A.V. Application des bases

A.V.1 Listes et chaînes de caractères

Le programme d'IPT mentionne le calcule d'extrema d'une liste, de sa moyenne et de sa variance. Nous aborderons ces algorithmes en TD/TP, nous ne donnerons donc ici que les principes de ces algorithmes.

A.V.1.a Extrema – Moyenne – Variance

A.V.1.a.i Recherche d'un extrema

Le principe de la recherche d'un minimum/maximum dans une liste est très simple. On peut se rapporter à un paquet de copies dans lequel on cherche la note la plus basse/haute.

Pour un maximum, par exemple, on réalise les opérations suivantes :

- On retient la première note du paquet, par exemple 12
- On passe les copies les unes après les autres, et dès qu'il y a une note supérieure à la première, par exemple 14, on change la note enregistrée dans notre mémoire en 14, et on continue
- A la fin du paquet, la note maximale est en mémoire

A.V.1.a.ii Calcul de la moyenne

Calculer une moyenne consiste à sommer les termes d'une liste les uns après les autres, puis à diviser le résultat obtenu par le nombre de termes. Nous avons eu l'occasion de mettre en pratique ce calcul dans le TP sur le lissage d'une courbe de poids obtenue par une balance Withings.

A.V.1.a.iii Calcul de la variance

La variance est la moyenne de l'écart au carré entre chaque valeur d'une liste et sa moyenne.

Soit N le nombre de termes appelés $L_i, i \in (1, N)$ d'une liste L de moyenne M :

$$V = \frac{1}{N}\sum_{i=1}^{N}(L_i - M)^2$$

En termes de programmation, il suffit donc par exemple de :

- Créer une fonction de calcul de moyenne permettant de calculer la moyenne M de L
- Créer une fonction qui calcul une nouvelle liste L' correspondant à la liste L à laquelle on a soustrait M et on élève au carré le résultat
- Appeler la fonction de calcul de moyenne sur L'

A.V.1.b Recherche d'un mot

A.V.1.b.i Principe

Soit une chaîne de caractères $Texte$ de T termes, par exemple "$Je\ suis\ sûr\ que\ je\ suis\ Suisse$"

On recherche dans cette chaîne de caractère un $Motif$ de M lettres, par exemple "$suis$"

Le principe de cette recherche consiste à :

- Placer le motif tel que sa première lettre soit à une position d'indice i allant de 0 à $T - M$ inclus, soit $range(T - M + 1)$!
- Créer un indice j initialisé à 0 pour chaque valeur de i
- Créer une boucle qui incrémente j tant qu'il y a concordance entre le caractère en position i du texte et le caractère en position j du motif
- Incrémenter un compteur chaque fois que la concordance est vérifiée pour les M caractères du motif
- Renvoyer le compteur à la fin

A.V.1.b.ii Exemple

	J	e		s	u	i	s		s	û	r		q	u	e		j	e		s	u	i	s		S	u	i	s	s	e
$i = 0$	s	u	i	s																										
$i = 3$				s	u	i	s																							
$i = 8$									s	u	i	s																		
$i = 8$																									s	u	i	s		
$i = 24$																											s	u	i	s

Voyons les détails ci-dessous :

- Lorsque $i = 0$, dès la première valeur de $j = 0$, la lettre 0 du Texte « J » n'est pas la même que la lettre 0 du motif « s », on passe au i suivant
- Lorsque $i = 3$
  - à la première valeur de $j = 0$, la lettre 0 du Texte « s » concorde avec la lettre 0 du motif « s », on incrémente j qui vaut 1
  - la lettre 1 du Texte « u » concorde avec la lettre 1 du motif « u » , on incrémente j qui vaut 2
  - la lettre 2 du Texte « i » concorde avec la lettre 2 du motif « i » , on incrémente j qui vaut 3
  - la lettre 3 du Texte « s » concorde avec la lettre 3 du motif « s » , on incrémente j qui vaut 4
  - j étant supérieur au dernier indice du motif, on incrémente le compteur (occurrence du motif) et on passe au i suivant
- Et ainsi de suite

Remarque : Attention, lettres majuscules et minuscules sont des caractères différents, l'algorithme appliqué à l'exemple ci-dessus renverra 2 !

A.V.2 Dichotomie

Le principe de la dichotomie est de diviser le domaine de recherche par 2 à chaque itération.

A.V.2.a Recherche dans un tableau trié

A.V.2.a.i Principe

Soit une liste (un tableau) L de N éléments. Appelons E l'élément recherché supposé présent une seule fois dans L.

Le principe de la recherche par dichotomie dans une liste triée est le suivant :

- Définir la plage d'indices de recherche notée $[I_g, I_d]$ (pour Indice gauche, Indice droite) qui au départ vaut $[0, N-1]$
- Déterminer l'indice milieu I_m de la plage de recherche : $I_m = \text{int}\left(\frac{I_g+I_d}{2}\right) = (I_g + I_d)\%2$
- Déterminer la valeur E_m de l'élément de la liste L à l'indice I_m
- Comparer la valeur E_m à la valeur recherchée E et, la liste étant triée :
  - Si $E = E_m$, l'élément recherché a été trouvé, son indice est I_m
  - Si $E < E_m$, définir la nouvelle plage de recherche à $[I_g, I_m - 1]$
  - Si $E > E_m$, définir la nouvelle plage de recherche à $[I_m + 1, I_d]$
- Répéter la procédure jusqu'à ce que l'élément soit obtenu (on a supposé qu'il existait)

A.V.2.a.ii Exemple

Soit le tableau trié suivant :

1	3	7	11	15	22	25	31	36	47	52	59	66	71	82	83	94	100	110	111
0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*

Recherchons la valeur $E = 25$.

D'après les notations précédentes : $I_g = 0, I_d = 19 \Rightarrow I_m = 9 \Rightarrow E_m = 47$. Comme $E < E_m$, le nouvel intervalle de recherche est $[0,8]$

1	3	7	11	15	22	25	31	36	47	52	59	66	71	82	83	94	100	110	111
0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*

$I_g = 0, I_d = 8 \Rightarrow I_m = 4 \Rightarrow E_m = 15$. Comme $E > E_m$, le nouvel intervalle de recherche est $[5,8]$

1	3	7	11	15	22	25	31	36	47	52	59	66	71	82	83	94	100	110	111
0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*

$I_g = 5, I_d = 8 \Rightarrow I_m = 6 \Rightarrow E_m = 25$. Comme $E = E_m$, on a trouvé l'élément recherché. Son indice est $I_m = 6$ et c'est le 7° élément de L

1	3	7	11	15	22	25	31	36	47	52	59	66	71	82	83	94	100	110	111
0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*

A.V.2.a.iii Complexité

Comme nous l'avons vu lors du calcul de complexités en temps des algorithmes dans le cas d'une dichotomie sur une liste de N termes, le nombre d'opérations n permettant à la fin d'aboutir à un seul terme dans la liste est solution de l'équation :

$$\frac{N}{2^n} = 1 \Leftrightarrow 2^n = e^{n \ln 2} = N \Leftrightarrow n = \frac{\ln N}{\ln 2}$$

Ainsi, la complexité d'un algorithme en dichotomie est en $o(\ln N)$. C'est bien évidemment moins couteux qu'une recherche sur les termes de la liste les uns après les autres, qui dans le pire des cas est en $o(N)$.

A.V.2.b Recherche du zéro d'une fonction

A.V.2.b.i Contexte

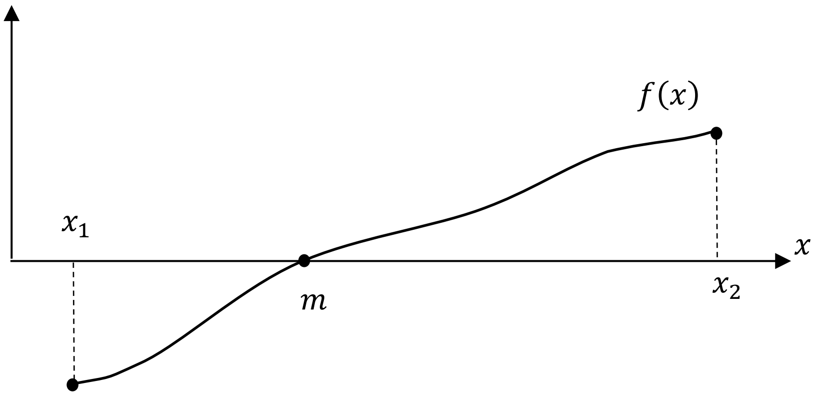
Soit une fonction $f(x)$ définie et continue sur un intervalle $[x_1, x_2]$ telle qu'il existe m tel que :

$$\forall a < m, \forall b > m, f(a)f(b) < 0$$

On se placera ici dans le cas où la fonction est monotone sur l'intervalle d'étude $[x_1, x_2]$.

Autrement dit, la fonction possède un signe sur $[x_1, m]$ et le signe opposé sur $[m, x_2]$.

Exemple :



A.V.2.b.ii Objectif

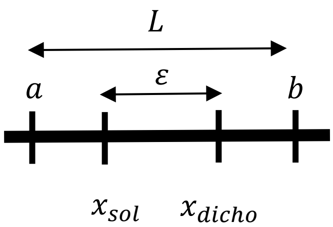
On souhaite déterminer une solution approchée de l'équation :

$$f(x) = 0$$

Appelons x_{sol} la solution exacte de cette équation. Par dichotomie, nous allons déterminer x_{dicho} tel que :

$$x_{sol} \in [a, b], x_{dicho} \in [a, b]$$

$$L = |b - a| \leq Critere \Rightarrow \varepsilon = |x_{dicho} - x_{sol}| \leq ou < Critere$$



Où $Critere$ est un réel définissant la précision de la solution obtenue.

A.V.2.b.iii Principe

Le principe de cette recherche par dichotomie est le suivant :

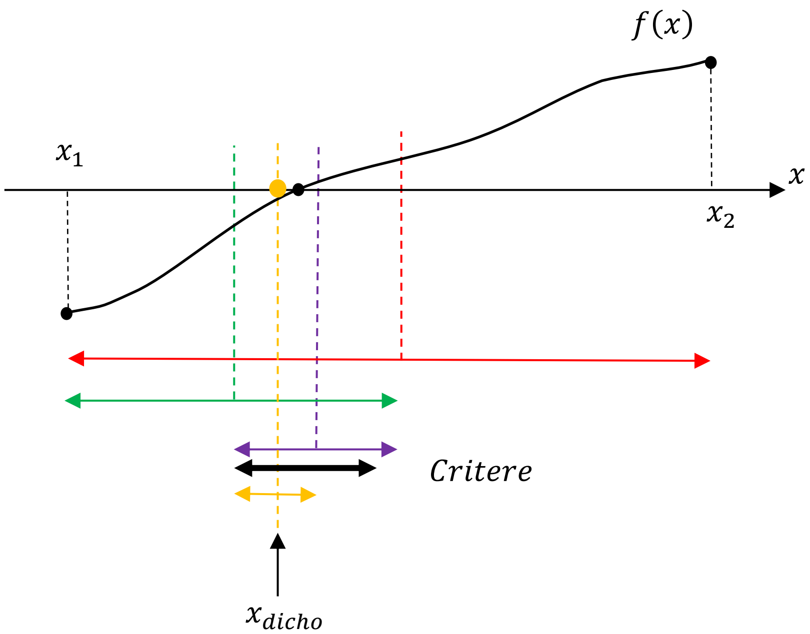
- Vérifier l'existence de la solution (facultatif) sur $[x_1, x_2]$
- Déterminer l'abscisse au centre de l'intervalle x_m et calculer son image par $f_m = f(x_m)$
- Identifier dans quel intervalle $[x_1, x_m]$ ou $[x_m, x_2]$ se trouve la solution de $f(x) = 0$
- Définir le nouvel intervalle de recherche $[x_1, x_2]$ comme celui dans lequel la solution existe
- Continuer tant que l'intervalle de recherche est de largeur supérieure au critère précisé
- Renvoyer une valeur de x dans l'intervalle obtenu lorsque le critère est vérifié, en général on renverra la valeur au centre x_m

Remarque : la solution $f(x) = 0$ existe sur l'intervalle $[x_1, x_2]$ si et seulement si $f(x_1)f(x_2) \leq 0$ – Le « ou égale » est important

A.V.2.b.iv Exemple

Soit un critère tel que la longueur de l'intervalle final soit inférieure à la longueur de la flèche ci-dessous :

On procède par itérations successives à la division par 2 de l'intervalle de recherche :



A.V.2.b.v Remarque

Il est envisageable de proposer un critère d'arrêt sur les ordonnées $|f(x)|$ plutôt que sur les abscisses.

A.V.3 Newton

La méthode de Newton est une seconde méthode de résolution d'équation de la forme $f(x) = 0$ qui converge plus vite que la méthode de dichotomie vers la solution recherchée.

A.V.3.a Contexte

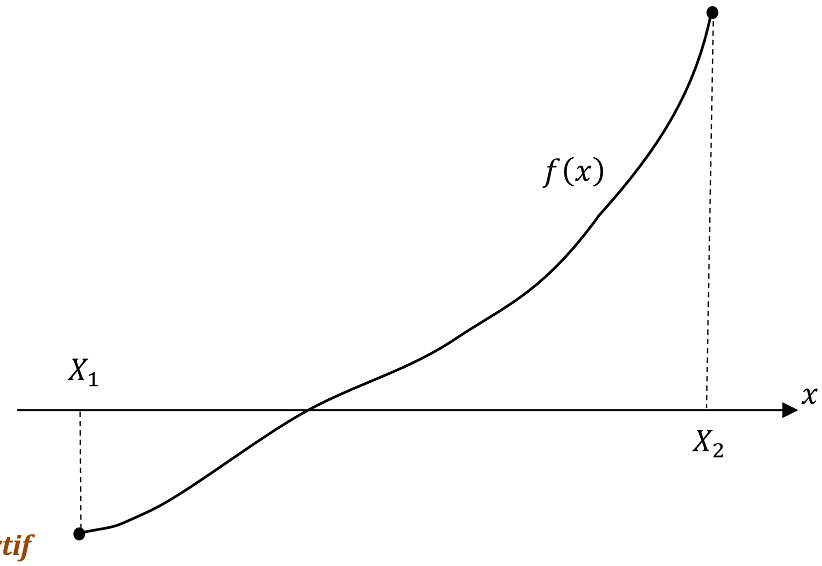
Soit une fonction $f(x)$ définie, continue et dérivable sur un intervalle $[X_1, X_2]$ telle qu'il existe m tel que :

$$\forall a < m, \forall b > m, f(a)f(b) < 0$$

On se placera ici dans le cas où la fonction est monotone sur l'intervalle d'étude $[X_1, X_2]$.

Autrement dit, la fonction possède un signe sur $[X_1, m]$ et le signe opposé sur $[m, X_2]$.

Exemple :



A.V.3.a.i Objectif

On souhaite déterminer une solution approchée de l'équation :

$$f(x) = 0$$

Contrairement à la résolution par Dichotomie vue au paragraphe précédent, nous ne sommes pas en mesure d'être sûrs que la solution approchée par la méthode de Newton est à une distance précise de la solution réelle.

Nous allons suivre l'écart entre deux solutions successives et proposer un critère d'arrêt lorsque deux solutions successives sont « assez proches ».

A.V.3.a.ii Principe

Le principe de la méthode de Newton consiste à approcher la courbe de $f(x)$ avec sa tangente $T_0(x) = f(x_0) + f'(x_0)(x - x_0)$ en un point initial x_0 choisi arbitrairement. On déterminer alors l'abscisse du point d'intersection $(x_1, 0)$ de cette tangente avec l'axe des abscisses, soit $T_0(x_1) = 0$.

On procède alors ainsi par itérations :

$$x_{i+1} / T_i(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) = 0$$

Soit :

$$f(x_i) + f'(x_i)x_{i+1} - x_i f'(x_i) = 0$$

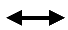
$$x_{i+1} = \frac{x_i f'(x_i) - f(x_i)}{f'(x_i)} = x_i - \frac{f(x_i)}{f'(x_i)}$$

On procède alors ainsi jusqu'à ce que :

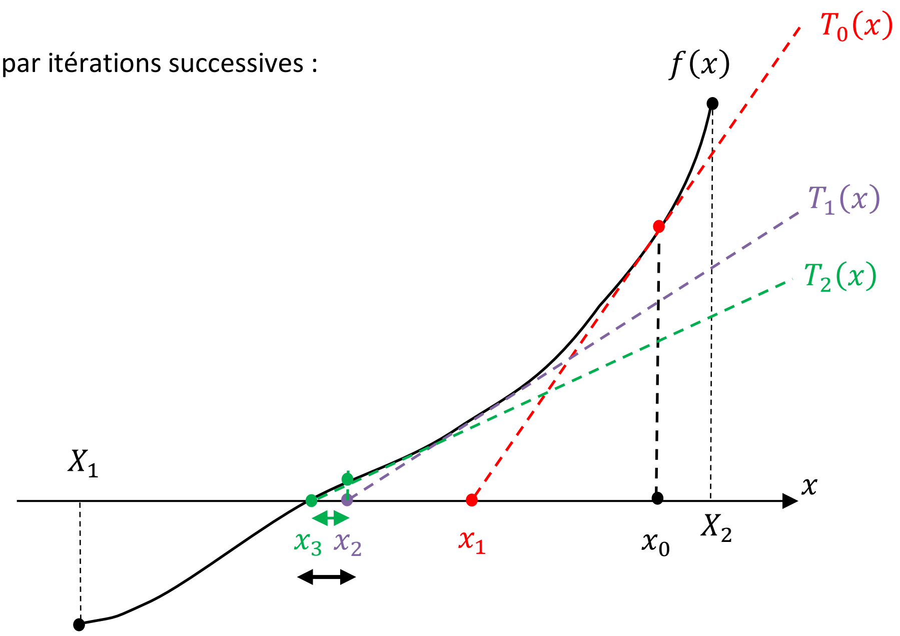
$$|x_{i+1} - x_i| \leq ou < Critere$$

A.V.3.a.iii Exemple

Soit un critère tel que la longueur de l'intervalle final soit inférieure à la longueur de la flèche ci-dessous :



On procède par itérations successives :

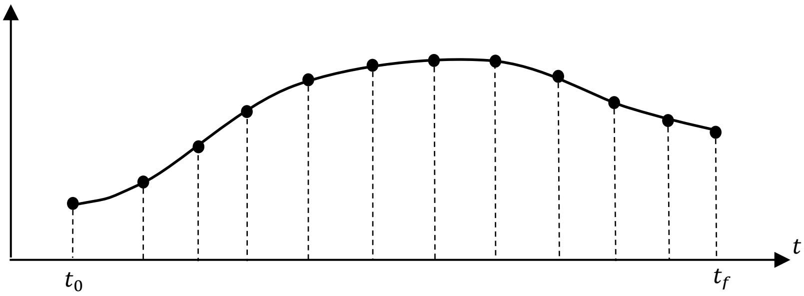


A.V.3.a.iv Remarque

Il est envisageable de proposer un critère d'arrêt sur les ordonnées $|f(x)|$ plutôt que sur les abscisses.

A.V.4 Intégration numérique

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 :

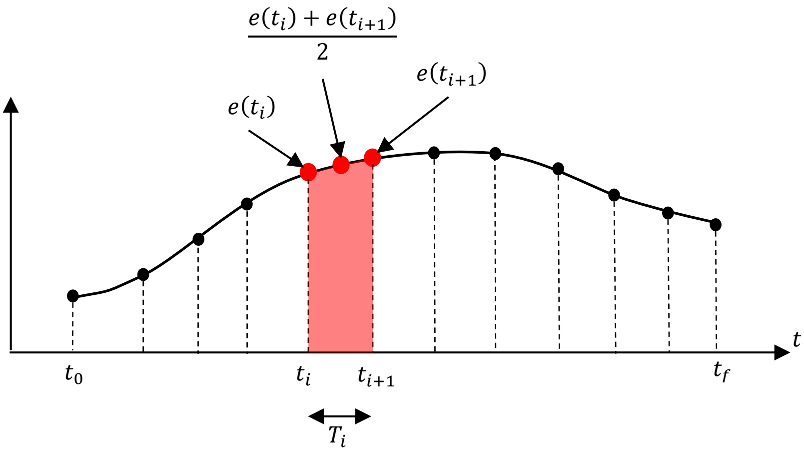
$$e(t_i)$$

- Valeur à droite :

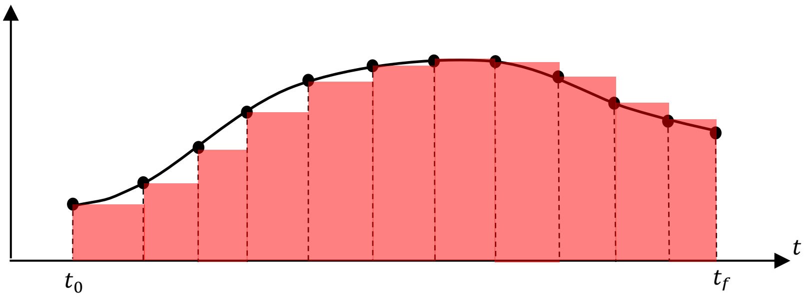
$$e(t_{i+1})$$

- Valeur centrée :

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



A.V.4.a Valeur à gauche – Méthode des rectangles

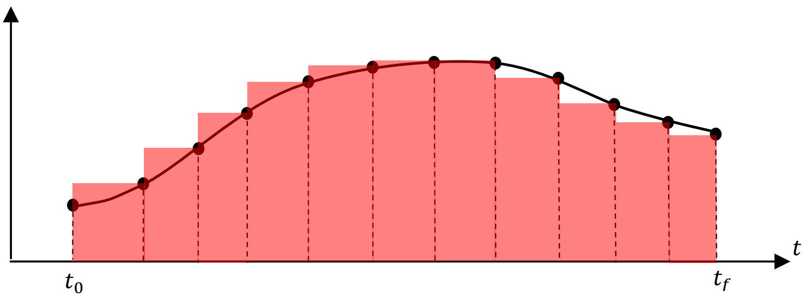


$$\mathcal{A} = \int_{t_0}^{t_f} e(t)dt = \sum_{i=0}^{n-1} T_i e(t_i) = \sum_{i=0}^{n-1} (t_{i+1} - t_i) e(t_i)$$

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.

A.V.4.b Valeur à droite - Méthode des rectangles



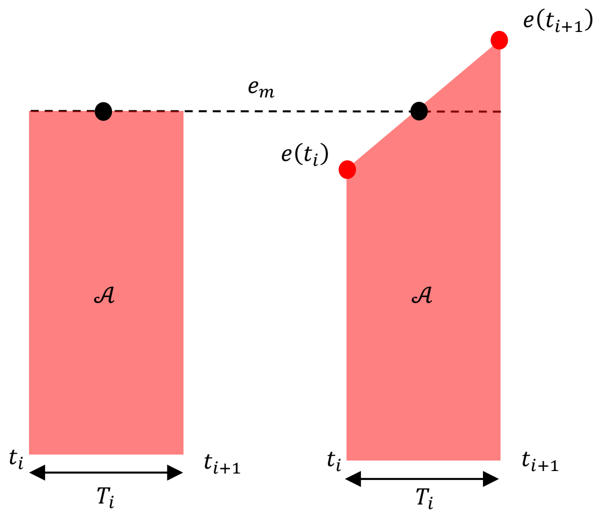
$$\mathcal{A} = \int_{t_0}^{t_f} e(t)dt = \sum_{i=0}^{n-1} T_i e(t_{i+1}) = \sum_{i=0}^{n-1} (t_{i+1} - t_i) e(t_{i+1})$$

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.

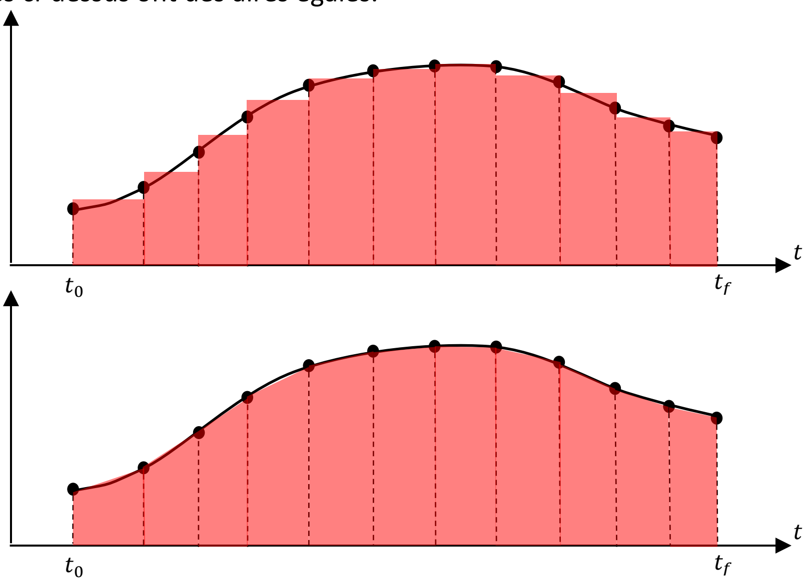
A.V.4.c Valeur centrée – Méthode des trapèzes

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 :



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

Les deux surfaces ci-dessus ont des aires égales.



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

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.

A.V.4.d Remarque

Plus T_i sera petit, soit plus il y aura de points, plus les résultats seront proches de la réalité.