Dans ce TP, on présente la méthode de Euler qui permet de donner une solution approximative des équations différentielles.

1 Problèmes

On considère un système (par exemple un circuit électronique) recevant $e(t)$ en entrée et produisant $s(t)$ en sortie.

$$e(t) \longrightarrow \boxed{\text{Système}} \longrightarrow s(t)$$

Le système obéit à une certaine équation différentielle.

$$(\mathcal{E}) : \quad \tau \cdot s' + s = e$$

On sait également que $s(0) = 0$.

2 Principe

On sait $s' = \frac{\mathrm{d}s}{\mathrm{d}t}$ avec $\mathrm{d}t$. Si $\mathrm{d}t$ est petit – au lieu d'être infinitésimal – cette égalité devient une approximation. On aura donc la variation $\mathrm{d}s \simeq s'(t) \cdot \mathrm{d}t$.

On se fixera donc un $\mathrm{d}t$ assez petit pour que l'approximation soit correcte.

— On connaît $s(t_0)$ et $e(t_0)$ à un instant t_0,
— en utilisant $\mathcal{E}$ on en déduit $s'(t_0) = \frac{1}{\tau}(e(t_0) - s(t_0))$,
— on calcule $s(t_0 + \mathrm{d}t) = s(t_0) + s'(t_0) \cdot \mathrm{d}t$

Ainsi, de proche en proche, on peut calculer tous les $s(t)$ avec $t \in [0; T]$.

En général, au lieu de fixer $\mathrm{d}t$ on préfère :

— fixer la durée T que l'on souhaite simuler,
— fixer un nombre N assez grand et poser $\mathrm{d}t = \frac{T}{N}$. Ainsi N représente le nombre de points que l'on obtiendra sur la courbe.

3 Tableur

(1) Ouvrez le tableur et recopier ce tableau

	A	B	C	D	E	F
1	tau	2	T	10	N	1000
2						
3	n	t0	e(t0)	s(t0)	s'(t0)	
4	0			0		
5	1					

Il faut étendre la colonne A pour que n aille jusque 1000

(2) Écrivez en B4 la formule permettant de calculer le temps t_0 en fonction de n et $\mathrm{d}t = \frac{T}{N}$.

Étendez la formule jusqu'en bas.

(3) Pour e on prendra $e(t) = \begin{cases} 1 \text{ si } 0 \leqslant t \leqslant 2.5 \\ 0 \text{ sinon} \end{cases}$ Complétez la colonne C.

(4) En E4, écrivez la formule permettant de calculer $s'(t_0)$ en fonction de $e(t_0)$ et $s(t_0)$
Étendez jusqu'en bas même si pour l'instant il manque les valeurs de $s(t_0)$.

(5) En D5, écrivez la formule permettant de calculer $s(t_0)$ en fonction du $s(t_0)$ précédent et du $s'(t_0)$ précédent.
Étendez jusqu'en bas.

(6) Tracez les courbes de e et s avec t en abscisse. Vous pourrez ensuite modifiez la valeur de τ pour voir l'effet.

4 Code Python

Les numéros à gauche sont les numéros de ligne. Ne pas les recopier.

(1) On commence par importer le module graphique

```python
import matplotlib.pyplot as plt # graphique
```

(2) On définit T et N ce qui permet de calculer $\mathrm{d}t$.

```python
T = 10
N = 1000
dt = T/N
```

(3) On fournit les paramètres de l'équation : il faut indiquer la valeur de τ et l'expression de $e(t)$. On reprend le même que précédemment.

```python
tau = 2
def e(t):
    if 0 <= t <= 2.5:
        return 1
    else:
        return 0
```

(4) On va utiliser des **tableaux**. Ces tableaux sont des listes de nombres que l'on va compléter au fur et à mesure. Les tableaux sont initilisés avec `[]`

```python
liste_t = [0]    # liste des temps
liste_e = [e(0)] # liste des valeurs de e
liste_s = [0]    # liste des valeurs de s
```

(5) Pour ajouter une valeur à une liste, on utilise `append`. Quand on veut lire le dernier élément d'une liste, on utilise `[-1]`. Pour parcourir tous les instants de la simulation, on utilise une boucle `for`.

```python
for i in range(N): # répète N fois
    s_deriv = 1/tau*(liste_e[-1] - liste_s[-1])
    nouveau_s = ...
    nouveau_t = ...
    nouveau_e = ...
    liste_s.append(nouveau_s)
    liste_t.append(nouveau_t)
    liste_e.append(nouveau_e)
```

Complétez les ...

(6) Il ne reste qu'à représenter les courbes.

```python
plt.plot(liste_t, liste_e, 'r', label='entrée')
plt.plot(liste_t, liste_s, 'b', label='sortie')
plt.legend()
plt.show()
```

Exécutez pour visualiser les courbes.

Vous pouvez répéter l'opération en modifiant des paramètres comme τ.