INFO-H-100 - Programmation
TP 10 - Exercices de synthèse

Juin 2009

La sécante hyperbolique d'un angle x est calculée à l'aide de la série suivante :

$$sech(x) = \Big(1 - \frac{E_2}{2!} \cdot x^2 + \frac{E_4}{4!} \cdot x^4 - \frac{E_6}{6!} \cdot x^6 + \cdots\Big)$$

C'est-à-dire :

$$sech(x) = \Big(\sum_{n \geq 0} \frac{(-1)^n \cdot E_{2n}}{(2n)!} x^{2n}\Big)$$

où E_n représente le n^{eme} nombre de Euler.

On vous demande d'écrire :

1. une fonction `double sech(double x)`, qui retourne la valeur de $sech(x)$, pour `x` *réel*. Vos calculs s'arrêteront lorsque la valeur du dernier terme ajouté est inférieure à un ε défini comme constante globale dans votre programme (par exemple $\varepsilon = 10^{-4}$), ou lorsque le nombre de termes considérés atteint une borne également définie au sein de votre programme.
2. un exemple de fonction `int main()` qui demande un nombre réel à l'utilisateur et affiche sa sécante hyperbolique, via un appel à votre fonction `sech`.

Pour répondre à cette question, vous devez faire appel à la fonction `int euler(int n)`, qui renvoie le n^{eme} nombre d'Euler, cette fonction vous est donnée, vous ne devez ni la définir, ni la déclarer.

Veillez à optimiser les calculs effectués (en particulier les calculs des termes successifs).

Juin 2004

Ecrivez une fonction *DerivKieme* qui reçoit :
– un polynôme de degré maximum MAX,
– le degré effectif n de ce polynôme, et
– un entier positif k.
Le polynome est représenté sous la forme d'un vecteur P où chaque $P[i]$ donne le coefficient d'indice i du polynôme.

La fonction *DerivKieme* calcule dans P, la dérivée $k^{ème}$ de P. Pour cela, elle doit utiliser la formule

$$P_n^{(k)} = \sum_{i=0}^{n-k} a_{i+k} \frac{(i+k)!}{i!} x^i$$

où $P_n^{(k)}$ est la dérivée $k^{ème}$ du polynome P_n.

Vous devez veiller à optimiser vos calculs et vous ne pouvez pas utiliser de vecteur suplémentaire au vecteur P lui même.

Septembre 2004

Soit un polynôme d'ordre n appelé $P(x)$ et sa dérivée $P'(x)$ dont les définitions sont les suivantes :

$$
\begin{aligned}
P(x) &= \sum_{i=0}^{n} a_i \cdot x^i \\
P'(x) &= \sum_{i=0}^{n-1} (i+1) \cdot a_{i+1} \cdot x^i
\end{aligned}
$$

On peut trouver une racine r d'un tel polynôme par la méthode de Newton. Pour ce faire, on utilise les itérations suivantes (r_k étant une approximation de la racine r du polynôme) :

$$
\begin{aligned}
r_{k+1} &= r_k - P(r_k)/P'(r_k) \\
r &= \lim_{k \to \infty} r_k
\end{aligned}
$$

Rappelons qu'en informatique, on peut représenter un polynôme $P(x)$ sous la forme d'un vecteur P où chaque élément $P[i]$ donne le coefficient d'indice i du polynôme ($i.e. P[i] = a_i$). On vous demande d'écrire une fonction `Newton()` qui renvoie la racine d'un polynôme et qui recevra en paramètres :

1. l'ordre n du polynôme
2. un vecteur $P[i]$ contenant les coefficients du polynôme $P(x)$
3. la précision ϵ à laquelle doit être calculée la racine, *i.e.* la valeur telle que $|r_{k+1} - r_k| < \epsilon$

Quelques pistes :
- Vous pouvez utiliser un vecteur de travail pour $P'(x)$.
- Il est nécessaire d'écrire une fonction qui calcule la valeur d'un polynôme qu'on donne en paramètre.
- On prendra pour valeur initiale lors de la recherche de la racine $r_0 = 0$.