Concours blanc

Informatique pour tous, première année

Julien Reichert

Cours

On rappelle que la variance d'une liste est la moyenne des carrés des écarts à sa moyenne, donc en notant $x_1, \dots, x_n$ les éléments et $\bar{x}$ la moyenne, la formule est $\frac{\sum_{i=1}^{n}(x_1-\bar{x})^2}{n}$.

Question C-1 : Écrire en Python une fonction `variance(L)` qui calcule la variance d'une liste `L`. Quelle est la complexité de cette fonction ?

Question C-2 : Écrire en Python une fonction `nbocc(x, L)` qui calcule le nombre de fois qu'un élément `x` est dans une liste `L`.

Exercice

Question E-1 : Écrire en Python une fonction `nbinter(L, LL)` qui calcule le nombre d'égalités entre un élément de la liste `L` et un élément de la liste `LL` ; par exemple, si une valeur apparaît deux fois dans une des listes et trois fois dans l'autre, cela devra compter pour six égalités. Quelle est la complexité de cette fonction ?

On définit la moyenne arithmético-géométrique de deux nombres x et y, avec $0 \leq y \leq x$, comme la limite des deux suites adjacentes (si on a $x < y$, ce sont les suites extraites en ignorant le premier terme de chacune qui sont adjacentes) $(x_n)_{n\in\mathbb{N}}$ et $(y_n)_{n\in\mathbb{N}}$, vérifiant $x_0 = x$, $y_0 = y$ et pour tout entier n les relations de récurrence $x_{n+1} = \frac{x_n+y_n}{2}$ et $y_{n+1} = \sqrt{x_n y_n}$.

Question E-2 : Écrire en Python une fonction `moyag(a, b, epsilon)` qui calcule une approximation à `epsilon` près de la moyenne arithmético-géométrique entre deux nombres `a` et `b`, c'est-à-dire que la fonction doit retourner un flottant `x` tel qu'en notant `m` la moyenne arithmético-géométrique de `a` et `b` obtenue mathématiquement, on ait un écart d'au plus `epsilon` entre `m` et `x`. Prouver la terminaison de la fonction.

Problème

Nous allons étudier ici une variante de la méthode d'Euler (explicite) avec un pas adaptatif. La subdivision des abscisses ne sera donc pas fournie ni calculée en début de fonction avec `linspace` ou `arange` mais construite au fur et à mesure.

Dans un premier temps, pour fixer les idées, voyons ce que cela donne sur la méthode des rectangles avec une méthode naïve d'adaptation du pas.

Soit une fonction f dont on cherche à **approximer l'intégrale** entre a et b (les nombres sont supposés donnés dans l'ordre). On fournit également un pas initial h.

Chaque étape de l'algorithme proposé se décompose alors ainsi :
- On démarre l'étape à l'abscisse x.
- On avance du pas h actuel (si on atteint ou dépasse b, on s'y place et on arrête le travail à la fin de cette étape).
- On calcule $f(x+h)$.
- On traite le rectangle ainsi formé (entre les abscisses x et $x+h$) comme dans la méthode des rectangles de base (aucune indication ici).
- On compare le taux d'accroissement f_p entre le point précédent et le nouveau point, c'est-à-dire $\frac{f(x+h)-f(h)}{h}$, au taux d'accroissement global D_f entre a et b, c'est-à-dire $\frac{f(b)-f(a)}{b-a}$: si f_p est supérieur à D_f, alors on réduit le pas de dix pour cent de sa valeur actuelle, s'il est inférieur alors on l'augmente de dix pour cent de sa valeur actuelle, et dans le cas improbable où il serait égal, on maintient le pas.

Question 1 : Écrire l'algorithme présenté ci-avant pour approximer une intégrale en Python.

Parlons à présent de la méthode d'Euler-Richardson, faisant une adaptation efficace du pas.

On se donne une fonction $F(y,t)$ décrivant une équation différentielle du premier ordre dont la solution est une fonction $y(t)$. On cherche une approximation de $y(x)$ pour x prenant un certain nombre de valeurs entre deux nombres a et b donnés, sachant qu'on fixe l'image de a à un certain $y0$. On donne ici aussi un pas initial h et on fournit également un seuil de précision ε. La méthode d'Euler explicite revient à calculer l'image du point x_{i+1} à partir du point précédent x_i et de son image, par la formule $y(x_{i+1}) = y(x_i) + (x_{i+1} - x_i)F(y(x_i), x_i)$, ceci étant répété pour tous les points. La méthode d'Euler-Richardson ajoute des opérations à chaque étape (en s'arrêtant bien entendu une fois le point `b` atteint), à savoir :
- Soit x l'abscisse actuelle, soit h le pas actuel. On a précédemment calculé et mémorisé $y(x)$ (en l'occurrence une valeur approchée de $y(x)$).
- Calculer la valeur $F(y(x), x)$, notée y_p, en tant que première approximation de $y'(x)$, comme dans la méthode d'Euler explicite.
- Calculer la valeur attendue après un demi-pas, à savoir $F(y(x) + y_p\frac{h}{2}, x + \frac{h}{2})$, notée y_{p2}.
- Calculer l'écart après un demi-pas, à savoir $\frac{h}{2}|y_{p2} - y_p|$, noté η.
- Soit α le rapport $\frac{\eta}{\epsilon}$. Si $\alpha < 1$, un nouveau point est validé en $x+h$, et on considère $y(x) + h * y_{p2}$ comme l'approximation de son image par la fonction solution, sinon on reste au point actuel ; dans tous les cas, on continue avec un nouveau pas de $0,9\frac{h}{\sqrt{\alpha}}$.

Question 2 : Écrire l'algorithme d'Euler-Richardson en Python.