Contrôle d'informatique

Durée : 1 heure

Exercice 1 On considère une fonction $f : \mathbb{R} \to \mathbb{R}$ de classe $\mathscr{C}^3$ ainsi que deux réels $a \in \mathbb{R}$ et $h > 0$. On rappelle (ou on admet) que la fonction $\mathrm{F} : x \mapsto \int_a^x f(t)\,\mathrm{d}t$ est une application de classe $\mathscr{C}^4$ telle que $\forall x \in \mathbb{R}$, $\mathrm{F}'(x) = f(x)$.

a) Déterminez les valeurs à choisir pour les coefficients α et β de telle manière que la formule :

$$\int_a^{a+h} f(t)\,\mathrm{d}t \approx \alpha f(a) + \beta f(a+h) \tag{1}$$

soit exacte pour des polynômes de degré le plus haut possible. On suppose désormais α et β ainsi choisis.

b) On note $\mathrm{I} = \int_a^{a+h} f(t)\,\mathrm{d}t$, $\mathrm{I}(h) = \alpha f(a) + \beta f(a+h)$ et $\mathrm{R}(h) = \mathrm{I} - \mathrm{I}(h)$.
En appliquant la formule de Taylor-Young à la fonction R, prouver que :

$$\mathrm{R}(h) = -\frac{h^3}{12} f''(a) - \frac{h^4}{24} f^{(3)}(a) + \mathrm{o}(h^4).$$

c) En utilisant la formule d'intégration (1), donner une approximation de

$$\mathrm{I}_1 = \int_a^{a+h/2} f(t)\,\mathrm{d}t \qquad \text{et} \qquad \mathrm{I}_2 = \int_{a+h/2}^{a+h} f(t)\,\mathrm{d}t$$

puis en écrivant que $\mathrm{I} = \mathrm{I}_1 + \mathrm{I}_2$, donner une approximation de I notée $\mathrm{J}(h)$.

d) On définit $r(h) = \mathrm{I} - \mathrm{J}(h)$. Par un calcul analogue à celui de la question b. *et qu'on ne demande pas d'effectuer*, on obtient :

$$r(h) = -\frac{h^3}{48} f''(a) - \frac{h^4}{96} f^{(3)}(a) + \mathrm{o}(h^4)$$

En déduire des coefficients constants λ et μ tels que :

$$\lambda \mathrm{J}(h) + \mu \mathrm{I}(h) = \mathrm{I} + ch^4 + \mathrm{o}(h^4)$$

où c est une constante à déterminer.

e) À quelle méthode d'intégration numérique correspond l'approximation de I par $\lambda \mathrm{J}(h) + \mu \mathrm{I}(h)$?

Exercice 2 On considère un polynôme p à coefficients réels de degré inférieur ou égal à n :

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0$$

représenté en machine par le tableau de ces coefficients : $[a_0, a_1, \ldots, a_{n-1}, a_n]$.

a) À tout réel α on associe la suite finie $(b_0, b_1, \ldots, b_n)$ définie par les relations :

$$b_n = a_n \qquad \text{et} \qquad \forall i \in [\![0, n-1]\!],\ b_i = a_i + \alpha b_{i+1} \tag{1}$$

et on note p_1 le polynôme $p_1(x) = b_n x^{n-1} + b_{n-1} x^{n-2} + \cdots, b_2 x + b_1$.
Montrer que $p(x) = (x - \alpha) p_1(x) + b_0$.

De la formule précédente il résulte immédiatement que $b_0 = p(\alpha)$. On appelle *schéma de* Hörner l'algorithme de calcul de $p(\alpha)$ à l'aide des relations (1). Cet algorithme est souvent utilisé pour évaluer $p(\alpha)$ car il provoque moins d'erreurs numériques qu'une démarche naïve.

b) Exprimer $p'(\alpha)$ en fonction de p_1 et de α et en déduire à l'aide du schéma de Hörner une suite finie $(c_1, \ldots, c_n)$ telle que $p'(\alpha) = c_1$.

c) L'algorithme de Newton-Hörner est une méthode d'approximation d'une racine d'un polynôme p obtenue en appliquant la méthode de Newton-Raphson à p mais en évaluant p et p' par le biais du schéma de Hörner.
Rédiger une fonction Python qui prend en arguments un polynôme p *représenté par la liste de ces coefficients* $[a_0, a_1, \ldots, a_n]$ et une valeur initiale x_0 et qui retourne la première itérée x_1 de la méthode de Newton (en supposant qu'elle existe).

♦ ♦
♦