Méthode d'Euler 1

1 Introduction

La méthode d'Euler est une méthode numérique de résolution d'équations différentielles pas à pas. Elle permet d'obtenir une fonction $y(t)$ solution d'une équation différentielle d'ordre 1. Connaissant la valeur $y(t)$ de la fonction à un instant t cette méthode évalue la valeur $y(t+h)$ de la fonction un pas h plus loin (on utilisera plutôt le nombre de points n en physique qui est lié au pas) en utilisant son taux de variation. On rappelle que la dérivée d'une fonction à l'aide du taux de variation $y(t)$ est :

$$\frac{\mathrm{d}y}{\mathrm{d}t} = \lim_{h \to 0} \frac{y(t+h) - y(t)}{h}$$

La méthode est d'autant plus fidèle que le pas h est petit, si on choisit h très petit alors on a :

$$\frac{\mathrm{d}y}{\mathrm{d}t} = \frac{y(t+h) - y(t)}{h}$$

1. Exprimer $y(t+h)$ à l'aide de l'expression ci-dessus.

2. Compléter les `None` dans le bout de code suivant afin que la fonction `euler` renvoie la liste des instants t ainsi que la liste des valeurs $y(t)$ permettant de tracer la fonction.

```
1  def euler(dy, y0, t0, tf, n): # n nombre de points
2          h = None # h le pas
3          liste_t = []
4          liste_y = []
5          y = None # initialisation
6          t = None # initialisation
7          for k in range(n):
8                  liste_t.append(t)
9                  liste_y.append(y)
10                 t = t + h
11                 y = y + h*dy(y,t)
12         return None, None # renvoi des listes
```

Que représente `dy(y,t)` en ligne `11` ?

3. Rappeler la forme de l'équation différentielle d'ordre 1 que l'on retrouve lors de la charge d'un condensateur. On introduira les notations τ et E. Isoler la dérivée $\frac{\mathrm{d}y}{\mathrm{d}t}$.

4. Écrire une fonction `ED_charge(y,t)` qui retourne la dérivée $\frac{\mathrm{d}y}{\mathrm{d}t}$ à l'aide de la question précédente. On posera les paramètres R et C dans la fonction avant de définir τ.

5. Exprimer $y(t+h)$ en fonction de $y(t)$, τ, E et h.

6. Importer `matplotlib.pyplot` avec l'alias `plt`.

7. Compléter le bout de code afin de tracer l'évolution de la charge d'un condensateur initialement déchargé pour $R = 1000$ kΩ, $C = 100$ μF, $E = 5$ V. Durée totale : 1000 s avec 200 valeurs.

```
    print('Tau = {} s'.format(round(R*C),1)) #
    n = None # nombre de points
    essai_t, essai_y = euler(None, None, None, None, None)
    plt.clf() # efface la figure courante
    plt.plot(essai_t, essai_y) #
    plt.title("Charge d'un condensateur") #
    plt.xlabel("Temps (s)") #
    plt.ylabel("uC (V)") #
    plt.savefig('essai') ; plt.show() #
```

8. Compléter les commentaires ci-dessus. Introduire ces instructions dans une fonction `essai()`.

2 Tracé des valeurs expérimentales

9. Reprendre le fichier LatisPro de la charge du condensateur et l'exporter au format `.txt`.

10. La fonction `lecture(nom)` permet de récupérer dans des listes les données issues du fichier texte.

```
def lecture(nom):
    """ lecture d'un fichier texte pour recuperer les donnees en listes """
    with open(nom, "r") as fichier :
        lignes = fichier.readlines() # on recupere les lignes comme chaines
    liste_t, liste_uC = [], [] # listes de stockage
    for i in range(len(lignes)): # pour toutes les lignes
        lignes[i] = lignes[i].replace(',', '.') # remplace , par . si besoin
        lignes[i] = lignes[i].split() # on separe la tabulation
        liste_t.append(float(lignes[i][0])) #
        liste_uC.append(float(lignes[i][1])) #
    return liste_t, liste_uC
```

Compléter les commentaires vides ci-dessus. Que représente l'argument `nom` de cette fonction ?

11. Modifier le bout de code suivant afin de lire votre fichier. Afficher ensuite le contenu de `donnees`.

```
donnees = lecture('RC_120.txt') #
```

12. Écrire une fonction `experience()` sans arguments qui exécute les instructions suivantes :

```
    experience_t, experience_uC = donnees[0], donnees[1]
    plt.clf () # efface la figure courante
    plt.plot(experience_t , experience_uC, "+") #
    plt.title ("Mesures de la charge d'un condensateur") #
    plt.xlabel("Temps (s)") #
    plt.ylabel("uC (V)") #
    plt.savefig('Mesures') #
    plt.show() #
```

13. Compléter tous les commentaires vides du code ci-dessus.

14. Repérer et noter la valeur initiale $u_C(0)$ ainsi que la valeur finale $u_C(\infty) = E$.

15. Exécuter votre code puis appeler la fonction `experience()` dans le shell.

3 Modèle et confrontation

16. Introduire les paramètres `E`, `R`, `C` et `uC0` comme variables globales.

17. Écrire une fonction `modele()` en adaptant la fonction `essai()` afin de tracer le modèle correspondant à votre expérience. On prendra $n = 100$ points.

18. Exécuter votre code puis appeler la fonction `modele()` dans le shell.

19. Écrire une fonction `confrontation()` qui permet de tracer sur le même graphe les points de l'expérience ainsi que le modèle obtenu par résolution de l'équation différentielle par la méthode d'Euler.

20. Exécuter votre code puis appeler la fonction `confrontation()` dans le shell. Conclure.

Bon courage et bon travail ! ☺