Séries et graphes

On trouvera sur le cahier de prépa de la classe la fiche des commandes usuelles pour l'affichage des graphes produites par le concours agro-veto.

Exercice 1 (★). 1. Écrire une fonction `SerieHarmonique(n)`, qui à $n \in \mathbb{N}^*$, renvoie la somme partielle d'indice n de la série harmonique :

$$H_n = \sum_{k=1}^n \frac{1}{k}$$

2. Écrire une fonction `Zeta2(n)`, qui à $n \in \mathbb{N}^*$, renvoie la somme partielle suivante : $S_n = \sum_{k=1}^n \frac{1}{k^2}$

3. À l'aide des fonctions précédentes, écrire une fonction `graphe` prenant en entrée un entier p et qui renvoie les nuages de points correspondants aux p premiers termes des suites de H_n et de S_n . Les points de H_n seront en rouge et les points de S_n seront en bleu.

4. Écrire une fonction `GrapheEquivalent` prenant en entrée un entier p et qui renvoie le nuages de points correspondants aux p premiers termes des suites de $H_n/\ln(n)$.

5. On admet que la série $\sum \frac{1}{n^2}$ converge et que $\sum_{k=1}^{+\infty} \frac{1}{k^2} = \frac{\pi^2}{6}$. Écrire une fonction `Seuil(epsilon)` qui renvoie le premier entier n_0 tel que

$$\left| \frac{\pi^2}{6} - \sum_{k=1}^{n_0} \frac{1}{k^2} \right| \leq \varepsilon$$

6. Il y a donc p appels de la fonction `SerieHarmonique` pour afficher le graphe. Pour $i \in \llbracket 1; p \rrbracket$, le calcul de $\sum_{k=1}^i \frac{1}{k}$ est une somme i termes (donc $i-1$ additions), ainsi il y a trop d'additions au total :

$$\sum_{i=1}^p (i-1) = \frac{p(p+1)}{2} - p \underset{p \rightarrow +\infty}{\sim} \frac{p^2}{2}$$

Écrire une fonction `SerieHarmonique2(p)` qui renvoie la liste des valeurs de H_1, H_2 jusqu'à H_p en faisant en sorte qu'il y ait au total que $p-1$ additions.

Exercice 2 (★). On considère f la fonction 2π -périodique valant 1 sur $\left[-\frac{\pi}{2}; \frac{\pi}{2}\right]$, -1 sur $\left] -\pi; -\frac{\pi}{2} \right[$ et $\left] \frac{\pi}{2}; \pi \right[$. On rappelle que les coefficients de fourrier de f valent :

$$\forall n \in \mathbb{N}^* \quad \begin{aligned} a_0 &= \frac{1}{2\pi} \int_{-\pi}^{\pi} f \\ a_n &= \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos(nt) dt \\ b_n &= \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \sin(nt) dt \end{aligned}$$

- Calculer les coefficients a_0, a_n et b_n .
- Écrire une fonction `Fourrier(n,x)` qui renvoie la somme partielle de la série de Fourier :

$$a_0 + \sum_{k=1}^n a_k \cos(kx) + b_k \sin(kx)$$

- Afficher le graphe de la fonction de la somme partielle jusqu'à n sur $[-\pi; \pi]$ pour plusieurs valeurs de n ainsi que le graphe de f .

Dichotomie

Exercice 3 (★★★). 1. Écrire une fonction `EstDedans` qui dépend de deux paramètres : une liste L et un élément x . Cette fonction renverra `True` si x est un élément de la liste L et `False` sinon. Pour ce faire, pour chaque élément de la liste, vérifier si x est égal à cet élément. Par exemple, `EstDedans([1,4,3],4)` renvoie `True` tandis que `EstDedans([1,4,3],2)` renvoie `False`.

Si $L=[1,5,7,8,9]$ alors $L[2]=7$ est au milieu de la liste. Si $M=[1,5,7,8,9,11]$, alors 7 ou 8 sont à peu près au milieu de la liste. Soit une liste que l'on suppose **triée par ordre croissant** (comme L et M).

- Quelle commande utiliseriez-vous pour obtenir la plus grande/petite valeur de la liste ?
- Quelle commande utiliseriez-vous pour obtenir un élément qui est à peu près au milieu de la liste ?

On veut déterminer si un élément x est dans cette liste sans tester tous les éléments de la liste contrairement à la première question. Pour cela, si x est strictement inférieur au plus petit élément de la liste ou strictement supérieur au plus grand élément de la liste, alors x n'est pas dans la liste. Dans le cas contraire, on compare x à l'élément au milieu, si x est plus grand on va regarder la partie droite de la liste, sinon la partie gauche. Puis on continue en divisant à chaque fois la taille de l'intervalle par deux.

4. Appliquer l'algorithme à la main, avec $x=19$ et $L=[1,2,3,5,12,14,16,18,22,24,25]$. Compter le nombre de comparaisons faites à la main, et compter le nombre de comparaison que vous auriez du faire si vous aviez appliqué l'algorithme de la première question à la main.

Cet algorithme utilise la dichotomie.

5. Implémenter cet algorithme avec une fonction `Dichotomie` à deux paramètres : L et x .
6. Tester cet algorithme sur des exemples variées (liste ayant un nombre pair ou impair d'éléments, élément en début de liste, fin de liste, milieu de liste etc.).
7. Comparer le nombre de comparaisons nécessaires dans le pire des cas dans cet exercice ainsi que dans le précédent en fonction de la taille de la liste.