

# Informatique

## **Intégration numérique**

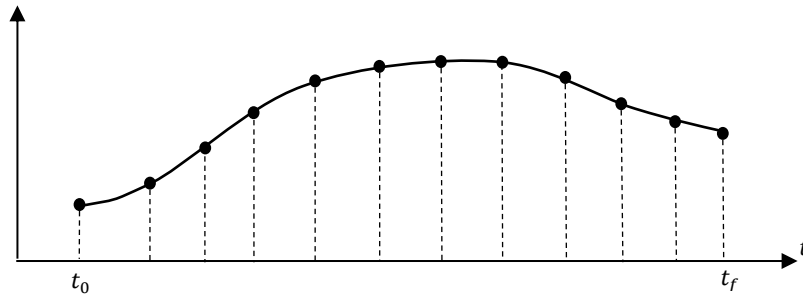
*Cours*

<b>Intégration numérique.....</b>	<b>3</b>
<b>1.I. Contexte.....</b>	<b>3</b>
<b>1.II. Valeur à gauche – Méthode des rectangles .....</b>	<b>4</b>
<b>1.III. Valeur à droite - Méthode des rectangles .....</b>	<b>4</b>
<b>1.IV. Valeur centrée – Méthode des trapèzes.....</b>	<b>5</b>
1.IV.1 Principe .....	5
1.IV.2 Remarques .....	6
<b>1.V. Remarque sur les trois méthodes.....</b>	<b>6</b>
<b>1.VI. Fonction python préprogrammée .....</b>	<b>6</b>

# Intégration numérique

## 1.I. Contexte

Soit le signal échantillonné  $e$  suivant contenant  $n$  valeurs à partir du temps  $t_0$  :



On veut :

$$\int_{t_0}^{t_f} e(t) dt$$

Appelons  $t_i$  et  $t_{i+1}$  les temps de part et d'autre de chaque intervalle.

On va sommer les aires rectangles sur chaque intervalle  $[t_i, t_{i+1}]$ , de largeur  $T_i = t_{i+1} - t_i$  et de hauteur, soit :

- Valeur à gauche :

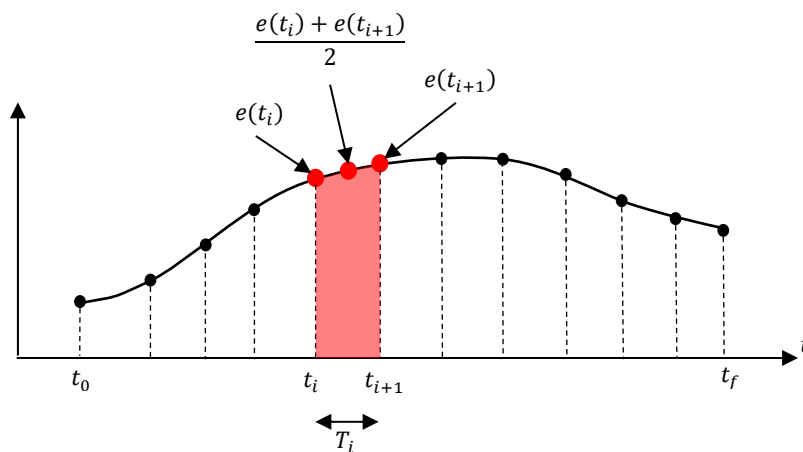
$$e(t_i)$$

- Valeur à droite :

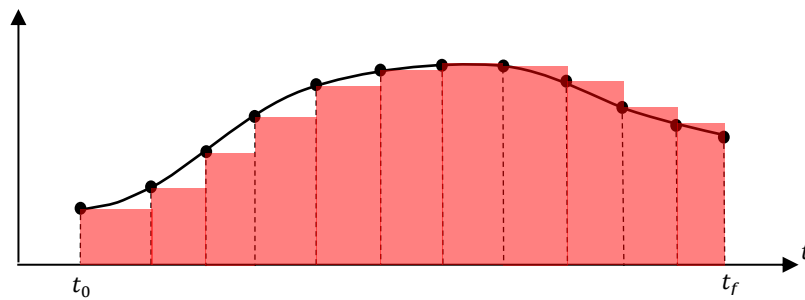
$$e(t_{i+1})$$

- Valeur centrée :

$$\frac{e(t_i) + e(t_{i+1})}{2}$$



## 1.II. Valeur à gauche - Méthode des rectangles

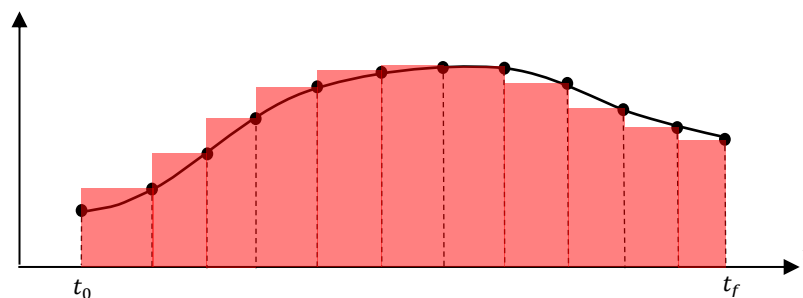


$$\int_{t_0}^{t_f} e(t) dt \approx \mathcal{A} = \sum_{i=0}^{n-1} T_i e(t_i) = \sum_{i=0}^{n-1} (t_{i+1} - t_i) e(t_i)$$

Attention à ne bien prendre que  $n - 1$  valeurs !

On voit qu'il y a sous-estimation de la courbe lorsqu'elle est croissante et surestimation lorsqu'elle est décroissante.

## 1.III. Valeur à droite - Méthode des rectangles



$$\int_{t_0}^{t_f} e(t) dt \approx \mathcal{A} = \sum_{i=0}^{n-1} T_i e(t_{i+1}) = \sum_{i=0}^{n-1} (t_{i+1} - t_i) e(t_{i+1})$$

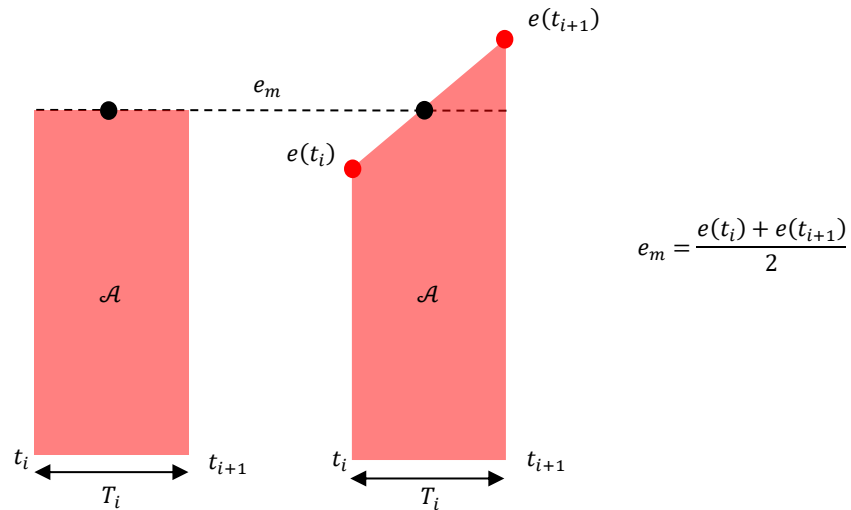
Attention à ne bien prendre que  $n - 1$  valeurs !

On voit qu'il y a surestimation de la courbe lorsqu'elle est croissante et sous-estimation lorsqu'elle est décroissante.

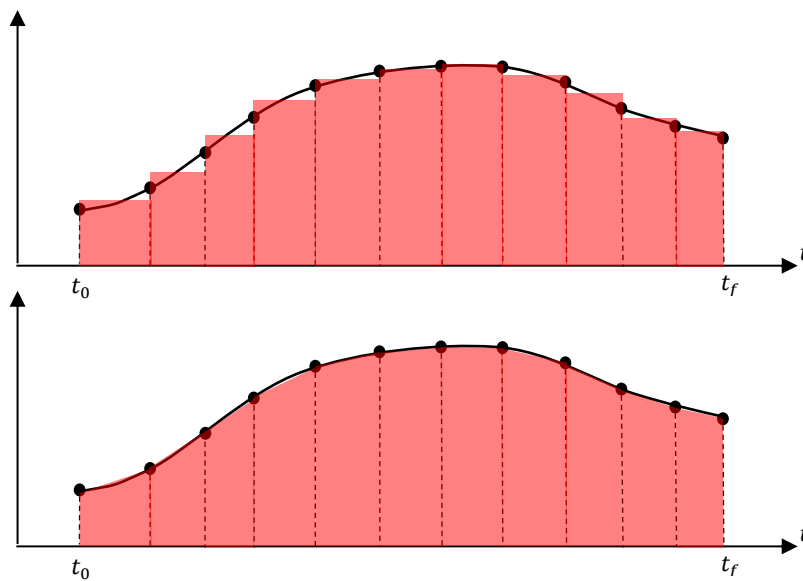
## 1.IV. Valeur centrée – Méthode des trapèzes

### 1.IV.1 Principe

Le calcul d'aires en tenant compte de la valeur centrée sur le segment de largeur  $T_i$  revient calculer les aires de trapèzes, d'où le nom de méthode des trapèzes :



Les deux surfaces ci-dessus ont des aires égales.



$$\int_{t_0}^{t_f} e(t) dt \approx \mathcal{A} = \sum_{i=0}^{n-1} T_i \frac{e(t_i) + e(t_{i+1})}{2} = \sum_{i=0}^{n-1} (t_{i+1} - t_i) \frac{e(t_i) + e(t_{i+1})}{2}$$

## 1.IV.2 Remarques

Attention à ne bien prendre que  $n - 1$  valeurs !

On voit que cette méthode compense à peu près la surestimation et la sous-estimation de  $e$  sur chacun des intervalles de largeur  $T_i$ . Elle sera donc privilégiée. Elle sous estimera ou surestimera les intégrales en fonction de la courbure (concavité/convexité) de la courbe intégrée.

On privilégiera le calcul avec les valeurs centrées plutôt que de faire la somme des aires des rectangles et des triangles supérieurs...

Lorsque l'on a bien des intervalles réguliers de largeur  $T_i = T \forall i$ , on a aussi :

$$\mathcal{A} = \sum_{i=0}^{n-1} T_i \frac{e(t_i) + e(t_{i+1})}{2} = \frac{T}{2} \sum_{i=0}^{n-1} (e(t_i) + e(t_{i+1}))$$

$$\text{or } \sum_{i=0}^{n-1} (e(t_i) + e(t_{i+1})) = e(t_0) + e(t_1) + e(t_1) + e(t_2) + \dots + e(t_{n-1}) + e(t_n)$$

$$\sum_{i=0}^{n-1} (e(t_i) + e(t_{i+1})) = e(t_0) + 2e(t_1) + 2e(t_2) + \dots + 2e(t_{n-1}) + e(t_n)$$

$$\sum_{i=0}^{n-1} (e(t_i) + e(t_{i+1})) = e(t_0) + e(t_n) + 2 \sum_{i=1}^{n-1} e(t_i)$$

Soit finalement :

$$\mathcal{A} = \frac{T}{2} \left[ e(t_0) + e(t_n) + 2 \sum_{i=1}^{n-1} e(t_i) \right] = T \frac{e(t_0) + e(t_n)}{2} + T \sum_{i=1}^{n-1} e(t_i)$$

## 1.V. Remarque sur les trois méthodes

Plus  $T_i$  sera petit, soit plus il y aura de points, plus les résultats seront proches de la réalité. Mais évidemment, la complexité étant en  $O(n)$ , le temps de calcul augmentera avec la précision :

$$\lim_{n \rightarrow \infty} \mathcal{A} = \int_{t_0}^{t_f} e(t) dt$$

## 1.VI. Fonction python préprogrammée

```
from scipy.integrate import quad
def f(x):
    return x
a = 1
b = 2
Res, Precision = quad(f, a, b)
print(Res)
```

Remarque : Infini possible via « `numpy.inf` ».

