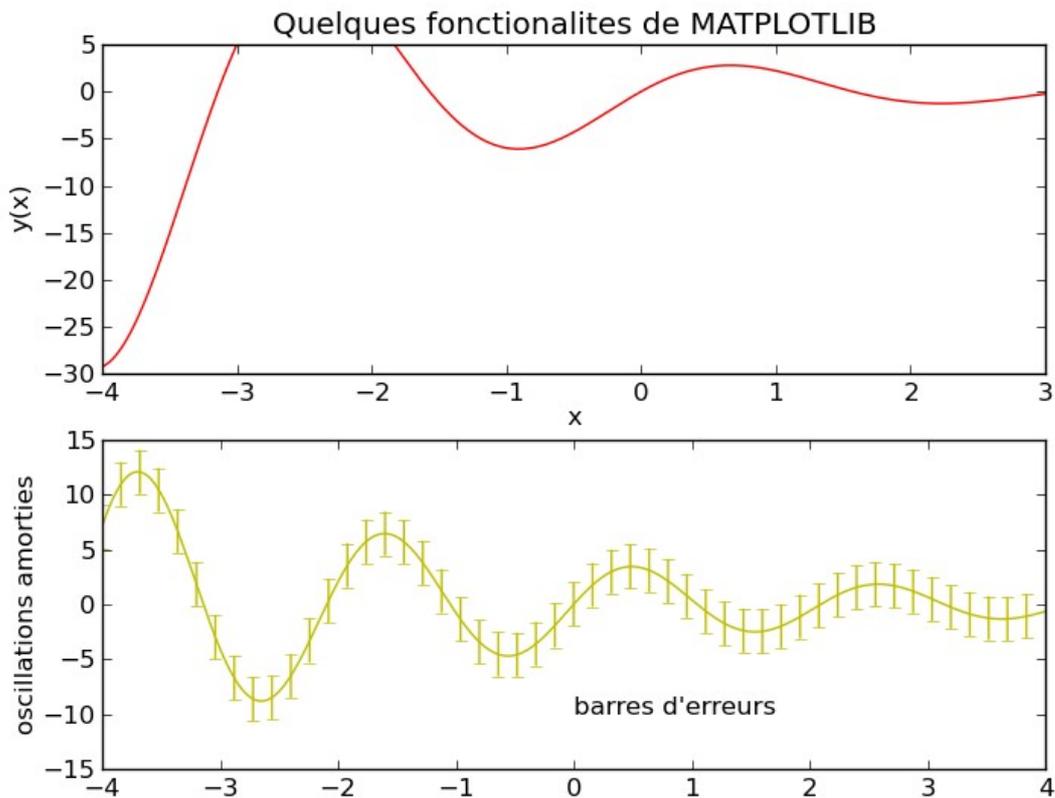
Sauver le graphique en png.

Donner la liste des instructions utilisées.

Ajoutez sur les mêmes axes la fonction en couleur verte :

$$z(x) = 4 \cdot e^{-0.3x} \cdot \sin(3x)$$

Créer un graphique montrant y et z comme ci dessous (dans un programme python) :





PYTHON UE3

matplotlib, numpy

2. Exploiter des données issues d'un fichier

De nombreux fichiers de mesures utilisent le format CSV (coma separated value). Les bibliothèques matplotlib, numpy et scipy proposent des fonctions évoluées pour lire ces fichiers. Cependant les fichiers sont rarement standards et il est important de savoir écrire ses propres fonctions de lectures pour pouvoir s'adapter à chaque cas particulier.

Les fichiers *DATALOGxxxxx.TXT* contiennent des données issues d'une station de mesures implantée sur l'île de Port Cros. L'organisation des fichiers est donnée dans le fichier *format_station_v1.txt*.

2.1 Extraction et séparation des données

A partir d'un fichier de mesures, créer 3 fichiers distincts. Le premier contenant les messages de service, le deuxième les données principales, le troisième les données de lumière.

Le nom des fichiers sera formé à partir du nom du fichier de base avec les extensions *service*, *data* et *lum* pour respectivement service, données principales et lumière.

Attention : ces fichiers ont été obtenus lors du test de la station qui à ce stade présentait un bug qui fait que la trame \$ est parfois tronquée. Dans ce cas la trame % qui suit se trouve sur la même ligne. Essayez de récupérer le plus de données possibles. Les données non récupérables des trames \$ tronquées seront remplacées par ---

2.2 Première exploitation

Affichez les températures moyenne, maximale et minimale pour un fichier.

Donnez le temps pour lequel on a les valeurs maximale et minimale.

Tracez l'évolution de la température en fonction du temps.

Tracez l'évolution de l'écart entre les deux capteurs de température en fonction du temps.

Tracez l'évolution de la température et de l'humidité en fonction du temps sur un même graphique : même axe des x et deux axes y distincts (voir twinx)

Tracez l'évolution de la tension batterie en fonction du temps. Attention aux données absentes.



PYTHON UE3

matplotlib, numpy

Tracez l'évolution des données de lumière du jour. Faire apparaître les saturations.

NB : On demande de "beaux graphiques" exploitant la richesse des fonctionnalités de matplotlib. En particulier doivent apparaître au minimum : le nom du fichier de mesure, les unités et une légende.

2.3 Exploitation de la date

La station de mesure (version 1) n'enregistre pas directement les données mais les transmet par une liaison série à une autre station (dite JASON

<http://sabiody.univ-tln.fr/jason.html>) qui possède une horloge temps réel et un disque dur. Un fichier issue de JASON est disponible :

JASON_PC_FORTEM_METEOTHYLUXV1_HSOL_20150412_151727.log

La station JASON insère la date et l'heure au moment de la réception de la trame comme deux nouveaux champs après le premier caractère (#,\$ ou%).

Écrire un programme qui transforme les fichiers DATALOG en fichiers JASON.

L'heure de départ est prise par défaut à partir de l'heure du PC ou à partir une heure de départ fournie par l'utilisateur. Ensuite l'heure est recalculée à partir du champ upTime.

2.4 Exploitation complète des données

Tracer toutes les données présentes dans un fichier JASON.

Votre programme produira des images png des différents graphiques demandés.

Le nom des images produites doit permettre de retrouver facilement le nom du fichier de départ et la nature de l'image.

-Températures (extTemp, humTemp,microTemp) et Humidité en fonction de la date avec un même axe des x.

- Tension batterie en fonction de la date.

- Tension batterie et luminosité diurne en fonction de la date avec un même axe des x.

- Luminosité nocturne en fonction de la date (éliminer ou faire apparaître clairement les plages saturées).

- Luminosité diurne en fonction de la date (éliminer ou faire apparaître clairement les plages nocturnes).