Université Paris-Sud **Travaux pratiques**
L3 Mathématiques **2012-2013**
M315 - Calcul Scientifique

Feuille de TP 3

Exercice 1. [Méthode de Newton et méthode de dichotomie]

On considère la fonction $g : \mathbb{R} \longrightarrow \mathbb{R}$ définie par $g(x) = x^5 - 2x^3 + 1.2x + 0.05$. Elle possède un unique zéro sur $\mathbb{R}$. On souhaite en préciser la valeur numérique.

1. Vérifier sur un graphique que la fonction g admet bien un unique zéro. En dressant le tableau de variation de g on constatera qu'il suffit de représenter g sur l'intervalle $[-3/2, 3/2]$, par exemple.

On s'intéresse à la méthode de Newton pour trouver un zéro c d'une fonction continue f. Cette méthode consiste à définir la suite $(x_n)_{n\in\mathbb{N}}$ par

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \;\; n \geq 0,$$

tant que $f'(x_n)$ ne s'annule pas. Si x_0 est choisi suffisamment proche de c, la suite x_n converge vers c.

2. Écrire une fonction Matlab `[N,C]=newton(f,df,x0,prec,Nmax)` qui calcule une solution approchée `C` de $\texttt{f}(x) = 0$ par la méthode de Newton. Les arguments d'entrée de la fonction `newton` sont la fonction `f`, sa dérivée `df`, le point `x_0` d'initialisation de la suite (x_n), la précision voulue `prec` et le nombre maximal d'itérations autorisées de l'algorithme `Nmax`. En sortie, `C` est la valeur approchée de la solution obtenue à la fin de `N` itérations de l'algorithme. On arrêtera l'algorithme si $|f(x_n)| \leq \texttt{prec}$ ou si $n = \texttt{Nmax}$.

Dans la suite on cherchera à calculer une solution approchée `C` de la solution de $g(x) = 0$ vérifiant $|g(\texttt{C})| \leq 10^{-15}$ et on fixera `Nmax`=100.

3. Appliquer la méthode de Newton à la fonction g en prenant pour point de départ $x_0 = 1.7$. Qu'observe-t-on ? La méthode semble-t-elle converger, si l'on considère cette valeur de x_0 ?
4. La convergence de la méthode de Newton étant locale, on va chercher à, avant l'utiliser, s'approcher de la solution, ayant comme but de prendre x_0 suffisamment proche du zéro de g. Pour ce faire, appliquer la méthode de dichotomie à partir de l'intervalle $[-1, 1.5]$ pour obtenir un intervalle de longueur inférieure à 0.5 qui contient le zéro de g. Appliquer ensuite la méthode de Newton, en prenant comme x_0 la valeur approchée du zéro de g donnée par la méthode de dichotomie avec cette précision. Afficher (en utilisant la fonction `fprintf`) les valeurs de la suite (x_n) jusqu'à la convergence, c'est-à-dire jusqu'à ce que $|g(x_n)| \leq 10^{-15}$.
5. Comparer le nombre d'itérations nécessaires pour obtenir une approximation du zéro de g avec une précision donnée (par exemple 10^{-15}) avec la méthode de dichotomie et la méthode de Newton. Pour ce faire, modifier la fonction `dicho` programmée au TP précédent pour que le critère d'arrêt des deux algorithmes soit le même. Quelles sont les avantages de chacune des méthodes par rapport à l'autre ?

Exercice 2. [Méthode de la sécante]
On considère la fonction φ définie sur $[0, 1]$ par la série alternée

$$\phi(x) = 1 + \sum_{k=1}^{+\infty} \frac{(-x)^k}{k^{3/2}}.$$

Un résultat théorique assure que le reste d'ordre n de la série (c'est-à-dire $\sum\limits_{k=n+1}^{+\infty} \frac{(-x)^k}{k^{3/2}}$) est majoré en valeur absolue par la valeur absolue $\frac{x^{n+1}}{(n+1)^{3/2}}$ de son premier terme.

1. Programmez une fonction `y=phi(x)=` qui calcule une valeur approchée de $\phi(\mathtt{x})$ en sommant seulement les termes de la série pour k allant de 1 à n où n est le premier entier tel que $\frac{x^n}{n^{3/2}} < 10^{-16}$. Tracez le graphe de la fonction ϕ sur l'intervalle $[0, 0.99]$. Graphiquement, pouvez-vous expliquer pourquoi la fonction ϕ prend une seule fois la valeur 0.8 sur cet intervalle ? On notera z l'unique nombre de $[0, 0.99]$ tel que $\phi(z) = 0.8$.

On souhaite calculer une valeur approchée de z par la méthode de la sécante. On rappelle que la méthode de la sécante pour déterminer le zéro d'une fonction f consiste à construire une suite $(x_n)_{n\in\mathbb{N}^*}$ initialisée par deux valeurs x_0, x_1 et définie par la relation de récurrence

$$x_{n+1} = \frac{x_n f(x_{n-1}) - x_{n-1} f(x_n)}{f(x_{n-1}) - f(x_n)}.$$

2. Programmez la méthode de la sécante initialisée par $x_0 = 0$ et $x_1 = 0.5$ afin de déterminer une valeur approchée de z. Vous calculerez tous les termes x_0 jusqu'à x_N où N est le premier entier tel que $|f(x_N)| < 10^{-15}$.
3. Affichez à l'aide de la commande `fprintf` les valeurs de `N`, $x_{\mathtt{N}}$ et $f(x_{\mathtt{N}})$. En prenant pour z la dernière valeur calculée par la méthode de la sécante, c'est-à-dire $z = x_N$, calculez $\frac{\log|x_n - z|}{\log|x_{n-1} - z|}$, pour n allant de 0 à $N-1$. Vers quelle valeur semble converger le rapport $\frac{\log|x_n - z|}{\log|x_{n-1} - z|}$? Faites un commentaire en lien avec l'ordre de convergence de la méthode de la sécante.

Exercice 3. [Matrices de Vandermonde]
L'objectif de cet exercice est de déterminer et de tracer le polynôme interpolateur de Lagrange en utilisant les matrices de Vandermonde.

1. Écrivez une fonction `vdm` qui a pour argument un vecteur $\mathtt{x} = (x_1, \ldots, x_n)$ et qui retourne la matrice de Vandermonde associée
$$V(x_1, \ldots, x_n) = \begin{bmatrix} 1 & x_1 & \ldots & x_1^{n-1} \\ 1 & x_2 & \ldots & x_2^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \ldots & x_n^{n-1} \end{bmatrix}.$$
2. Écrivez une fonction `y = algohorner(p,x)` qui a pour argument un vecteur $\mathtt{p} = (a_0, a_1, \ldots, a_n)$ des coefficients d'un polynôme $P = a_0 X^n + a_1 X^{n-1} + \ldots, a_{n-1} X + a_n$ et un vecteur $\mathtt{x} = (x_1, \ldots, x_k)$ et qui retourne un vecteur $\mathtt{y} = (y_1, \ldots, y_k)$ avec $y_j = P(x_j)$, $j = 1, \ldots, k$ par la méthode de Horner.
L'algorithme de Horner consiste à calculer la séquence $p_0 = a_0$, $p_{i+1} = p_i x + a_{i+1}$, pour $0 \leq i \leq n-1$. On a alors $P(x) = p_n$.
On rappelle qu'un polynôme est représenté en `Matlab` par un vecteur ligne contenant la liste des coefficients par ordre de degré décroissant. Exemple le polynôme $3x^2 + 2x + 1$ s'écrit `[3 2 1]`. Ainsi, la fonction `polyval` de `Matlab` rend le même résultat que notre fonction `algohorner`.
Vous testerez l'algorithme de Horner en effectuant le tracé sur $[0, 1]$ du polynôme $P = 2X^2 - 3X + 1$, où les valeurs seront calculées par `algohorner`.
3. Écrivez une fonction `InterpVdm` qui prend en arguments trois vecteurs $\mathtt{X} = (X_1, \ldots, X_N)$, $\mathtt{Y} = (Y_1, \ldots, Y_N)$ et $\mathtt{x} = (x_1, \ldots, x_k)$ et qui retourne un vecteur $\mathtt{y} = (y_1, \ldots, y_k)$ tel que $y_j = P(x_j)$, pour $j = 1, \ldots, k$, où P est le polynôme interpolateur de Lagrange aux points (X_n, Y_n), $1 \leq n \leq N$.
Vous calculerez le polynôme P en utilisant les matrices de Vandermonde et vous déterminerez le vecteur `y` en évaluant $P(\mathtt{x})$ par la méthode de Horner.
Indication : pour résoudre le système linéaire $AX = b$, vous utiliserez la commande `Matlab` `X1=A;`
Vous testerez cet algorithme sur l'intervalle $[1, 3]$ avec $N \in \{4, 8, 12, 16, 20\}$ points uniformément répartis pour les fonctions suivantes
$$\exp(-3(x-1.2)^2), \qquad \frac{x^2-2}{1+2x}, \qquad \frac{1}{1+(x-1.5)^2}, \quad \frac{\sin(2\pi x)}{1.1 - \sin(\pi x)}.$$

Pour chacune de ces fonctions, vous créerez deux graphiques. Sur le premier, vous tracerez la fonction et son interpolée pour les différentes valeurs de N. Sur le second, vous tracerez l'erreur, différence entre la fonction et l'interpolée.