# Électrocinétique, activité expérimentale 7 : Méthode d'Euler pour le premier ordre

Pour les programmes python, on pourra utiliser Capytale, EduPython ou un compilateur en ligne. Pensez à régulièrement obtenir une sauvegarde du programme pour anticiper un éventuel crash de l'ordinateur.

Il est recommandé de poser les valeurs ou listes en début de programme.

## I. Méthode d'Euler pour le premier ordre

La méthode d'Euler est un algorithme permettant de résoudre, par approximation, des équations différentielles à partir de conditions initiales.

L'algorithme calcule pas à pas, au cours du temps, les valeurs prises par la grandeur étudiée, à partir des valeurs précédentes et de l'équation différentielle.

On cherche à exprimer $x(t+dt)$ en utilisant $x(t)$ et $\frac{dx}{dt}(t)$, ce dernier pouvant être déduit de $x(t)$ grâce à l'équation différentielle.

En physique-chimie, puisque la durée infinitésimale $dt$ est aussi petite que l'on souhaite, on a :

$$\frac{dx}{dt}(t) = \frac{x(t+dt) - x(t)}{dt}$$

Au passage, on rappelle que la variation infinitésimale $dx$ de la grandeur $x$ est alors telle que :

$$dx = x(t+dt) - x(t)$$

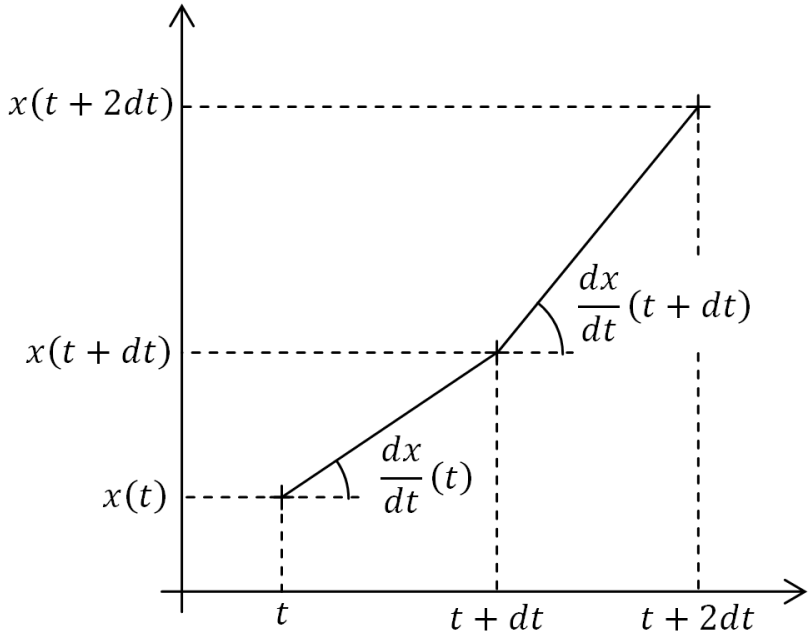
On a :

$$x(t+dt) - x(t) = \frac{dx}{dt}(t)\,.\,dt$$

$$\boxed{x(t+dt) = x(t) + \frac{dx}{dt}(t)\,.\,dt}$$

Ainsi, l'algorithme a le fonctionnement suivant : on choisit une valeur $dt$ assez faible pour que l'approximation soit correcte, et on part d'une condition initiale. Ensuite, on calcule les valeurs suivantes en calculant la grandeur $x(t) + \frac{dx}{dt}(t).\,dt$ .

Pour obtenir $\frac{dx}{dt}(t)$ , on utilise l'équation différentielle et $x(t)$.




Si on pose $i$ la variable qui permet de repérer le temps dans l'algorithme, dt le pas de temps, et deriv la fonction qui donne la valeur de la dérivée de x en fonction des autres paramètres, on a alors, pour $i \in [\![0, n-1]\!]$ avec n le nombre de points :

x[i+1]=x[i]+deriv(x[i], t[i])*dt

Dans le cas d'une équation d'ordre 2, il faudra également une condition initiale pour $\frac{dx}{dt}$, et appliquer la formule $\frac{dx}{dt}(t+dt) = \frac{dx}{dt}(t) + \frac{d^2x}{dt^2}(t)\,.\,dt$

Le pas de temps $dt$ ne doit cependant pas être trop petit pour que la simulation ne soit pas trop longue et prenne trop de place dans la mémoire vive. Par exemple on pourra choisir que $dt$ est un millième de la durée typique du signal.

## II. Algorithme pour le circuit RC

On souhaite écrire un algorithme de méthode d'Euler appliqué au circuit RC, afin de pouvoir vérifier que le programme fonctionne en accort avec la théorie. Pour les valeurs on prendra des valeurs typiques d'expériences.

Dans le cas du circuit RC, puisque l'on a :

$$\frac{du_C}{dt} + \frac{1}{RC}u_C = \frac{e(t)}{RC}$$

On a alors :

$$\boxed{\frac{du_C}{dt} = -\frac{1}{RC}u_C + \frac{e(t)}{RC}}_H$$

On utilise la condition initiale $u_C(t=0) = 0$ et $e(t)$ est un échelon de tension tel que $e(t) = E$ pour $t \geqslant 0$.

1. Écrire une fonction derivu qui retourne la dérivée par rapport au temps de $u_C$ à partir des valeurs de $u_C$ et de e.
2. Écrire une boucle pour calculer n points à l'aide de la méthode d'euler.
3. Tracer $u_C(t)$ et $e(t)$ avec une légende et des axes. Si les courbes ne sont pas cohérentes avec la théorie, corriger les erreurs. Choisir une durée d'observation pour que les variations soient visibles, et reproduire le résultat sur copie.

On choisit à présent la fonction $e(t) = \left|\sin\left(\frac{\pi}{RC}t\right)\right|$.

Un filtre RC est un filtre passe-bas de pulsation propre $\frac{1}{RC}$.

4. Tracer $u_C(t)$ et $e(t)$ dans ce cas. Reproduire la courbe et commenter.
5. Tracer $u_C(t)$ et $e(t)$ pour une fonction en dents de scie de 0 à E de pulsation $\frac{4}{RC}$. Reproduire la courbe et commenter.
6. Tracer $u_C(t)$ et $e(t)$ pour une fonction carrée de 0 à E de pulsation $\frac{4}{RC}$. Reproduire la courbe et commenter.

## III. Algorithme généraliste

7. Écrire une fonction euler1 qui prend comme arguments une fonction derivf qui retourne la dérivée temporelle utile, une liste t de valeurs de temps n'étant pas forcément réguliers, et une valeur f0. Vérifier que cette fonction donne toujours le résultat de la question précédente.
8. Écrire une fonction euler2 qui prend comme arguments une fonction derivf qui retourne la dérivée temporelle utile, une fonction deriv2f qui retourne la dérivée temporelle seconde utile, une liste t de valeurs de temps n'étant pas forcément réguliers, et une valeur f0.
9. Vérifier la fonction euler2 avec l'exemple du circuit LC idéal tel que $u_C(0) = E$, $\frac{du_C}{dt}(0) = 0$, et :

$$\frac{d^2u_C}{dt^2} + {\omega_0}^2 u_C = 0$$

## Annexe A : Fonctions python

len(liste) #nombre d'éléments dans la liste
liste.append(x) #ajoute la valeur x à la fin de la liste
for i in range(n): #boucle faisant passer l'entier i de 0 à n−1
a%b #reste de la division euclidienne de a par b

import numpy as np #importe numpy avec l'abréviation np
np.pi #valeur de pi
np.sin(x) #valeur de sin(x) avec x en radians
np.zeros(n) #liste avec n zéros
np.linspace(f0, f1, n) #liste de n valeurs espacées d'une même différence entre f0 et f1 inclus
np.array([fonc(i) for i in liste]) #liste avec les valeurs renvoyées par fonc() avec les valeurs dans la liste

import matplotlib.pyplot as plt #importe pyplot avec l'abréviation plt
plt.figure() #initialise un graphique
plt.plot(liste_x, liste_y, color='couleur', label='legende', linewidth=2) #ajoute le graphe y(x) en largeur de trait de 2 ; pour une couleur grise on écrira '0.6' et pour du noir on écrira 'k'
plt.xlabel('legende abs (unite)') #légende pour l'axe des abscisses
plt.ylabel('legende ord (unite)') #légende pour l'axe des ordonnées
plt.legend(loc=1) #ajoute une légende des courbes en haut à droite du graphe
plt.grid(linewidth=0.2) #ajoute un quadrillage d'épaisseur 0.2
plt.show() #affiche le graphe