BCPST 1 | **Corrigé du devoir d'infomatique** | 27-05-2023

Lycée Fénelon | **Intégration numérique** | Informatique

Méthode de Simpson

1. On regarde $f : x \mapsto x^4$, $a = -1$, $b = 1$. On a alors

$$\begin{aligned} I_{a,b}(f) &= \frac{2}{5} \\ I_{a,b}^{(\text{Simpson})}(f) &= \frac{2}{3} \\ &\neq I_{a,b}(f) \end{aligned}$$

2. Soit $f : x \mapsto ux^2 + vx + w$. On calcule

$$\begin{aligned} I_{a,b}^{(\text{Simpson})}(f) &= \frac{b-a}{6}\left(u(b^2 + a^2 + (b+a)^2) + v(b + a + 2(b+a)) + 6w\right) \\ I_{a,b}(f) &= u\frac{b^3 - a^3}{3} + v\frac{b^2 - a^2}{2} + w(b-a) \\ &= (b-a)\left(\frac{u}{3}(b^2 - ab + a^2) + \frac{v}{2}(b+a) + w\right) \\ &= \frac{b-a}{6}\left(u(b^2 + b^2 - 2ab + a^2 + a^2) + v(3b + 3a) + 6w\right) \\ &= I_{a,b}^{(\text{Simpson})}(f) \end{aligned}$$

ce qui conclut.

3. On calcule à nouveau

$$\begin{aligned} I_{a,b}^{(\text{Simpson})}(P_{a,b}) &= \frac{b-a}{6}((\frac{a+b}{2})^3 + (-\frac{a+b}{2})^3) \\ &= 0 \\ I_{a,b}(f) &= \int_a^b (t - \frac{a+b}{2})dt \\ &= \int_{-\frac{a+b}{2}}^{\frac{a+b}{2}} t^3 dt \\ &= 0 \end{aligned}$$

4. On remarque que la méthode de Simpson est linéaire, i.e.

$$I_{a,b}^{(\text{Simpson})}(\alpha f + g) = \alpha I_{a,b}^{(\text{Simpson})}(f) + I_{a,b}^{(\text{Simpson})}(g)$$

Ainsi soit $P \in \mathbb{R}_3[X]$, $a, b \in \mathbb{R}$. On écrit $P = \alpha P_{a,b} + Q$ grâce à la question précédente. On a alors

$$\begin{aligned} I_{a,b}^{(\text{Simpson})}(P) &= \alpha I_{a,b}^{(\text{Simpson})}(P_{a,b}) + I_{a,b}^{(\text{Simpson})}(Q) \\ &= \alpha I_{a,b}(P_{a,b}) + I_{a,b}^{(\text{Simpson})}(Q) \quad \text{grâce à la question 3.a} \\ &= \alpha I_{a,b}(P_{a,b}) + I_{a,b}(Q) \quad \text{grâce à la question 2} \\ &= I_{a,b}(P) \quad \text{par linéarité de l'intégrale} \end{aligned}$$

ce qui conclut.

5. a)

```python
def simpson(f, a, b):
return (b-a)/2 * (f(a) + f(b) + f((a+b)/2))
```

b)

```python
def simpson_composite(f, a, b, n):
    x = np.linspace(a, b, n)
    tot = 0
    for k in range(n-1):
        tot += simpson(f, x[k], x[k+1])
    return tot
```

6. La méthode des rectangles est d'ordre 0. Il faut lui préférer la méthode de Simpson : l'erreur de la généralisation composite de la méthode des rectangles décroit en $O(\frac{1}{n})$ tandis que celle de la généralisation composite de la méthode de Simpson décroit en $O(\frac{1}{n^4})$. Remarquez qu'on ne demande pas de se souvenir de la puissance de décroissance exacte, il est suffisant de mentionner qu'expérimentalement, l'erreur de Simpson décroit plus rapidement avec n que celle des rectangles.

Gymnastique Python

7. a)

```python
def trouve_minimum(L):
    mini = L[0]
    for x in L:
        if x < mini:
            mini = x
    return mini
```

b)

```python
def trouve_indice_du_minimum(L):
    mini, argmini = L[0], 0
    for i, x in enumerate(L):
        if x < mini:
            mini = x
            argmini = i
    return argmini
```

Si on ne connait pas la fonction `enumerate`, on peut également écrire

```python
def trouve_indice_du_minimum(L):
    mini, argmini = L[0], 0
    for i in range(len(L)):
        x = L[i]
        if x < mini:
            mini = x
            argmini = i
    return argmini
```

8.

```python
def sous_liste_pair(L):
    ret = []
    for x in L:
        if x%2 == 0:
            ret.append(x)
    return ret
```

9.

```python
def applati(L):
    ret = []
    for sous_liste in L:
        for x in sous_liste:
            ret.append(x)
    return ret
```

ou encore

```
def applati(L):
    ret = []
    for x in L:
        ret += x
    return ret
```

10. `L.append(a)`

11. a) `True`

b) `True`

c) `False`

12. a) `[3]`

b) `"cpst"`

c)

```
2
4
```