
[E] TP n°2 – Circuit RC

Dans ce TP, nous allons :

- déterminer expérimentalement la constante de temps d'un circuit RC ;
- résoudre l'équation différentielle du circuit RC numériquement.

I) Constante de temps du circuit RC

1) Montage électrique

- ⚙ Générer un signal crête à crête d'amplitude 5 V, de fréquence $f = 1$ kHz et avec un décalage (offset) de 5 V. Observer ce signal sur la voie CH1 de l'oscilloscope.
- ⚙ Réaliser un circuit RC série, alimenter par le signal précédent au GBF. Choisir $R = 2$ k Ω et $C = 10$ nF. Observer sur la voie CH2 de l'oscilloscope la tension aux bornes du condensateur.

2) Mesure de τ à l'oscilloscope

- ⚙ À l'aide de l'outil CURSEURS, mesurer le temps t_1 du début de la charge avec son incertitude-type associée $u(t_1)$ (incertitude de type B). Mesurer, de même, le temps t_2 auquel le condensateur est chargé de 63 % avec son incertitude-type associée $u(t_2)$.
- ⚙ En déduire la constante de temps $\tau = RC$ du circuit, ainsi que $u(\tau)$, à l'aide de la formule de composition des incertitudes.

II) Résolution numérique d'une ED d'ordre 1

On rappelle l'équation différentielle du circuit RC soumis à une tension d'alimentation E .

$$\frac{du}{dt} + \frac{u}{\tau} = \frac{E}{\tau}$$

On prendra comme condition initiale : $u(t = 0) = 0$.

On rappelle également les grandes étapes de la résolution d'une ED par la méthode d'Euler :

- écrire $du = u(t + dt) - u(t)$ puis isoler $u(t + dt)$;
- discrétiser la relation obtenue : remplacer $u(t) \mapsto u[n]$ et $u(t + dt) \mapsto u[n + 1]$;
- faire une boucle qui calcule récursivement les valeurs de $u[n]$ à l'aide de la relation précédente.

- 📄 Sur le site internet, télécharger le script PYTHON. Exécuter-le : il affiche la solution analytique de l'équation différentielle.
- 📄 Compléter le script PYTHON : résoudre par la méthode d'Euler l'équation différentielle du circuit RC.
- 📄 Exécuter le code et comparer la solution analytique avec la solution numérique. Faire varier la valeur de dt . Commenter.

AIDE POUR PYTHON

`import numpy as np` permet d'importer l'ensemble des fonctions du module numpy.

`u.append(a)` ajoute l'élément `a` à la fin de la liste `u`.

`u[-1]` renvoie le dernier élément de la liste `u`.