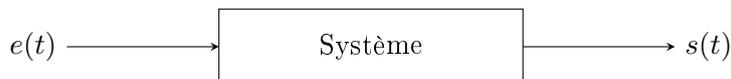


Dans ce TP, on présente la méthode de Euler qui permet de donner une solution approximative des équations différentielles.

## 1 Problèmes

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



Le système obéit à une certaine équation différentielle.

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

On sait également que  $s(0) = 0$ .

## 2 Principe

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

On se fixera donc un  $dt$  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  $s(t_0 + dt) = s(t_0) + s'(t_0) \cdot dt$

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

En général, au lieu de fixer  $dt$  on préfère :

— fixer la durée  $T$  que l'on souhaite simuler,

— fixer un nombre  $N$  assez grand et poser  $dt = \frac{T}{N}$ . 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  $dt = \frac{T}{N}$ .

Étendez la formule jusqu'en bas.

(3) Pour  $e$  on prendra  $e(t) = \begin{cases} 1 & \text{si } 0 \leq t \leq 2.5 \\ 0 & \text{sinon} \end{cases}$  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 modifier la valeur de  $\tau$  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

```
Py 1 import matplotlib.pyplot as plt # graphique
```

- (2) On définit  $T$  et  $N$  ce qui permet de calculer  $dt$ .

```
Py 2 T = 10
3 N = 1000
4 dt = T/N
```

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

```
Py 5 tau = 2
6 def e(t):
7     if 0 <= t <= 2.5:
8         return 1
9     else:
10        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 initialisés avec []

```
Py 11 liste_t = [0] # liste des temps
12 liste_e = [e(0)] # liste des valeurs de e
13 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`.

```
Py 14 for i in range(N): # répète N fois
15     s_deriv = 1/tau*(liste_e[-1] - liste_s[-1])
16     nouveau_s = ...
17     nouveau_t = ...
18     nouveau_e = ...
19     liste_s.append(nouveau_s)
20     liste_t.append(nouveau_t)
21     liste_e.append(nouveau_e)
```

Complétez les ...

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

```
Py 22 plt.plot(liste_t, liste_e, 'r', label='entrée')
23 plt.plot(liste_t, liste_s, 'b', label='sortie')
24 plt.legend()
25 plt.show()
```

Exécutez pour visualiser les courbes.

Vous pouvez répéter l'opération en modifiant des paramètres comme  $\tau$ .