



## PYTHON – Travaux Dirigés N°1

# - ETUDE DE LA METHODE D'EULER EXPLICITE -

Physique Chimie CPGE PT T.Roque

lien vers le blog <http://cpge-reunion-ptsiblogspot.com> (<http://cpge-reunion-ptsiblogspot.com>)

Entrée [ ]:

Dans ce notebook, on rappelle le principe de la méthode d'Euler explicite pour intégrer une équation différentielle dont on se limite pour le moment au premier ordre. On verra plus tard, avec la méthode d'Euler implicite comment transformer une équation différentielle d'ordre 2 en 2 équations différentielles d'ordre 1.

## PREMIERE PARTIE, BASES THEORIQUES

La méthode d'Euler est la base de toute méthode d'intégration numérique d'équation différentielle. C'est une méthode d'intégration numérique rudimentaire et simple à mettre en œuvre. Il est donc important de la maîtriser avant d'explorer des situations plus complexes.

### Rappel de la méthode

La méthode d'Euler explicite sert à intégrer une équation différentielle du premier ordre du type :

$$\frac{dy}{dt} = f(y(t), t)$$

avec une condition de type

$$y(0) = y_0$$

où  $f$  est une fonction qui dépend à la fois de la fonction inconnue  $y(t)$  et de la variable temporelle  $t$ . La procédure consiste à interpréter la dérivée  $\frac{dy}{dt}$  comme le rapport de la variation infinitésimale  $dy$  de la grandeur  $y$  pendant la durée infinitésimale  $dt$ . L'équation différentielle se réécrit alors, si on multiplie par  $dt$  de part et d'autre,

$$dy \approx f(y(t), t) dt$$

et la nouvelle valeur de  $y$  en  $t + dt$  correspond à peu près à celle en  $t$  à laquelle on ajoute  $dy$ :

$$y(t + dt) = y(t) + dy \approx y(t) + f(y(t), t) dt$$

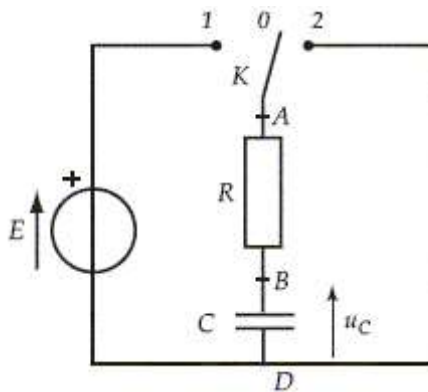
Pour avancer dans le temps, il suffira d'y aller par "pas" temporel de durée  $dt$  et de calculer l'évolution de  $y$  de proche en proche en utilisant à chaque fois la dernière valeur connue de  $y$  au "rang précédent  $n$ " en  $t$ , ou encore " $y_n$  en  $t_n$ " pour calculer la suivante, au "rang suivant", en  $t + dt$ , ou encore " $y_{n+1}$  en  $t_{n+1}$ ". On pourra alors percevoir l'équation approchée précédente comme une équation de récurrence :

$$y_{n+1} = y(t_n) + f(y(t_n), t) \cdot dt = y_n + f(y_n, t) \cdot dt$$

**Remarque : on utilise souvent la variable  $t$  pour étudier l'évolution de fonctions comme  $y$  au cours du temps, mais rien n'empêche d'avoir une autre variable comme l'altitude  $z$  dans l'équation de l'hydrostatique dans le cas d'une atmosphère avec évolution affine de la température et dans le modèle du gaz parfait. On aurait alors à appliquer Euler à :**

$$\frac{dP}{dz} = f(P, z) = -\frac{P(z) \cdot M}{R \cdot T_0 \cdot (1 - a \cdot z)}$$

### EXEMPLE DE LA DECHARGE D'UN CONDENSATEUR



Supposons que  $E=10\text{V}$  et que l'interrupteur  $K$  est en position 1 depuis longtemps. A  $t=0$ , on le bascule en position 2, démontrer que  $\frac{dy}{dt} = -\frac{1}{\tau} y$  avec  $y(0) = 10$  et  $\tau = R \cdot C$ . Résoudre l'équation de façon exacte.

Si on choisit de trouver une solution numérique approchée avec la méthode d'Euler, on remarque que ici, on a simplement  $f(y, t) = -\frac{1}{\tau} y$ . Il nous reste à :

- choisir  $dt$  suffisamment petit pour que les erreurs successives ne soient pas trop grandes
- stocker les valeurs des temps successifs et des valeurs correspondantes calculées de  $y$  avec l'équation de récurrence
- faire la représentation graphique et la comparer à la solution analytique exacte de l'équation différentielle

Entrée [5]:

```

import numpy as np                # Pour avoir des np.array
import matplotlib.pyplot as plt  # Pour faire les courbes à la fin

dt = 1e-1
t0 = 0
y0 = 10
tau = 2
tmax = 5 * tau # On va intégrer jusqu'à 5\tau
N = int((tmax-t0) / dt) # Pour savoir le nombre de places à prévoir

# On prépare les contenants qui stockeront les valeurs calculées
# Le "+1" correspond à la valeur initiale
T = np.zeros(N+1)
Y = np.zeros(N+1)

# On impose les valeurs initiales
T[0] = t0
Y[0] = y0

def f(y, t):
    """
    La fonction en question ne dépend que de y,
    mais ce sera plus facile à généraliser quand on aura
    un second membre non constant dans les prochains épisodes.
    En outre, informatiquement, il n'est pas abhèrent de demander
    une variable (ici t) et finalement ne pas l'utiliser si cela
    simplifie une généralisation future.
    """
    return -1 / tau * y

# La boucle de calcul proprement dite dans laquelle chaque Y est calculé
# à partir de celui d'avant.
for i in range(1, N+1):
    # On utilise les valeurs précédentes pour calculer l'incrément
    dy = f(Y[i-1], T[i-1]) * dt
    # et on le rajoute à la valeur précédente pour avoir la nouvelle
    Y[i] = Y[i-1] + dy
    # Le temps s'incrémente aussi d'un pas
    T[i] = T[i-1] + dt

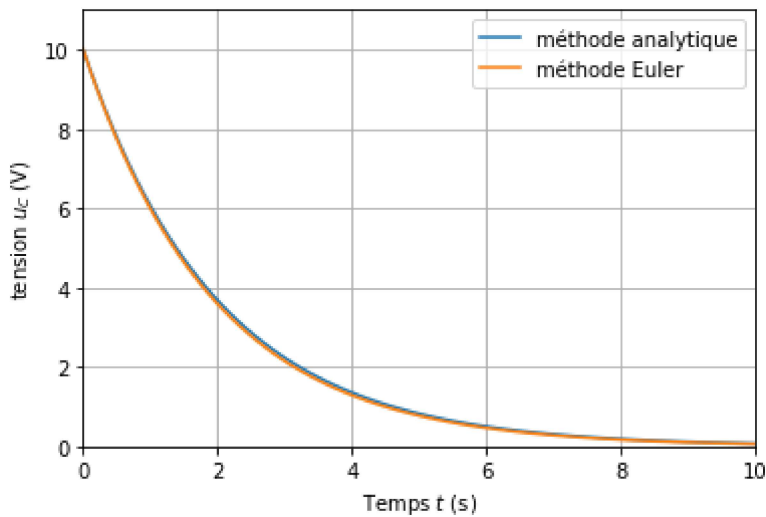
# On peut à présent représenter notre solution ainsi que la solution
# connue du cours en  $y_0 * \exp(-t/\tau)$ 

theorique = y0 * np.exp(-T / tau)
plt.plot(T, theorique, label = 'méthode analytique')
plt.plot(T, Y, label = 'méthode Euler')
plt.xlabel('Temps $t$ (s)')
plt.ylabel('tension $u_C$ (V)')
plt.xlim(t0, tmax)
plt.ylim(0, 1.1*y0)
plt.grid()
plt.legend()

```

Out[5]:

&lt;matplotlib.legend.Legend at 0x292e8668bc8&gt;



On voit qu'avec ce pas de temps, les deux versions sont très proches, même si on voit une différence entre 1 et 5 secondes qui montre que l'intégration par Euler a tendance à surestimer la baisse de  $y$ , mais dans l'ensemble, c'est acceptable d'autant qu'on tend vers une constante (ici nulle).

### A VOUS DE CODER : CHARGE DU CONDENSATEUR

Supposons le condensateur déchargé et basculons l'interrupteur en position 1. Démontrer que :

$$\frac{dy}{dt} + \frac{1}{\tau} y = \frac{E}{\tau}$$

avec la condition initiale  $y(0) = 0$ .

On impose les valeurs de  $dt$ ,  $t_{\max}$ ,  $\tau$  et  $E$ . À vous d'adapter de terminer le code Vous mettrez dans la variable `reponse` (SANS ACCENT) à la fois la valeur de  $y$  en  $t = t_0 + \tau$  et  $t = t_0 + 5\tau$ .

Entrée [13]:

```

import numpy as np                # Pour avoir des np.array
import matplotlib.pyplot as plt  # Pour faire les courbes à la fin
E=10
tau=5
tmax=25
dt=0.01
t0 = 0
y0 = 0
N = int((tmax-t0) / dt) # Pour savoir Le nombre de places à prévoir

# On prépare les contenants qui stockeront les valeurs calculées
# Le "+1" correspond à la valeur initiale
T = np.zeros(N+1)
Y = np.zeros(N+1)

# On impose les valeurs initiales
T[0] = t0
Y[0] = y0

def f(y, t):
    """
    La fonction en question ne dépend que de y,
    mais ce sera plus facile à généraliser quand on aura
    un second membre non constant dans les prochains épisodes.
    En outre, informatiquement, il n'est pas abhèrent de demander
    une variable (ici t) et finalement ne pas l'utiliser si cela
    simplifie une généralisation future.
    """
    return (E - y) / tau

# La boucle de calcul proprement dite dans laquelle chaque Y est calculé
# à partir de celui d'avant.
for i in range(1, N+1):
    # On utilise les valeurs précédentes pour calculer l'incrément
    dy = f(Y[i-1], T[i-1]) * dt
    # et on le rajoute à la valeur précédente pour avoir la nouvelle
    Y[i] = Y[i-1] + dy
    # Le temps s'incrémente aussi d'un pas
    T[i] = T[i-1] + dt

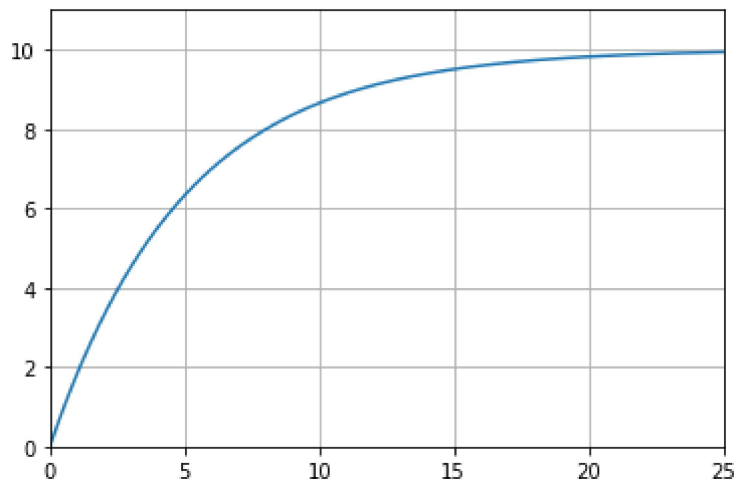
plt.plot(T, Y)
plt.ylim(0, 1.1*E)
plt.xlim(t0, tmax)
plt.grid()

#valeurs à tau et 5tau
reponse = Y[int(tau/dt)], Y[5*int(tau/dt)]

print(reponse)

```

(6.32488745142841, 9.932957034143806)



- Superposez sur la même courbe différentes charges avec des  $\tau$  différents
- Réalisez une charge à 10Volts alors que le condensateur est encore chargé, au temps initial, à 1 Volt
- Recommencez l'exercice en utilisant des listes plutôt que des tableaux

Entrée [ ]: