Informatique en PCSI et MPSI
Champollion 2013-2014
Méthodes d'Analyse Numériques
Implémentation et Application en Python
Intégration Numérique

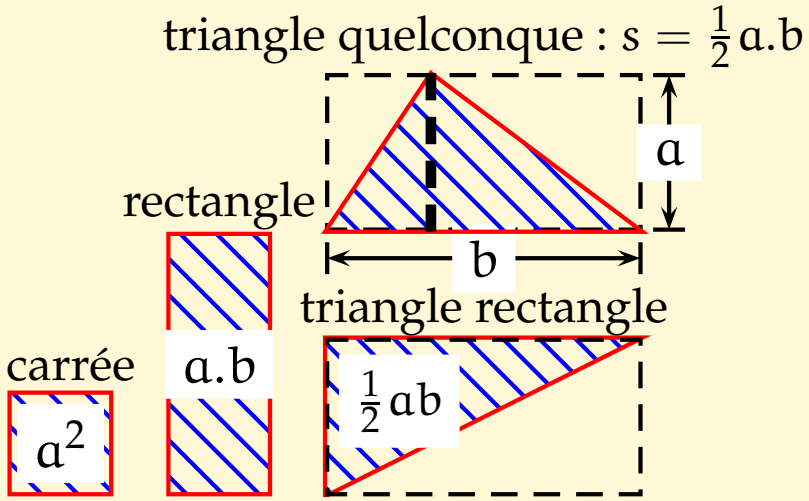
A. HASSAN

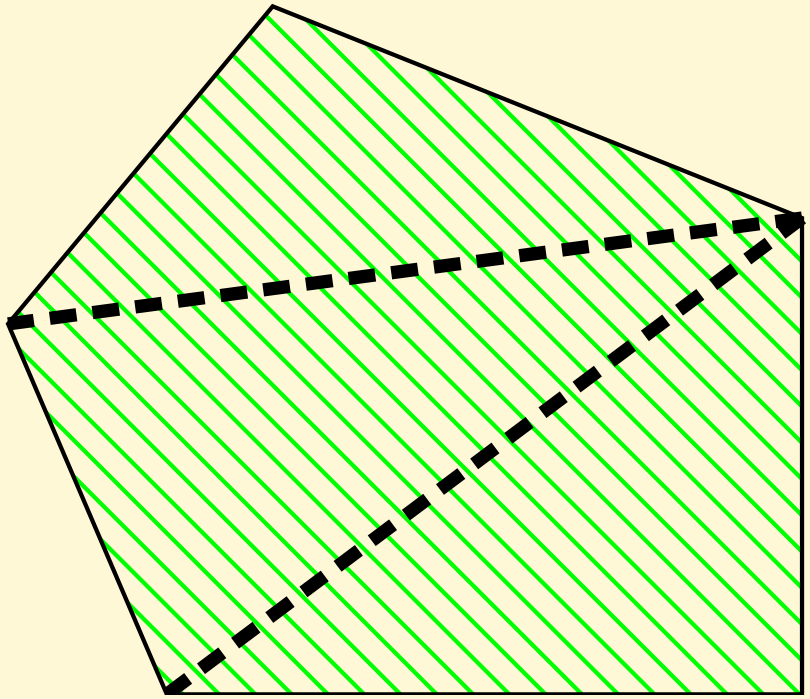
23 avril 2014

Intégration Numérique (Quadrature)

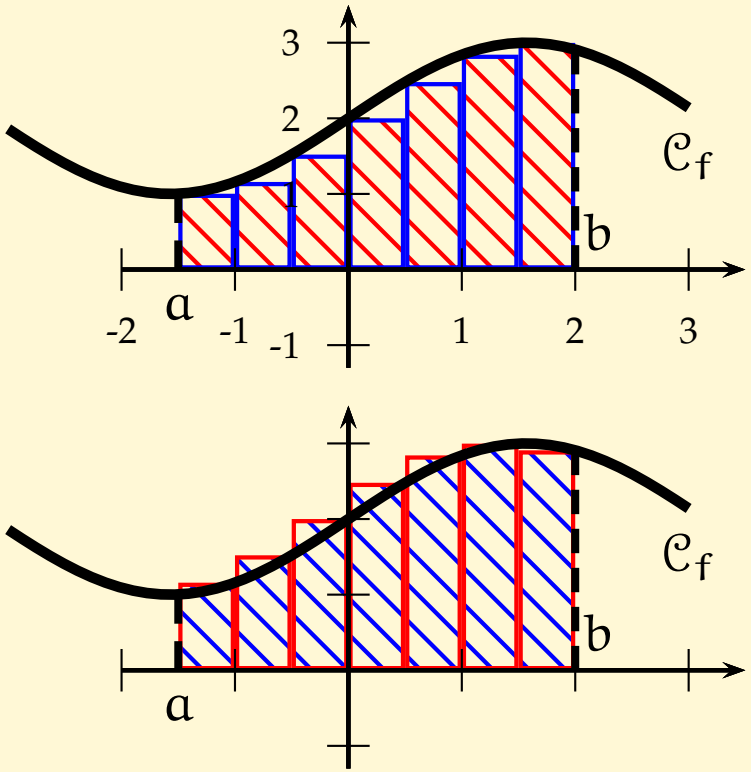
Intégration Numérique (Quadrature)
L'intégration : une histoire d'Aire
L'intégration : une Histoire d'Aire
Introduction
Pourquoi l'intégration numérique?
Le problème
Le Problème (suite)
Polynômes d'interpolation de Lagrange
Méthodes Élémentaires de Quadrature (de Newton-Cotes)
(Suite)
(suite)
Méthodes d'ordre zéro: Exemple des Rectangles gauches
Méthodes d'ordre zéro: Rectangles Droits
Méthodes des **Rectangles Milieux**
Implémentation en Python (à compléter)
Implémentation en Python
Méthode des rectangles: Calcul de l'erreur
Rectangle des milieux: Calcul de l'erreur

Aire par Triangulation

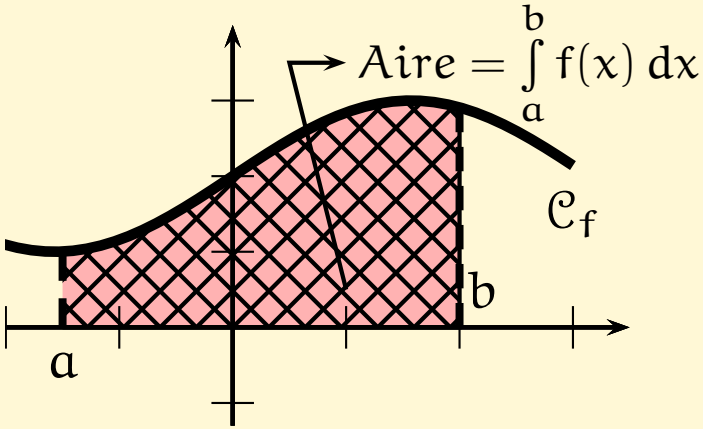




Le cas des contours courbés



$\xrightarrow[n\to+\infty]{}$

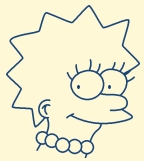


Intégrale (de Riemann) **sur un segment** : f est intégrable sur $[a;b]$, ssi pour toute subdivision $\sigma_n = (\underbrace{x_0}_{=a} < x_1 < \cdots < \underbrace{x_n}_{=b})$ les sommes ci-dessous convergent

$$\sum_{i=0}^{i=n-1} (x_{i+1} - x_i) f(t_i) \xrightarrow[n\to+\infty]{\forall t_i \in [x_i;x_{i+1}]} L \left(\text{noté} \int_{x=a}^{x=b} f(x)\,dx \right)$$

Théorème fondamental de l'Analyse :
Soit f continue sur $[a;b]$ et F une primitive de f alors :

$$\int_a^b f(t)\,dx = F(b) - F(a)$$



Le calcul d'une intégrale devient un calcul de primitive que l'on peut exprimer, si possible, à l'aide des fonctions élémentaires et leurs composées.

Pourquoi l'intégration numérique ?

1. **Impossibilité** d'exprimer une primitive à l'aide des fonctions usuelles :

$$\int_0^{2013} \frac{1}{\sqrt{1+x^4}}\,dx; \qquad \int_0^{2014} e^{-x^2}\,dx; \qquad \int_0^{2} \frac{\cos(x)}{x+6}\,dx;$$

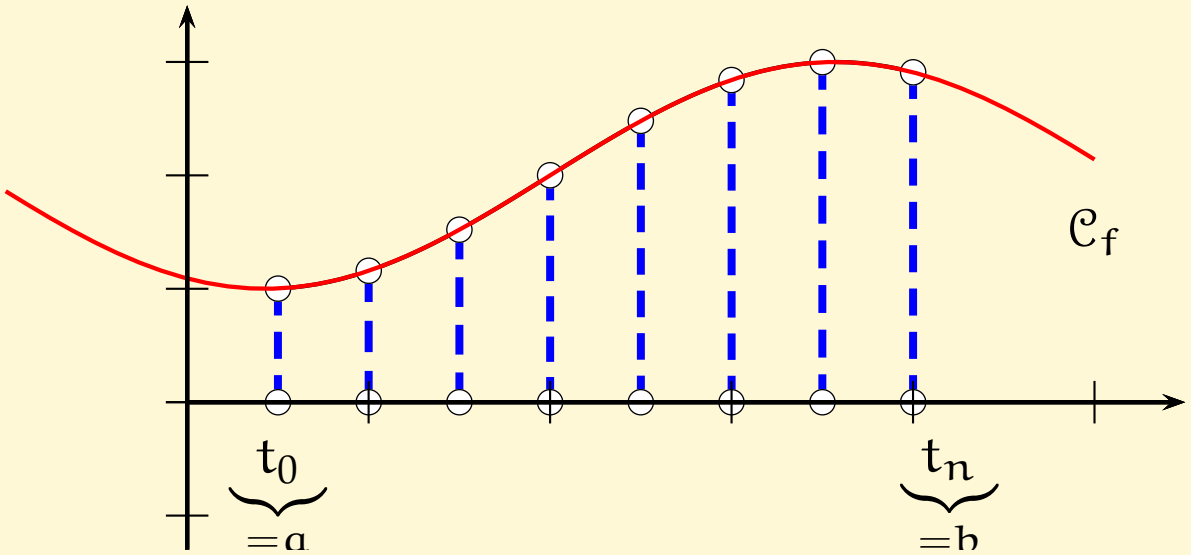
2. **Primitive compliquée** : coût d'évaluation élevé

$$\int_0^{x} \frac{1}{1+t^4}\,dt = \frac{\sqrt{2}}{8}\ln\left(\frac{x^2+x\sqrt{2}+1}{x^2-x\sqrt{2}+1}\right) + \frac{\sqrt{2}}{4}\left[\arctan\left(x\sqrt{2}+1\right)+\arctan\left(x\sqrt{2}-1\right)\right]$$

3. **Expression analytique de** f **non disponible** : f est donnée sous forme d'une table de valeurs i.e. $(x_i; f(x_i))$ (Exemple : mesures physiques).

Intégration Numérique (Quadrature)
L'intégration : une histoire d'Aire
L'intégration : une Histoire d'Aire
Introduction
Pourquoi l'intégration numérique?
Le problème
Le Problème (suite)
Polynômes d'interpolation de Lagrange
Méthodes Élémentaires de Quadrature (de Newton-Cotes)
(Suite)
(suite)
Méthodes d'ordre zéro: Exemple des Rectangles gauches
Méthodes d'ordre zéro: Rectangles Droits
Méthodes des **Rectangles Milieux**
Implémentation en Python (à compléter)
Implémentation en Python
Méthode des rectangles: Calcul de l'erreur
Rectangle des milieux: Calcul de l'erreur

Données : En général on dispose des valeurs (abscisses) $(x_i)_{1\leqslant i\leqslant n}$ (subdivision) dans $[a;b]$ et leurs images $(f(x_i))_{1\leqslant i\leqslant n}$



Deux méthodes principales :

1. **Subdivision régulière** de $[a;b] \Rightarrow x_i = x_{i-1} + h = a + h.i$ où $h = (b-a)/n$
2. **À la Gauss** : les points de la subdivision sont fixés à l'avance : Racines de famille orthogonale de polynômes.

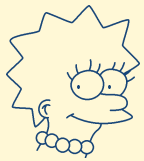
Intégration Numérique (Quadrature)
L'intégration : une histoire d'Aire
L'intégration : une Histoire d'Aire
Introduction
Pourquoi l'intégration numérique?
Le problème
Le Problème (suite)
Polynômes d'interpolation de Lagrange
Méthodes Élémentaires de Quadrature (de Newton-Cotes)
(Suite)
(suite)
Méthodes d'ordre zéro: Exemple des Rectangles gauches
Méthodes d'ordre zéro: Rectangles Droits
Méthodes des **Rectangles Milieux**
Implémentation en Python (à compléter)
Implémentation en Python
Méthode des rectangles: Calcul de l'erreur
Rectangle des milieux: Calcul de l'erreur

Soit $I = [a; b]$ et f une fonction définie sur I, on cherche à évaluer numériquement : $I(f) = \int_a^b f(t)\,dt$

Soit $T = (\underbrace{x_0}_{=a} < x_1 < x_2 < \cdots < x_{n-1} < \underbrace{x_n}_{=b})$ une subdivision (pas nécéssairement régulière) de $I = [a; b]$

On suppose les valeurs de $f(x_i)$ connues

Problème : Trouver $(\omega_i)_{0\leqslant i\leqslant n}$ tels que $I(f) = \int_a^b f(x)\,dx \approx S = \sum_{i=0}^{i=n} \omega_i f(x_i)$



Trouver $(\omega_i)_{0\leqslant i\leqslant n}$les **coefficients de pondération** qui minimisent $\left|I(f) - \sum_{i=0}^{i=n} \omega_i f(x_i)\right|$?

Polynômes d'interpolation de Lagrange

Soit les $n+1$ points $(x_k, f(x_k))_{0\leqslant k\leqslant n}$ tels que : $\quad x_0 < x_1 < x_2 < \cdots < x_n$
Trouver $P \in \mathbb{K}_n[X]$ tel que $P(x_k) = f(x_k)$ pour $k \in [\![\, 0, n \,]\!]$?.
Une solution(polynômes de Lagrange) :

$$L_k(X) = \frac{\prod\limits_{\substack{1\leqslant j\leqslant n \\ k\neq j}} (X - x_j)}{\prod\limits_{\substack{1\leqslant j\leqslant n \\ k\neq j}} (x_k - x_j)} \qquad \Rightarrow L_k(x_i) = \delta_{ki} = \begin{cases} 1 & \text{si } k = i \\ 0 & \text{sinon} \end{cases}$$

Les $(L_k)_{1\leqslant k\leqslant n}$ forment une base de $\mathbb{K}_n[X]$ (le démontrer ? ?),

$$P(X) = f(x_0).L_0(X) + f_1(x_1).L_1(X) + \cdots + f(x_n).L_n(X) = \sum_{k=0}^{k=n} f(x_k).L_k(X)$$

est **LE** polynôme interpolant f aux points $(x_k; f(x_k))_{0\leqslant k\leqslant n}$

Méthodes Élémentaires de Quadrature (de Newton-Cotes)

Sur un **intervalle élémentaire** $[x_i; x_{i+1}]$, la fonction f est **remplacée** par un polynôme interpolateur P_n de degré n.
La méthode renvoie **un résultat exact** si f est un polynôme de degré n.

On pose $x_{i+1} - x_i = h = \dfrac{b-a}{N}$ (pas régulier)

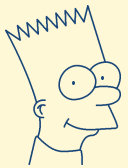
Définition : Une méthode d'intégration est dite d'ordre n, si elle renvoie un résultat exact pour tout $f \in \mathbb{R}_n[X]$ et non exact pour un élément de $\mathbb{R}_{n+1}[X]$.
Voici les plus célèbres :

Méthode des rectangles où le Polynôme interpolateur est **constant**

$P_0(x) = f(x_i)$ Rectangles gauches . . . . . Ordre 0

$P_0(x) = f(x_{i+1})$ Rectangles droits . . . . . Ordre 0

$P_0(x) = f(\frac{1}{2}(x_i + x_{i+1}))$, Rectangles milieux . . . . . Ordre 1



Remarque : Ces méthodes font référence aux **sommes de Riemann**.

Méthode d'ordre 1, ou des **trapèzes** :

Polynôme interpolateur de degré 1 (droite affine)

$$\begin{aligned} P_1(x) &= \frac{x - x_{i+1}}{x_i - x_{i+1}} f(x_i) + \frac{x - x_i}{x_{i+1} - x_i} f(x_{i+1}) \\ &= \frac{1}{h} \left[(x - x_i) f(x_{i+1}) - (x - x_{i+1}) f(x_i) \right] \\ &= \frac{1}{h} \left[(x - x_i) y_{i+1} - (x - x_{i+1}) y_i \right] \end{aligned}$$

On verra que cette méthode est d'ordre 1, i.e. elle est exacte pour les fonctions affines (polynômes de degré 1)

Méthode d'ordre 2, ou de **SIMPSON** :

Polynôme interpolateur de degré 2 (parabole). On interpole les points $(x_i, f(x_i))$, $(x_{i+1}; f(x_{i+1}))$ et $(x_{i+2}; f(x_{i+2}))$

$$P_2(x) = \frac{(x-x_{i+2})(x-x_{i+1})}{(x_i-x_{i+1})(x_i-x_{i+2})} f(x_i) + \frac{(x-x_i)(x-x_{i+2})}{(x_{i+1}-x_i)(x_{i+1}-x_{i+2})} f(x_{i+1}) + \frac{(x-x_i)(x-x_{i+1})}{(x_{i+2}-x_i)(x_{i+2}-x_{i+1})} f(x_{i+2})$$

$$P_2(x) = \frac{(x-x_{i+2})(x-x_{i+1})}{h^2} y_i - \frac{(x-x_i)(x-x_{i+2})}{h^2} y_{i+1} + \frac{(x-x_i)(x-x_{i+1})}{2h^2} y_{i+2}$$

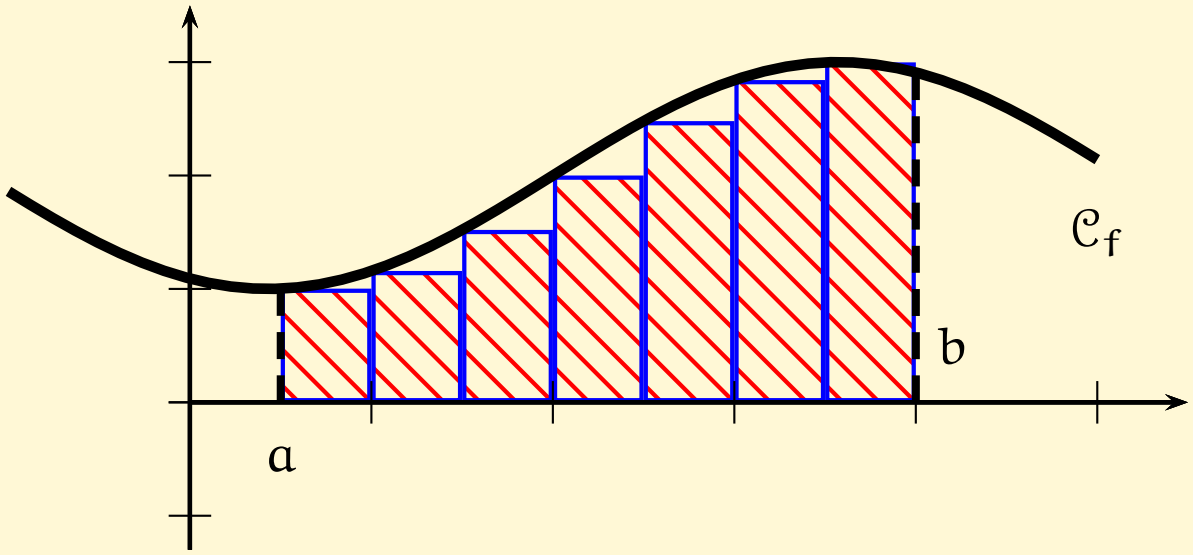
Remarque : D'autres approches interpolent les points d'abscisses x_i, x_{i+1} et leur milieu $\frac{1}{2}(x_i + x_{i+1})$

Cette méthode est exacte pour les polynômes de degré 2.

Méthodes d'ordre n ou de **Newton-Cotes** : On utilise un polynôme interpolateur de degré n.

Méthodes d'ordre zéro : Exemple des Rectangles gauches

Le polynôme interpolateur est constant $L_0(X)|_{[x_i;x_{i+1}]} = f(x_i) = y_i$

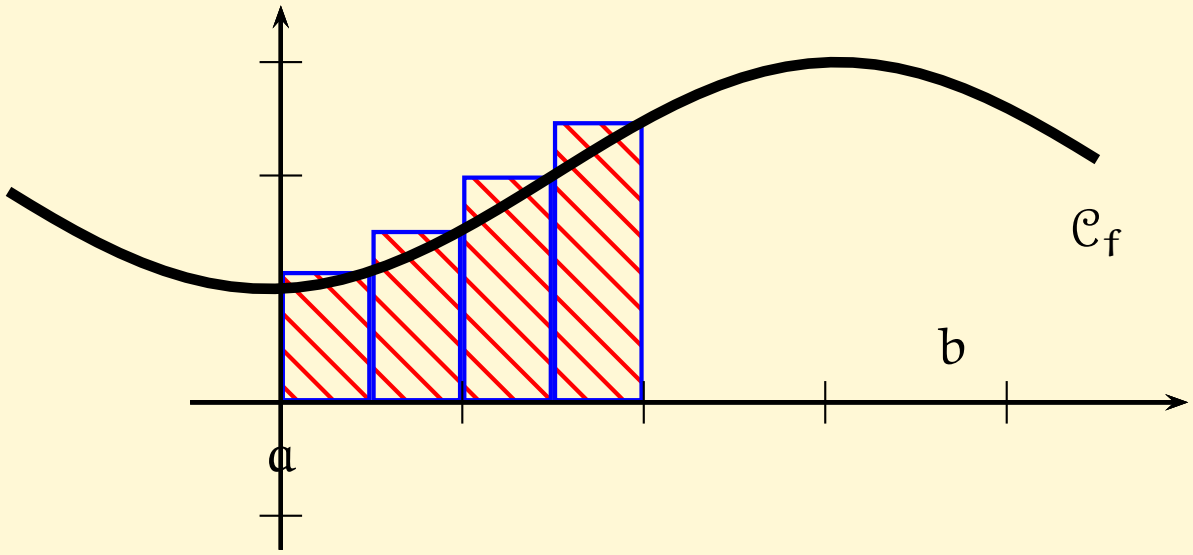


Intervalle élémentaire : $I_e = \int_{x_i}^{x_{i+1}} f(t)\,dt \approx \int_{x_i}^{x_{i+1}} P_0(t)\,dt \approx (x_{i+1} - x_i) f(x_i)$ (i.e. évaluation de f à gauche).

$$\text{Il vient : } I(f) = \int_a^b f(x)dx \approx S_R = \sum_{i=0}^{n-1} (x_{i+1} - x_i) f(x_i) = h \sum_{i=0}^{n-1} y_i$$

Méthodes d'ordre zéro : Rectangles Droits

Le polynôme interpolateur est constant $P_0(X)|_{[x_i;x_{i+1}]} = f(x_{i+1}) = y_{i+1}$

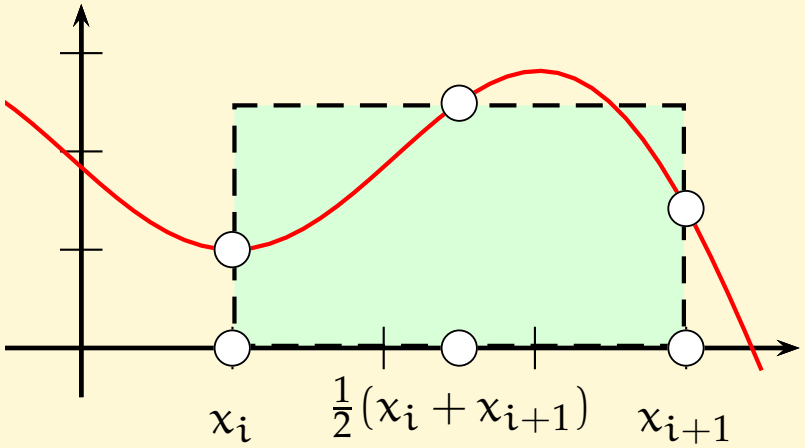


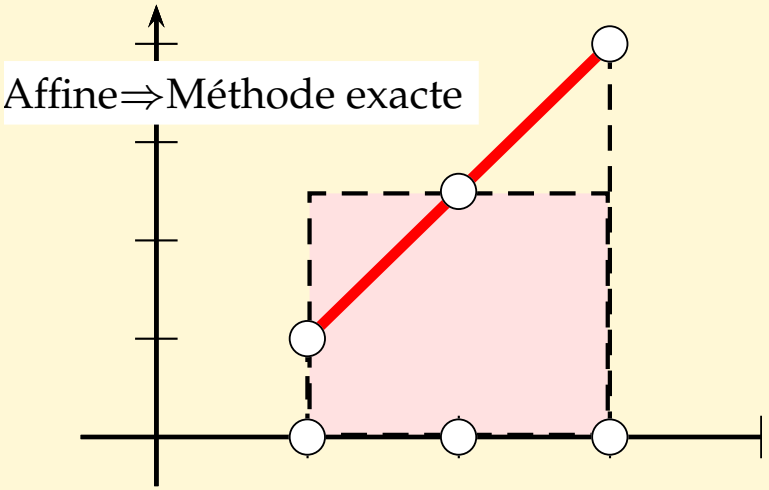
Intervalle élémentaire : $\int_{x_i}^{x_{i+1}} f(t)\,dt \approx (x_{i+1} - x_i) f(x_{i+1})$ (i.e. évaluation de f à gauche).

Il vient : $\int_a^b f(t)dt \approx \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} P_0(t)\,dt = \sum_{i=0}^{n-1} (x_{i+1} - x_i) f(x_{i+1}) = h \sum_{i=0}^{n-1} y_{i+1}$

Le polynôme interpolateur est toujours constant

$$P_0(X)|_{[x_i;x_{i+1}]} = \frac{1}{2}\left[f(x_i) + f(x_{i+1})\right]$$





Intervalle élémentaire $\int_{x_i}^{x_{i+1}} f(t)dt \approx (x_{i+1} - x_i) f(\frac{1}{2}(x_i + x_{i+1}))$

$$\int_a^b f(t)dt \approx \sum_{i=0}^{n-1} (x_{i+1} - x_i) f(\tfrac{1}{2}(x_i + x_{i+1})) = \boxed{h \sum_{i=0}^{n-1} f(x_i + \frac{h}{2})}$$

Implémentation en Python (à compléter)

Rectangles Gauches :

```
1 def LeftBox(f,a,b,N):
2     s,x,h=0,a,(b-a)/n
3     ..................
4     ..................
5     ..................
6     ..................
7     return s
```

Rectangles Droits :

```
1 def RightBox(f,a,b,n):
2     s,x,h=............
3     ..................
4     ..................
5     ..................
6     ..................
7     return s
```

Rectangles des Milieux :

```
1 def MiddleBox(f,a,b,n):
2     s,x,h=0,a,(b-a)/n
3     ..................
4     ..................
5     ..................
6     ..................
7     ..................
8     return s
```


Intégration Numérique (Quadrature)
L'intégration : une histoire d'Aire
L'intégration : une Histoire d'Aire
Introduction
Pourquoi l'intégration numérique?
Le problème
Le Problème (suite)
Polynômes d'interpolation de Lagrange
Méthodes Élémentaires de Quadrature (de Newton-Cotes)
(Suite)
(suite)
Méthodes d'ordre zéro: Exemple des Rectangles gauches
Méthodes d'ordre zéro: Rectangles Droits
Méthodes des **Rectangles Milieux**
Implémentation en Python (à compléter)
Implémentation en Python
Méthode des rectangles: Calcul de l'erreur
Rectangle des milieux: Calcul de l'erreur

Rectangle Gauches

```
1 def LeftBox(f,a,b,N):
2     s,x,h=0,a,(b-a)/n
3     for k in range(n):
4         s += h*f(x)
5         x +=h
6     return s
```

Rectangles des Milieux

```
1 def MiddleBox(f,a,b,n):
2     s,x,h=0,a,(b-a)/n
3     for k in range(n):
4         s +=h*f(x+h/2)
5         x+=h
6     return s
```

Rectangles Droits

```
1 def RightBox(f,a,b,n):
2     s,x,h=0,a,(b-a)/n
3     for k in range(n):
4         x += h
5         s += h*f(x)
6     return s
```

Intégration Numérique (Quadrature)
L'intégration : une histoire d'Aire
L'intégration : une Histoire d'Aire
Introduction
Pourquoi l'intégration numérique?
Le problème
Le Problème (suite)
Polynômes d'interpolation de Lagrange
Méthodes Élémentaires de Quadrature (de Newton-Cotes)
(Suite)
(suite)
Méthodes d'ordre zéro: Exemple des Rectangles gauches
Méthodes d'ordre zéro: Rectangles Droits
Méthodes des **Rectangles Milieux**
Implémentation en Python (à compléter)
Implémentation en Python
Méthode des rectangles: Calcul de l'erreur
Rectangle des milieux: Calcul de l'erreur

Méthode des rectangles : Calcul de l'erreur

Sur l'intervalle $I_{e_i} = [x_i; x_{i+1}]$

$$f(x) - y_i = f'(c_i)(x - x_i) \Rightarrow \int_{x_i}^{x_{i+1}} f(x)\,dx - hy_i = \int_{x_i}^{x_{i+1}} f'(c_i)(x - x_i)\,dx$$

Si f' bornée i.e. $\exists M = \sup_{c \in [a;b]} |f'(c)|$

$$\left| \int_{x_i}^{x_{i+1}} f(x)\,dx - hy_i \right| \leqslant M\frac{1}{2}(x_{i+1} - x_i)^2 = M\frac{(b-a)^2}{2n^2}$$

En sommant $\left| \int_a^b f(x)\,dx - h\sum_{i=0}^{n-1} y_i \right| \leqslant M\frac{(b-a)^2}{2n}$

Pour une précision de ε on prend $n > M\frac{(b-a)^2}{2\varepsilon}$

Remarque : La méthode des rectangles-milieux est d'ordre 1.

Rectangle des milieux : Calcul de l'erreur

Preuve : Par DL_2 au voisinage du point milieu $x_i + \frac{h}{2}$ et $\forall t \in I_i = [x_i; x_i + h]$, $\exists c_i \in I_i$ tel que

$$f(t) = f(x_i + \frac{h}{2}) + (t - x_i - \frac{h}{2}) f'(x_i + \frac{h}{2}) + \frac{1}{2}(t - x_i - \frac{h}{2})^2 f''(c_i)$$

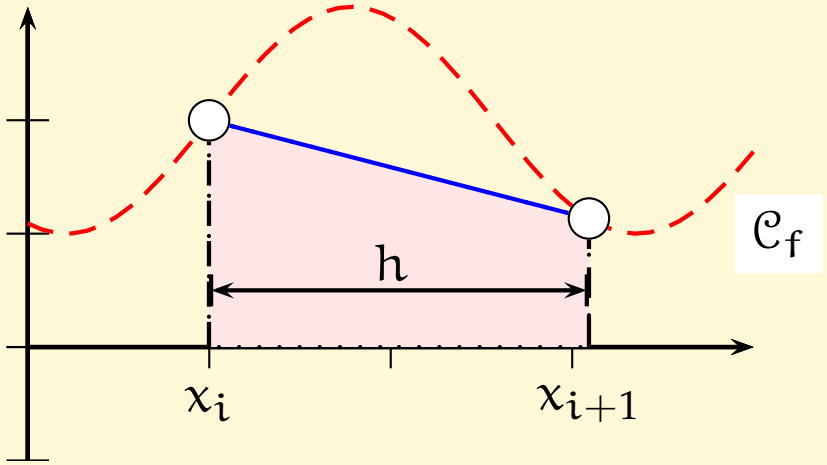
$$\int_{x_i}^{x_i+h} f(t)\,dt - hf(x_i + \frac{h}{2}) = \int_{x_i}^{x_i+h} (t - x_i - \frac{h}{2}) f'(x_i + \frac{h}{2}) + \frac{1}{2}(t - x_i - \frac{h}{2})^2 f''(c_i) dt$$

$$\xrightarrow[u = t - x_i - \frac{1}{2}h]{chang.\ var} \int_{x_i}^{x_i+h} (t - x_i - \frac{h}{2})\,dt = \int_{-\frac{1}{2}h}^{\frac{1}{2}h} u\,du = 0 \text{ et si } \sup_{x \in [a;b]} |f''(x)| \leqslant M$$

$$\left| \int_{x_i}^{x_{i+1}} f(x)\,dx - hf(x_i + \frac{h}{2}) \right| \leqslant \frac{M}{3} h^3$$

Par sommation : $$\left| \int_a^b f(x)\,dx - h \sum_{i=0}^{n-1} f(x_i + \frac{h}{2}) \right| \leqslant \frac{M}{3} \times nh^3 = M \frac{(b-a)^3}{3n^2}$$

Méthodes d'ordre un : méthode des Trapèzes



$$I_e = \int_{x_i}^{x_{i+1}} f(t)dt \approx (x_{i+1} - x_i)\frac{1}{2}.(y_i + y_{i+1}) = \frac{h}{2}(y_i + y_{i+1})$$

$$\int_a^b f(t)\,dt \approx S = \frac{h}{2}\left[(y_0 + y_1) + (y_1 + y_2) + \cdots + (y_{n-1} + y_n)\right]$$

$$S = h\left[-\frac{1}{2}(y_0 + y_n) + \sum_0^n y_i\right] \Rightarrow S = \frac{h}{2}\left[\frac{1}{2}(y_0 + y_n) + \sum_1^{n-1} y_i\right]$$

Méthode des Trapèzes : Calcul de l'erreur

Théorème : Si f est $\mathcal{C}^2$ telle que $\sup|f''(t)| < M$ alors $\boxed{|I(f)-S| \leqslant M\dfrac{(b-a)^3}{12n^2}}$

Preuve : Soit $[x_i;x_{i+1}] = [x_i;x_i+h]$, $L_2(x) = \dfrac{1}{h}\left[(x-x_i)y_{i+1} - (x-x_{i+1})y_i\right]$

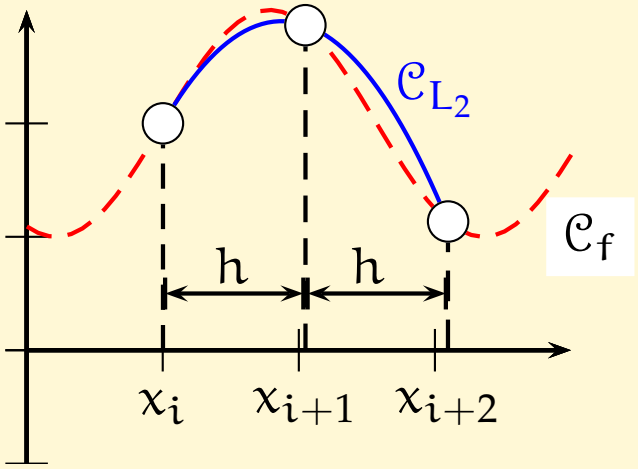
$$\begin{cases} \varphi(t) = f(t) - \dfrac{1}{2}M(t-x_i)(x_{i+1}-t) \Rightarrow \varphi''(t) = f''(t) + M \geqslant 0 \Rightarrow \varphi \text{ convexe} \\ \psi(t) = f(t) - \dfrac{1}{2}M(t-x_i)(x_{i+1}-t) \Rightarrow \psi \text{ est concave} \end{cases}$$

φ en dessous de L_2 et ψ en dessus

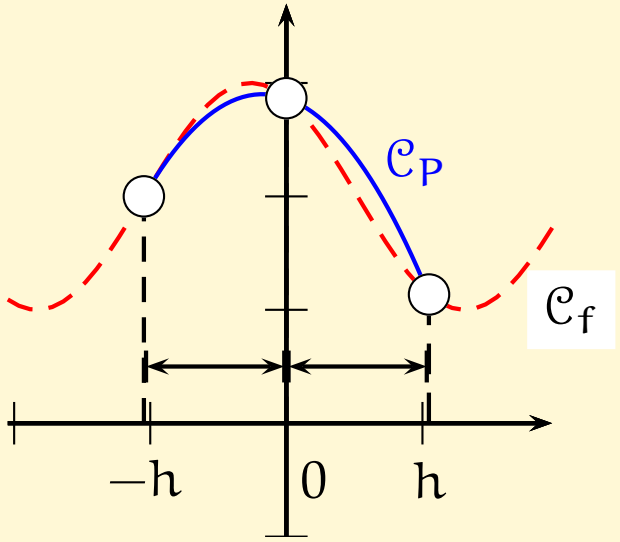
$$f(t) + \frac{1}{2}M(t-x_i)(x_{i+1}-t) \leqslant L_2 < f(t) - \frac{1}{2}M(t-x_i)(x_{i+1}-t) \overset{t\in[x_i;x_{i+1}]}{\Longleftrightarrow}$$

$$|f(t)-L_2(t)| \leqslant \frac{1}{2}M(t-x_i)(x_{i+1}-t) \xrightarrow[\text{intégration}]{\text{par}} \left|\int_{x_i}^{x_i+h} f(t)-L_2(t)\,dt\right| \leqslant \frac{M}{12}(x_{i+1}-x_i)^3$$

En sommant : $\left|\displaystyle\int_{x=a}^{x=b} f(x)\,dx\right| \leqslant \sum_{i=0}^{i=n-1} \frac{M}{12}\underbrace{(x_{i+1}-x_i)^3}_{=h=\frac{1}{n}(b-a)} \Rightarrow \boxed{|I(f)-S| \leqslant \frac{M}{12n^2}(b-a)^3}$



$\Longleftrightarrow$ changement var $x - x_{i+1} = t$



$$L_2(x) = \frac{(x-x_{i+1})(x-x_{i+2})}{h^2} y_i + \frac{(x-x_i)(x-x_{i+2})}{(-h^2)} y_{i+1} + \frac{(x-x_i)(x-x_{i+1})}{2h^2} y_{i+2}$$

Intégrale élémentaire :

$$I_e = \int_{x_i}^{x_{i+2}} f(x)\,dx \approx \int_{x_i}^{x_{i+2}} L_2(x)\,dx$$

$$I \approx \sum ()$$

Preuve : Soit $[x_i; x_{i+2}]$ tel que $\begin{cases} \deg(L_2) = 2 \\ y_i = f(x_i) = L_2(x_i) \\ y_{i+1} = f(x_{i+1}) = L_2(x_{i+1}) \\ y_{i+2} = f(x_{i+2}) = L_2(x_{i+2}) \end{cases}$

$$I_e = \int_{x_i}^{x_{i+2}} f(x)\,dx \approx \int_{x_i}^{x_{i+2}} L_2(x)\,dx$$

Par un changement de variable $t = x_i - h$ on se ramène à

$$I \approx \int_{-h}^{h} (at^2 + bt + c)\,dt \text{ avec } \begin{cases} y_i = ah^2 - bh + c \\ y_{i+1} = c \\ y_{i+2} = ah^2 + bh + c \end{cases}.$$

$$\xrightarrow[\text{Parité}]{\text{Par}} I_e \approx 2\int_{t=0}^{h} at^2 + c\,dt = \frac{2}{3}a.h^3 + 2ch$$

$$I_e \approx \frac{h}{3}\left(2ah^2+6c\right) \xrightarrow[y_i+y_{i+2}=2ah^2+2y_{i+1}]{y_i+y_{i+2}=2ah^2+2c} \boxed{I_e \approx \frac{h}{3}\left(y_i+4y_{i+1}+y_{i+2}\right)}$$

Il vient :

$$\begin{aligned} S = \frac{h}{3}[&y_0+4y_1+y_2\\ &+y_2+4y_3+y_4\\ &+y_4+4y_5+y_6\\ &+\vdots+\vdots+\vdots\\ &+y_{2n-2}+4y_{2n-1}+y_{2n}] \end{aligned}$$

Avec un nombre pair de subdivisions, $h = \frac{b-a}{2n}$:

$$\int_a^b f(x)\,dx \approx= \frac{h}{3}\left[y_{2n}-y_0+\sum_{i=0}^{n-1} 2y_{2i}+4y_{2i+1}\right]$$

Mise en œuvre de la méthode de SIMPSON(à compléter)

On choisit un nombre pair d'intervalles $\Rightarrow h = \dfrac{(b-a)}{2n}$ (donc $2n+1$ points).

SINON : l'interpolation est faite sur les points : x_i ; $m_i = \dfrac{1}{2}(x_i + x_{i+1})$ et x_{i+1} et

$$I_e = \int_{x_i}^{x_{i+1}} f(t)\,dt \approx \frac{h}{6}(y_i + 4f(m_i) + y_{i+1})$$

```
1  ##
2  def Simpson(f,a,b,n):
3      n2=2*n            # nombre pair de points
4      s,h=0,(b-a)/n2    # initialisation
5      x1=a              # abscisses d'ordre pair
6      for ::::::::::::::
7          ..................
8          ..................
9      ..................
10     ..................
11     ..................
12     return s*h/3
```

Mise en œuvre de la méthode de SIMPSON

On choisit un nombre pair d'intervalles $\Rightarrow h = \dfrac{(b-a)}{2n}$ (donc $2n+1$ points).

SINON : l'interpolation est faite sur les points : x_i ; $m_i = \dfrac{1}{2}(x_i + x_{i+1})$ et x_{i+1} et

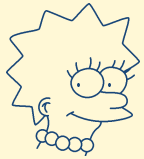
$$I_e = \int_{x_i}^{x_{i+1}} f(t)\,dt \approx \frac{h}{6}(y_i + 4f(m_i) + y_{i+1})$$

```
1  ##
2  def Simpson(f,a,b,n):
3      n2=2*n            # nombre pair de points
4      s,h=0,(b-a)/n2    # initialisation
5      x1=a              # abscisses d'ordre pair
6      for i in range(0,n):
7          s += 2*f(x1)+4*f(x1+h)
8          x1=x1+2*h
9      s += f(b)-f(a)
10     return s*h/3
```

Erreur de la méthode de Simpson, méthodes de Newton-Cotes

Théorème[admis] : Si la dérivée d'ordre 4 de f vérifie $\sup_{x\in[a;b]} |f^{(4)}(x)| \leqslant M$ alors

$$|I(f) - S_p| \leqslant M\frac{(b-a)^5}{n^4} \text{ avec } h = \frac{b-a}{n}$$



Les méthodes précédentes font partie des méthodes de **Newton-Cotes**, où l'on **interpole** $n+1$ **points** par un **polynôme de degré** n.

1. Newton-Cotes d'ordre 1 $\iff$ Méthode des trapèzes
2. Newton-Cotes d'ordre 2 $\iff$ Méthode Simpson
3. On montre que, selon la parité de n :
   n **pair** $\Rightarrow$ Résultat exact dans $\mathbb{R}_{n+1}[X]$, n **impair** $\Rightarrow$ Résultat exact dans $\mathbb{R}_n[X]$
   Ainsi la méthode de **Simpson** (i.e. $n = 2$) est **exacte pour des polynômes de degré** 3

Exemple : pour $n = 4$, i.e. 4 points d'interpolations et donc 3 subdivisions (avec $h = x_{i+1} - x_i$ et $x_i < c < x_{i+3}$) :

$$\int_{x_i}^{x_{i+3}} f(t)\,dt = \frac{3}{8}h\left[f(x_i) + 3f(x_{i+1}) + 3f(x_{i+2}) + f(x_{i+3})\right] + \underbrace{E(h)}_{=-3h^5.f^{(4)}(c)/80} \quad \text{avec } E(h) \leqslant M.h^5$$

Comparaison des méthodes

```
##
def f(x):
    # x ↦ πx sin(2πx)
    return np.pi*x*np.sin(2*np.pi*x)
#---------------------------------
# fonction definies a la volee :
#---------------------------------
fc=lambda x:1          # ( x ↦ 1)
fa=lambda x:x          # ( x ↦ x)
fq=lambda x:x**2       # ( x ↦ x^2)
fcub=lambda x:x**3     # ( x ↦ x^3)
ft=lambda x:x**(-0.5)# ( x ↦ 1/√x)
#
```

```
a,b=0,1      # intervalle d'int{\'e}gration [0,1]
# Valeur exacte ∫_0^1 1 dx = 1
cl=LeftBox(fc,a,b,100)
cr=RightBox(fc,a,b,100)
cm=MiddleBox(fc,a,b,100)
ct=Trapeze(fc,a,b,100)
cs=Simpson(fc,a,b,50)
cq=spi.quad(fc,a,b)
# commande de Python
print("----␣fonction␣CONSTANTE␣----")
print("LeftBox=%5.5f,␣RightBox=%5.5f"%(cl,cr));
print("Middle=%5.5f,␣Trapeze=%5.5f"%(cm,ct))
print("Simpson=%5.5f,␣Quad=%5.5f"%(cs,cq[0]))
```

Résultats

On considère $n = 100$ subdivisions de l'intervalle $[0, 1]$
On a utilisé la commande `quad(f,a,b)` du module `scipy.integrate`

Intégrales	Rect. Gauches	Rect. Droits	Rect. Mil.	Trapèzes	Simpson	Python quad	Val. Exact.
$\int_0^1 1\,dx$	1.000	1.000	1.000	1.000	1.000	1.000	1
$\int_0^1 x\,dx$	0.49500	0.50500	0.5000	0.5000	0.5000	0.5000	$1/2$
$\int_0^1 x^2\,dx$	0.32835	0.33835	0.333325	0.33335	0.3333	0.3333	$1/3$
$\int_0^1 x^3\,dx$	0.24502	0.25502	0.24998	0.250025	0.2500	0.2500	$1/4$
$\int_0^1 \pi x \sin(2\pi x)\,dx$	-0.49984	-0.49984	-0.50009	-0.49984	-0.50001	-0.500	$-1/2$
$\int_0^1 \frac{1}{\sqrt{x}}\,dx$	2.84757	1.85767	1.93715	2.35262	2.20601	2.000	2

Erreurs d'intégration : Tableau comparatif

Définition : On dit que $f \underset{a}{=} O(g)$ (ou $\underset{a}{O}$), f est "*grand Oh*" de g au voisinage de a, ssi il existe $K > 0$ tel que

$$\exists V \in \mathcal{V}(a), \forall x \in V : \ |f(x)| \leqslant K|g(x)|$$

Notons n le nombre de subdivisions de $[a; b]$ (donc $h = \dfrac{b-a}{n}$)

Méthode	Erreur
Rectangles Gauches (ou Droits)	$\underset{+\infty}{O}\left(\frac{1}{n}\right)$
Rectangles Milieux	$\underset{+\infty}{O}\left(\frac{1}{n^2}\right)$
Trapèzes	$\underset{+\infty}{O}\left(\frac{1}{n^2}\right)$
Simpson	$\underset{+\infty}{O}\left(\frac{1}{n^4}\right)$

L'instruction `quad` du module `scipy.integrate`

Syntaxe : forme simplifiée :

```
from scipy.integrate import quad
s=quad(func, a, b)
```

func : Fonction à intégrer
a : borne inférieur de l'intégrale
b : borne supérieur de l'intégrale

Elle renvoie `s[0]` valeur approchée l'intégrale et `s[1]` : une estimation de l'erreur
Pour une syntaxe plus complète, faire `help(quad)`
Exemples :

```
1 from scipy.integrate import quad
2 f=lambda x: x**2 # f : x ↦ x^2
3 s,err=quad(f,0,6)# s = ∫_0^6 x^2 dx
4 >>> print(s)
5 72.00000000000001# val. exacte 72
6 >>> print(err)
7 7.993605777301129e-13
```

```
1 # definition de la fonction a la volee
2 ff=lambda x:np.exp(-x**2)# i.e. x ↦ e^{-x^2/2}
3 # utilisation de quad
4 s,err=quad(ff,-10,10)# ≈ ∫_{-∞}^{∞} e^{-x^2} dx
5 >>> s # val theorique √π
6 1.772453850905516
7 >>> err # erreur
8 3.696676120408319e-13
9 >>> (np.pi)**(0.5)# √π
10 1.7724538509055159
```

Exemple : Période du pendule simple

Déterminer l'expression de la période d'un pendule simple en fonction de θ_0 l'angle initial.

Solution :

$$\frac{1}{2}mv^2 - mgh = 0 - mgh_0 = \text{const (conservation de l'énergie)} \xrightarrow[h=L\cos(\theta)]{v=L\theta'}$$

$$\frac{1}{2}mL^2\theta'^2 - mgL\cos(\theta) = 0 - mgL\cos(\theta_0) \Rightarrow \theta'^2 = \frac{2g}{L}(\cos(\theta) - \cos(\theta_0))$$

Par radical : $\dfrac{-\theta'}{\sqrt{\cos(\theta) - \cos(\theta_0)}} = \sqrt{\dfrac{2g}{L}}$ Attention au signe $(-)$: variations θ et θ' oppposées

Intégration de $t = 0$ à $t = T/4$ avec $\theta(0) = \theta_0$ et $\theta(T/4) = 0 \Rightarrow$

$$T = 4\sqrt{\frac{L}{2g}} \int_0^{\theta_0} \frac{d\theta}{\sqrt{\cos(\theta) - \cos(\theta_0)}} \quad \text{Angle moitié : } \cos(\theta) = 1 - 2\sin^2\left(\frac{\theta}{2}\right)$$

$\Rightarrow$

$$T = 2\sqrt{\frac{L}{g}} \int_0^{\theta_0} \frac{d\theta}{\sqrt{\sin^2(\frac{\theta_0}{2}) - \sin^2(\frac{\theta}{2})}} \xrightarrow[chang.\,var]{\sin(u)=\frac{\sin(\theta/2)}{\sin(\theta_0/2)}} T = 4\sqrt{\frac{L}{g}} \int_0^{\frac{\pi}{2}} \frac{du}{\sqrt{1 - \sin^2(\frac{\theta_0}{2})\sin^2(u)}}$$

$$T = 4\sqrt{\frac{L}{g}} \cdot K(\sin(\frac{\theta_0}{2})) \text{ où } K(x) = \int_0^{\frac{\pi}{2}} \frac{du}{\sqrt{1 - x^2\sin^2(u)}}$$

La fonction $x \mapsto K(x) = \int_0^{\frac{\pi}{2}} \frac{du}{\sqrt{1-x^2\sin^2(u)}}$, s'appelle fonction **intégrale elliptique**, elle ne peut s'exprimer à l'aide des fonctions usuelles(sinon on l'aurait su).

$$T = 4\sqrt{\frac{L}{g}} \int_0^{\frac{\pi}{2}} \frac{du}{\sqrt{1-\sin^2(\frac{\theta_0}{2})\sin^2(u)}} = \frac{2}{\pi} T_0.K(\sin(\theta_0/2))$$

Quelques approximations :

(•) $\theta_0 \approx 0 \Rightarrow$ $T \approx T_0 = 2\pi\sqrt{\frac{L}{g}}$ (formule connue)

(•) θ_0 petit $\xrightarrow[(1+u)^\alpha \approx 1+\alpha u]{DL}$ $(1-\sin^2(\theta_0/2)\sin^2(u))^{-\frac{1}{2}} \approx 1 + \frac{1}{2}\underbrace{\sin^2(\theta_0/2)}_{\frac{1}{2}\theta_0{}^2}\sin^2(u)$

$\xrightarrow[\text{avec } \int_0^{\pi/2}\sin^2(u)\,du = \frac{\pi}{4}]{\text{En intégrant}}$ $T \approx T_0(1+\frac{1}{16}\theta_0^2)$ [Formule de Bordas]

Implémentation en Python et comparaison

```
#
import numpy as np
import scipy.integrate as spi
import matplotlib.pyplot as mp
#-----------------------------
def fellip(x,u):#x ↦ 1/√(1−u².sin²(x))
    return 1/np.sqrt( 1-u**2*(np.sin(x))**2)
#
t=np.linspace(0.01,1.57,20)
nt=len(t)
y=[]
yellip=[]
for tg in t:
    s=2*spi.quad(f,0,tg,args=(tg,))[0]/np.sqrt(2)/np.pi
    u=np.sin(tg/2)
    sellip=2*spi.quad(fellip,0,np.pi/2,args=(u,))[0]/np.pi
    y.append(s)
    yellip.append(sellip)
#
fig=mp.figure()
mp.draw()
mp.axis([0.,1.7,0.8,1.3])
mp.plot(t,yellip,'g',t,(1+t**2/16),'or')
mp.title("Comparaison")
mp.xlabel("Angle␣initial")
mp.ylabel("Période␣T")
mp.legend(["Exact","approch."])
mp.grid()
mp.show()
```

Illustration graphique

Courbe de $T(\theta_0) = \frac{2}{\pi} T_0 K(\sin(\theta_0/2))$ et $T_a(\theta_0) = T_0(1 + \frac{\theta_0^2}{16})$ (en pointillé)