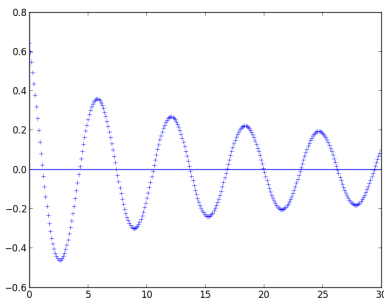


DEVOIR EN TEMPS LIBRE D'INFORMATIQUE N° 2

Récupération d'infos sur un circuit RLC

Le but de ce DM va être de récupérer un certain nombre d'informations concernant la décharge d'un condensateur dans un circuit RLC en régime libre. Une fois le DM effectué et validé, vous pourrez utiliser vos procédures dans un "vrai" TP de physique pour obtenir directement les valeurs à mesurer pour le TP. Le but du présent DM est d'automatiser les procédures de récupération du facteur de qualité et de la pulsation propre afin de pouvoir les utiliser en TP de physique. Ci-dessous une représentation des données possibles.

En abscisses : le temps t en secondes, et en ordonnées : la tension u en Volts



Temps ; EA1

```
0,0 ; 0,642970376624
0,100334448161 ; 0,595507088449
0,200668896321 ; 0,544739840012
0,301003344482 ; 0,491155291358
0,401337792642 ; 0,435251157651
0,501672240803 ; 0,377532941519
0,602006688963 ; 0,318510451838
```

1 Lecture du fichier

La première chose à faire est d'écrire une procédure de lecture du fichier. Celui-ci est du type vu au dessus et peut continuer ainsi sur plusieurs milliers de lignes. La procédure **lecture(fichier)** prend en argument une chaîne de caractères **fichier** qui donne l'adresse où se trouve le fichier sur l'ordinateur et renvoie deux listes: respectivement la liste des temps et la liste des tensions. Pour l'écrire, vous devrez

- faire attention à ignorer toute ligne commençant par **Temps;**
- remplacer les virgules (séparateur décimal en français) par des points (séparateur décimal anglo-saxon et donc de Python) en utilisant la méthode **replace** (voir **help(str.replace)**);
- séparer les colonnes selon le caractère ; en utilisant la méthode **split(';')** ;
- Convertir les chaînes de caractères correspondant au temps et à la tension associée en flottants (à l'aide de la fonction **float**) avant de les stocker dans les deux listes qui seront à renvoyer à la fin.

2 Traitement des données

2.1 Annulations de tension

Écrire une fonction **points_d_annulation(tensions)** qui prend en argument une liste de flottants et pour laquelle on veut trouver les positions des annulations successives de la fonction représentée par la liste. On définira les annulations comme les positions i telles que

- **tensions[i]** et **tensions[i+1]** soient de signes contraires ou **tensions[i]** est rigoureusement nulle;
- deux positions successives d'annulation soient séparées d'au moins 10 cases.

La fonction doit renvoyer la liste des positions successives d'annulation.

2.2 Maxima successifs

Écrire une fonction **maxima_successifs(tensions)** qui prend en argument une liste de flottants et qui renvoie une liste des positions i de ces maxima. On considère que le premier maximum ne peut avoir lieu *qu'après* la première annulation (puisque l'on ne sait pas exactement quand l'acquisition a été lancée). En outre,

on signale qu'on trouvera forcément un seul maximum entre la première et la troisième annulation, puis un autre entre la troisième et la cinquième, puis...

NB: si jamais plusieurs tensions venaient à avoir la même valeur maximale dans une même période, la position attendue sera la première pour laquelle le maximum est atteint.

3 Récupération des valeurs

Le but de cette section est de récupérer les paramètres ω_0 (pulsation propre) et Q (facteur de qualité) pour le circuit RLC dont la modélisation mathématiques peut être mise sous la forme :

$$u(t) = A \exp\left(-\frac{\omega_0 t}{2Q}\right) \cos(\Omega t + \phi) \quad \text{avec} \quad \Omega = \omega_0 \sqrt{1 - \frac{1}{4Q^2}}$$

3.1 Calcul de la pseudo-période, deux méthodes possibles

3.1.1 Méthode des zéros

La pseudo-période T vaut $2\pi/\Omega$. Elle est donnée par le double du temps séparant deux passages successifs par 0. écrivez donc une fonction **pseudo_periode_par_zeros(temps,tensions)** qui prend en argument une liste (triée) des temps et une liste des tensions associées à ces temps et renvoie une estimation (si possible précise) de la pseudo-période. NB: comme les dernières annulations peuvent être imprécises, on considèrera un maximum de 10 pseudo-périodes consécutives pour effectuer les calculs.

3.1.2 Méthode des maxima

La pseudo-période T est aussi (approximativement) égale au temps séparant deux maxima consécutifs. En regardant à nouveau au maximum les dix premières pseudo-périodes, écrire une fonction **pseudo_periode_par_maxima(temps,tensions)** qui prend en argument une liste (triée) des temps et une liste des tensions associées à ces temps et renvoie une estimation (si possible précise) de la pseudo-période.

On pourra s'amuser à regarder si les deux estimations diffèrent significativement l'une de l'autre.

3.2 Décrément logarithmique

Pour un régime libre (amorti vers 0), on appelle "décrément logarithmique" la quantité

$$\delta = \frac{1}{m} \ln\left(\frac{u_{\max}(n)}{u_{\max}(n+m)}\right)$$

où $u_{\max}(n)$ est la valeur du n^{e} maxima en tension. L'intérêt du décrément logarithmique est qu'on pourra aisément le relier au facteur de qualité.

Écrire une fonction **decrement_logarithmique(tensions)** qui prenne la liste des tensions en entrée et renvoie la valeur du décrément logarithmique δ comme défini plus haut et calculé entre le premier et le quatrième maximum. On signale que la fonction **log** du module **math** correspond justement au logarithme népérien noté "ln" de par chez nous (alors que le logarithme décimal noté "log" chez nous correspond à la fonction **log10** du module **math**).

3.3 Facteur de qualité et pulsation propre

On donne que le décrément logarithmique δ est lié au facteur de qualité Q via la relation $\delta = \frac{\pi}{\sqrt{Q^2 - 1/4}}$

et on rappelle que la pseudo-période T est liée à la pulsation propre ω_0 et au facteur de qualité Q via $\frac{2\pi}{T} = \Omega = \omega_0 \sqrt{1 - \frac{1}{4Q^2}}$, écrire une fonction **Q_et_w0(temps,tension)** qui prennent en entrée les listes de temps et des tensions et renvoie le facteur de qualité et la pulsation propre associés.