Informatique en CPGE (2018-2019)
TP 4 : les fonctions

Exercice 1

On considère les fonction f et g définies sur $]0;+\infty[$ respectivement par $f(x)=\dfrac{\sqrt{1+x}-1}{x}$ et $g(x)=\dfrac{1}{\sqrt{1+x}+1}$. Vérifier que $f(x)=g(x)$.

Ecrire un programme qui calcule les images par f et par g de 10^{-5}, 10^{-10}, 10^{-15} et 10^{-20}.

Expliquer les résultats puis conjecturer la limite de la fonction f en 0.

Les exercices 2 à 5 consistent à implémenter les algorithmes présentés dans le cours sur les fonctions.

Exercice 2 : recherche de solutions de l'équation $f(x)=0$ par **dichotomie**.

Ecrire une fonction `dichotomie(f,a,b,eps)` qui prend en arguments une fonction f, les bornes a et b d'un intervalle et la précision `eps`, et qui renvoie une solution approchée de l'équation $f(x)=0$ sur $[a;b]$ à `eps` près.

Exécuter le programme afin d'obtenir une solution à 10^{-3} près de l'équation $f(x)=0$ sur l'intervalle $[1;2]$ avec $f(x)=\ln x-2+x$.

Exercice 3 : recherche d'extremum

1. Implémenter l'algorithme 1 du cours (déterministe à pas constant) et exécuter le programme afin de déterminer une valeur approchée du maximum de la fonction f définie par $f(x)=x\mathrm{e}^{-x}$ sur l'intervalle $[0;2]$.

   Pour cela, écrire une fonction `maximum1(f,a,b,n)` qui prend en arguments une fonction f, les bornes a et b d'un intervalle et le nombre n de valeurs calculées, et renvoie une valeur approchée du maximum en calculant 1000 valeurs équiréparties sur $[a;b]$.
2. Implémenter l'algorithme 2 du cours (tabulation « aléatoire » d'une fonction) pour résoudre le même problème.

   Pour cela, écrire une fonction `maximum2(f,a,b,n)` qui prend en arguments une fonction f, les bornes a et b d'un intervalle et le nombre n de valeurs calculées, et renvoie une valeur approchée du maximum en calculant 1000 valeurs aléatoires sur $[a;b]$.

Exercice 4 : test de la monotonie

Implémenter l'algorithme du cours et le tester avec la fonction f définie sur l'intervalle $[0;\pi]$ par $f(x)=x+1,0001\cos x$.

Pour cela, écrire une fonction `monotonie(f,a,b,n)` qui prend en arguments une fonction f, les bornes a et b d'un intervalle et le nombre n de valeurs calculées, et affiche suivant les cas le message "la fonction n'est pas monotone" ou le message "la fonction semble monotone" .

Exécuter le programme avec $n=100$ puis avec $n=1000$.

Exercice 5 : calcul d'intégrales

Ecrire un programme permettant de déterminer une valeur approchée de l'intégrale $\int_0^1 x\mathrm{e}^{-x}\,\mathrm{d}x$ en utilisant la méthode des trapèzes.

Pour cela, écrire une fonction `trapezes(f,a,b,n)` qui prend en arguments une fonction f, les bornes a et b d'un intervalle et le nombre n de valeurs calculées, et renvoie une valeur approchée de l'intégrale de f sur $[a;b]$. Tester avec $n = 100$.

Exercice 6 : **méthode de Monte-Carlo**

Il existe de nombreuses méthodes d'approximation d'intégrales du type $\int_0^1 f(x)\,\mathrm{d}x$ par des formules du type $\sum_1^n w_i f(x_i)$. Par exemple pour la méthode des trapèzes, $w_i = \dfrac{1}{n}$ et $x_i = \dfrac{i}{n}$. Dans la méthode de Monte-Carlo on prend encore $w_i = \dfrac{1}{n}$ mais on choisit les x_i selon la loi uniforme sur $[0;1]$.

Ecrire un programme utilisant la méthode de Monte-Carlo pour approcher l'intégrale de l'exercice 5 et comparer le résultat avec celui obtenu par la méthode des trapèzes.

Pour cela, écrire une fonction `monte_carlo(f,a,b,n)` qui prend en arguments une fonction f, les bornes a et b d'un intervalle et le nombre n de valeurs calculées, et renvoie une valeur approchée de l'intégrale de f sur $[a;b]$.

Note : la convergence de la méthode de Monte-Carlo est assurée par la loi des grands nombres. La vitesse de convergence est relativement faible par rapport à d'autres méthodes en dimension 1. L'intérêt de cette méthode apparaît lorsque la dimension augmente. En dimension d les méthodes numériques classiques nécessitent d'utiliser n^d points pour garder la même précision alors que la méthode de Monte-Carlo est insensible à la dimension.