Méthodes numériques

Dérivation et intégration

Objectif : voir comment dériver et intégrer numériquement.

I – Dérivation

Soit une fonction f continue et dérivable. Informatiquement, elle est représentée par une liste finie de valeurs $[f(x_0),\ f(x_1),\ ...,\ f(x_N))]$. Par définition, un nombre dérivé est la limite du taux d'accroissement. Cette limite peut s'écrire de plusieurs façons :

$$f'(x) = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon) - f(x)}{\varepsilon} = \lim_{\varepsilon \to 0} \frac{f(x) - f(x-\varepsilon)}{\varepsilon} = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon) - f(x-\varepsilon)}{2\varepsilon}$$

Ces trois expressions ont la même valeur grâce à la limite mathématique. Mais informatiquement, il est impossible de faire tendre l'écart entre deux points vers 0. Cet écart est au minimum égal à une valeur appelée le pas, qui est traditionnellement noté h et qui est la distance entre deux points :

$$h = x_{i+1} - x_i.$$

Les trois expressions de la limite mathématique peuvent donc chacune être approchée par un quotient.

Limite à droite :

$$f'(x_i) = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon) - f(x)}{\varepsilon} \approx \frac{f(x_{i+1}) - f(x_i)}{h}$$

Ceci consiste à approximer la pente autour de x_i par la pente moyenne entre x_i et x_{i+1}.

Limite à gauche :

$$f'(x_i) = \lim_{\varepsilon \to 0} \frac{f(x) - f(x-\varepsilon)}{\varepsilon} \approx \frac{f(x_i) - f(x_{i-1})}{h}$$

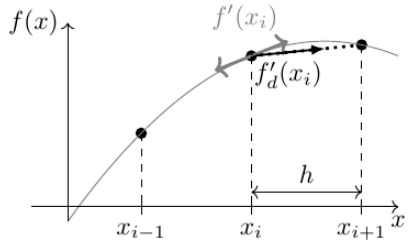
Ceci consiste à approximer la pente autour de x_i par la pente moyenne entre x_i et x_{i-1}.

Limite moyenne :

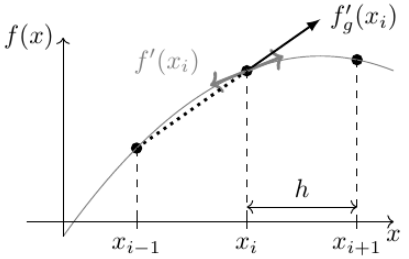
$$f'(x_i) = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon) - f(x-\varepsilon)}{2\varepsilon} \approx \frac{f(x_{i+1}) - f(x_{i-1})}{2h}$$

Ceci consiste à approximer la pente autour de x_i par la pente moyenne entre x_{i+1} et x_{i-1}.

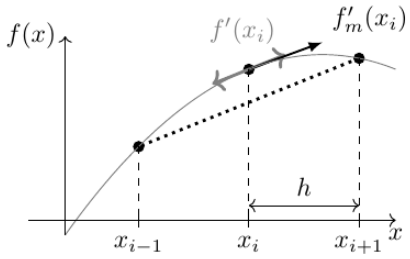
Dans tous les cas, il est nécessaire que le pas h soit très petit devant la longueur typique des variations de la fonction f.



(a) La limite à droite consiste à estimer la tangente par la flèche noire à l'aide de la droite moyenne en pointillé.



(b) La limite à gauche consiste à estimer la tangente par la flèche noire à l'aide de la droite moyenne en pointillé.



(c) La limite moyenne consiste à estimer la tangente par la flèche noire à l'aide de la droite moyenne en pointillé.

Exercice

On cherche à calculer des fonctions dérivées et à tracer ces fonctions. Pour cela, on utilise les bibliothèques numpy et matplotlib.

1 - Écrire trois fonctions `derivee_plus(y,x)`, `derivee_moins(y,x)` et `derivee_moy(y,x)`, avec en entrée `y` et `x` deux listes représentant `y = f(x)` avec f une fonction quelconque (donc `x` est la liste des x_i et `y` la liste des $f(x_i)$), et qui retournent une liste correspondant à la dérivée numérique de f.

Pour tester les fonctions de dérivation, on utilise la fonction $f(x) = sin(kx)$ avec $k = 2\pi$.

2 - Définir la fonction f. Construire une liste `x` qui contient $N = 300$ points allant de $x = -2$ à $x = 2$. Puis construire la liste `y` des valeurs de $f(x_i)$ correspondantes.

3 - Tracer sous python la dérivée analytique (théorique) attendue de cette fonction.

Tracer ensuite sur le même graphique les trois courbes correspondantes aux différentes façon d'évaluer la dérivée. Vérifier si ceci fonctionne bien.

Les signaux réels sont toujours bruités. Cela signifie que pour chaque point de mesure, une erreur aléatoire est présente. Pour simuler ce phénomène, on utilise la bibliothèque `random`.

4 - Copier les lignes de codes suivantes dans un script puis l'exécuter pour différentes valeurs de a :

```
x = np.linspace(-2, 2, 300)
a = 0.1
y_bruit = np.sin(k*x) + a*np.random.rand(len(x))
plt.figure(1)
plt.plot(x, y_bruit, 'r', label='signal bruité')
plt.show()
```

5 - Calculer et tracer la dérivée numérique de `y_bruit` à l'aide de la fonction `derivee_moy` définie dans la partie précédente.

6 - Pour amoindrir le problème des fluctuations aléatoires dues au bruit, écrire une fonction `lissage(y,k)` qui prend en entrée une liste `y` et un entier `k` et qui renvoie en sortie une liste dont l'élément `i` correspond à la moyenne glissante de `y` entre les éléments `i-k` et `i+k`.

On laissera inchangé les points au début et à la fin du signal, pour lesquels il y a des problèmes d'indice.

On indique que `range(m,n+1)` va de m à n.

Visualiser sur un graphique l'effet du lissage pour `k = 3` et `k = 5`.

7 - Calculer et tracer la dérivée numérique de `lissage(y_bruit,k)` pour `k = 5` à l'aide de la fonction `derivee_moy`. Refaire le tracé pour plusieurs valeurs de l'amplitude du bruit `a`.

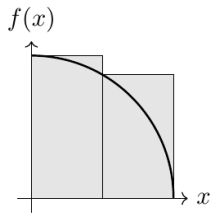
II – Intégration

Soit une fonction f continue et intégrable sur un segment [a,b]. Informatiquement, elle est représentée par une liste finie de valeurs aux points [$a = x_0$, x_1, ..., $b = x_N$].
La formule de Riemann donne une définition mathématique pour l'intégrale sur le segment $[a, b]$ grâce à la relation

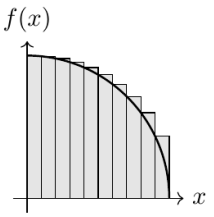
$$\int_a^b f(x)\mathrm{d}x = \lim_{N\to+\infty} \frac{b-a}{N} \sum_{i=0}^{N-1} f(t_i)$$

avec t_i compris entre x_i et x_{i+1}.
Cette formule permet de construire l'intégrale comme une somme de rectangles de largeur $h = x_{i+1} - x_i = \dfrac{b-a}{N}$ et de hauteur $f(t_i)$. Le nombre h est appelé "pas" de l'intégrale et doit être choisi le plus petit possible. Autrement dit, le nombre N de subdivisions du segment $[a, b]$ doit être pris le plus grand possible. La façon de choisir le point t_i permet de définir plusieurs méthodes d'approximation de l'intégrale.

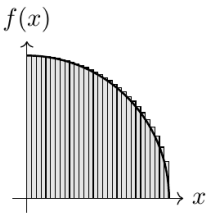
Méthode des rectangles à droite : on prend $t_i = x_i$.



(a) 2 subdivisions, l'aire grisée vaut environ 0.933 012.

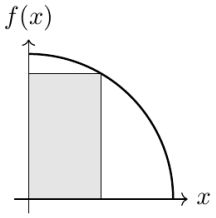


(b) 10 subdivisions, l'aire grisée vaut environ 0.826 129.

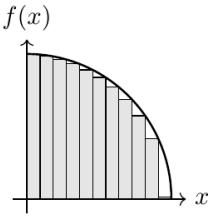


(c) 30 subdivisions, l'aire grisée vaut environ 0.800 277.

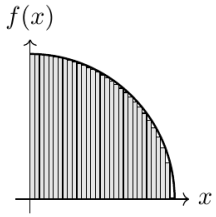
Méthode des rectangles à droite : on prend $t_i = x_{i+1}$.



(a) 2 subdivisions, l'aire grisée vaut environ 0.433 012.

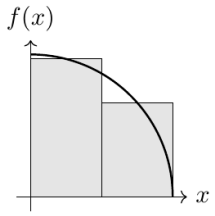


(b) 10 subdivisions, l'aire grisée vaut environ 0.726 129.

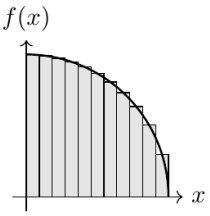


(c) 30 subdivisions, l'aire grisée vaut environ 0.766 944.

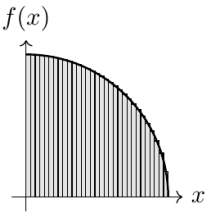
Méthode des rectangles à droite : on prend $t_i = (x_i + x_{i+1})/2$.



(a) 2 subdivisions, l'aire grisée vaut environ 0.814 841.



(b) 10 subdivisions, l'aire grisée vaut environ 0.788 102.



(c) 30 subdivisions, l'aire grisée vaut environ 0.785 921.

On peut prouver que cette dernière méthode converge plus rapidement que les deux premières (erreur en $1/N^2$ au lieu de $1/N$).

Fonction python prédéfinie :
Il existe également une fonction prédéfinie qui s'utilise ainsi :

```
import scipy.integrate as sp
import numpy as np

# un exemple de fonction
def f(x) :
    return np.exp(-x)

# entre 0 et 1 :
print ( sp.quad(f, 0, 1) )  # retourne deux valeurs : celle de l'intégrale,
# entre 0 et +infini :              et une estimation de l'erreur commise.
print ( sp.quad(f, 0, np.inf) )
```

Exercice

Nous avions établi que le temps de chute d'un arbre s'écrit $T = T_0 \int_0^{\theta_0} \frac{\mathrm{d}\theta}{\sqrt{\sin\theta_0 - \sin\theta}}$ (TD mécanique) avec $T_0 = \sqrt{\frac{L}{3g}}$, L la hauteur de l'arbre, g la pesanteur, et θ_0 l'angle initial à partir duquel l'arbre chute. Il faut calculer numériquement l'intégrale. Pour $\theta_0 = 0{,}9 \times \pi/2$, nous avions donné $\displaystyle\int_0^{\theta_0} \frac{\mathrm{d}\theta}{\sqrt{\sin\theta_0 - \sin\theta}} = 4{,}315$. Voyons comment retrouver ceci.

1 - Définir la fonction `f(theta)` qui est dans l'intégrale ci-dessus. On suppose θ_0 défini en dehors de la fonction.

2 - Écrire une fonction `int_rect(N)` qui retourne la valeur approchée de l'intégrale de la fonction f entre 0 et θ_0, avec la méthode des rectangles (méthode à droite par exemple), en utilisant N points (donc le pas est $h = (\theta_0 - 0)/N$).

3 - Tester votre fonction pour voir si vous retrouver la valeur 4,315. Tester avec différentes valeurs de N.

4 - Faire de même, mais cette fois en utilisant la fonction `quad` de la bibliothèque `scipy`. Comparer.

5 - Vous pouvez aussi coder les deux autres méthodes des rectangles pour voir la différence.