Proposition d'exercices supplémentaires :

Détermination numérique de zéros : Méthode de Newton

Souvent il faut résoudre une équation de type $f(x) = 0$ oú $f(x)$ est une fonction assez compliquée telle que cette équation n'admet pas de solution analytique exacte, par exemple pour $f(x) = x - \exp(-x)$ ou $f(x) = x + \log(x)$ etc.

Une méthode efficace pour résoudre une telle équation numériquement avec l'ordinateur est la **méthode de Newton**. Supposons qu'on connaisse une approximation x_0 du vrai zéro x et que $f'(x_0) \neq 0$. Dans ce cas on peut approcher la fonction $f(x)$ par une fonction linéaire au voisinage de x_0 :

$$0 = f(x) = f[x_0 + (x - x_0)] \approx f(x_0) + (x - x_0)f'(x_0) \quad \Rightarrow \quad x \approx x_1 \equiv x_0 - \frac{f(x_0)}{f'(x_0)}$$

où $f'(x_0)$ est la dérivée de f en x_0. En principe x_1 obtenu par la derniere expression n'est pas encore exactement le vrai zéro x mais elle represente une bien meilleure approximation du zéro que x_0. Si $|x - x_0| \sim \varepsilon \ll 1$ on peut montrer que $|x - x_1| \sim \varepsilon^2 \ll |x - x_0|$, c.-à-d. le nombre de chiffres correctes dans x_0 double apres une seule itération. Donc l'itération de Newton devient :

$$x_{n+1} \equiv x_n - \frac{f(x_n)}{f'(x_n)} \quad , \quad n = 0, 1, 2, \ldots$$

et x_n converge (très rapidement !) vers un zéro x si x_0 est suffisamment proche à x.

1) Écrire un programme C qui implemente la méthode de Newton pour une fonction $f(x)$ quelconque. $f(x)$ et $f'(x)$ devront étre fournies par deux fonctions du programme C. Le programme doit demander d'entrer x_0 et l'itération sera arrètée quand $|x_{n+1} - x_n| < \varepsilon$, avec par exemple $\varepsilon = 10^{-14}$. Tester cette méthode pour trouver numériquement les zéros des deux fonctions $f_1(x) = x - \exp(-x)$ et $f_2(x) = x + \log(x)$.

2) Pour certaines fonctions le calcul (ou la programmation dans une fonction C) de la dérivée $f'(x)$ est très compliquée. Dans ce cas on peut remplacer la dérivée par la pente de f entre x_n et x_{n-1} qui est : $[f(x_n) - f(x_{n-1})]/(x_n - x_{n-1})$ et obtient ainsi la **Méthode de Regula Falsi** :

$$x_{n+1} \equiv x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})} \quad , \quad n = 1, 2, 3, \ldots$$

Ceci est une double récurrence avec deux valeurs initiales x_0 et x_1. Écrire un programme C qui implemente la méthode de Regula Falsi.

3) Écrire un programme qui calcule, en fonction d'une valeur initiale x_0, les zéros du polynôme d'ordre n :

$$P(x) = \sum_{l=0}^{n} a_l \, x^l$$

par la méthode de Newton. Le programme devra aussi compter le nombre des itérations et arrêter l'itération quand un nombre limite est dépassé, notamment quand le polynôme ne possède pas de zéro réel.

4) Écrire un programme qui calcule les zéros du polynóme de Legendre $P_n(x)$. Les polynómes de Legendre sont obtenus par la relation de récurrence :

$$P_0(x) = 1 \quad , \quad P_1(x) = x \quad , \quad P_{n+1} = \frac{1}{n+1}\Big((2n+1)x\,P_n(x) - n\,P_{n-1}(x)\Big)$$

où $n = 1, 2, 3, \ldots$. De cette relation on peut aussi déduire une autre relation de récurrence pour la dérivée $P_n'(x)$ qu'on peut résoudre en même temps que la récurrence de $P_n(x)$ et ainsi déterminer le rapport $P_n(x)/P_n'(x)$ nécessaire pour la méthode de Newton.

On rappelle que chaque polynôme de Legendre possède exactement n zéros simples dans l'intervalle [−1,1]. Comme valeurs initiales pour chaque zéro on conseille de prendre :

$$x_0 = \cos\left(\frac{\pi}{n+0.5}(j+0.75)\right) \quad , \quad j = 0, 1, \ldots, n-1.$$

Pour chaque valeur de j on devrait obtenir un autre zéro.

5) Si une fonction $f(x)$ possède plusieurs zéros la méthode de Newton permet en principe de déterminer tous les zéros en choisissant pour chaque zéro la bonne valeur initiale. Toute fois il peut s'avérer difficile de trouver les bonnes valeurs intiales.

Cependant pour cette situation la méthode de Newton peut être améliorée pour tenir compte des zéros qu'on a déja trouvés. Supposons qu'on connaisse déja k zéros $x^{(1)}, \ldots, x^{(k)}$. Pour trouver de nouveaux zéros différents de $x^{(1)}, \ldots, x^{(k)}$ il est judicieux d'appliquer la méthode de Newton à la fonction

$$f(x) \;\rightarrow\; \frac{f(x)}{\prod_{j=1}^{k}(x - x^{(j)})}$$

Montrer qu'avec cette modification l'itération de Newton devient :

$$x_{n+1} \equiv x_n - \frac{f(x_n)}{f'(x_n) - f(x_n)\sum_{j=1}^{k}\frac{1}{x_n - x^{(j)}}} \quad , \quad n = 0, 1, 2, \ldots$$

Au debut on pose $j = 0$ et on détermine le premier zéro par la méthode de Newton simple. Après on pose $j = 1$ et on met le premier zéro dans un tableau et on applique l'itération modifiée pour $k = 1$ pour trouver le 2ème zéro. De cette manière on augmente k jusqu'on a trouvé tous les zéros de la fonction $f(x)$. Cette variante s'appelle la **Méthode de Newton-Maehly**.

Modifier les programmes des exos 3 et 4 pour utiliser la Méthode de Newton-Maehly.

Remarque : *La partie suivante est particulièrement ambitieuse, même pour des exercices supplémentaires !*

6) La Méthode d'intégration par Gauss

Cette méthode permet de calculer les intégrales entre -1 et 1 selon la formule:

$$\int_{-1}^{1} f(x)\,dx \approx \sum_{j=1}^{n} w_j\, f(x_j)$$

où x_j sont les zéros du polynóme de Legendre $P_n(x)$ qu'on peut calculer numériquement par la méthode de Netwon (voirs les deux exos précédants) et w_j sont de coéfficients positifs donnés par l'expression:

$$w_j = \frac{2}{P_n'(x_j)^2\,(1-x_j)^2} \quad , \quad j = 1,\,\ldots,n$$

et qu'on peut aussi calculer en utilisant la récurrence combinée de $P_n(x)$ et $P_n'(x)$ (voir l'exo 4). En fait le calcul de w_j peut se faire directement après celui de x_j car dans la méthode de Newton on utilise/obtient de toute manière la value de $P_n'(x_j)$.

La formule d'intégration ci-dessous est d'une très haute précision si on augmente les valeurs de n. Programmer une fonction C qui calcule l'intégrale d'une fonction $f(x)$ (entre $x = -1$ et $x = 1$) donnée par la méthode de Gauss et en utilisant la solution à la question 4 (ou 5) pour déterminer au préalable les zéros x_j de $P_n(x)$ et les coéfficients w_j pour $j = 1,\,\ldots,n$.

7) La formule d'intégration dans l'exo 6 est limitée à des intégrales entre -1 et 1. Pour calculer des intégrales générales :

$$\int_a^b f(x)\,dx$$

utiliser la transformation $s \to x = (a+b)/2 + (b-a)\,s/2$ où $-1 \le s \le 1$ pour obtenir une formule d'intégration généralisée et programmer une fonction C pour cette formule.

8) Pour augmenter la précision d'une inégrale entre a et b on peut couper l'intégrale en plusiéurs morceaux entre $a_0 = a$ et $a_1 = a + h$, entre a_1 et $a_2 = a_1 + h$, …, avec $h = (b-a)/m$ et $m =$ nombre de morceaux. Écrire une fonction C qui applique la formule d'intégration de Gauss généralisée (selon exo 7) à chaque de ces morceaux afin d'augmenter la précision.

Pour tester les différentes formules d'intégration on peut utiliser comme exemples les intégrales connues :

$$\int_0^{\pi} \sin(x)\,dx = 2 \quad , \quad \int_0^1 \frac{1}{1+x^2}\,dx = \frac{\pi}{4} \quad , \quad \int_1^a \frac{1}{x^2}\,dx = 1 - \frac{1}{a} \quad , \quad a = 2{,}3,\ldots\ .$$