TD 7 - Analyse : spline cubique d'interpolation

On considère un intervalle $[a\,;\,b]$ et $n+1$ points, x_i, régulièrement répartis dans cet intervalle :

$x_0 = a\,;\, x_n = b$ et $x_i = x_0 + ih$ où $h = \dfrac{b-a}{n}$.

On se donne ensuite une fonction f par : $\sigma_i = f(x_i)$, $0 \leq i \leq n$.

On construit la fonction spline cubique d'interpolation, *Spline*, de la manière suivante :

• sur chaque intervalle $[x_i\,;\,x_{i+1}]$, $0 \leq i \leq n-1$, $Spline(x) = p_i(x)$ où p_i est un polynôme de degré 3.

• *Spline* est un interpolant des points $(x_i,\,\sigma_i)$ pour $0 \leq i \leq n$.

Ceci se traduit par les égalités pour $0 \leq i \leq n$: $p_i(x_i) = \sigma_i$ et $p_i(x_{i+1}) = \sigma_{i+1}$.

• *Spline* est une fonction appartenant à $\mathcal{C}^2([a\,;\,b])$.

c'est-à-dire : pour tout entier i , $1 \leq i \leq n-1$, on a :

$p_i(x_{i+1}) = p_{i+1}(x_{i+1})$,
$p_i'(x_{i+1}) = p_{i+1}'(x_{i+1})$,
$p_i''(x_{i+1}) = p_{i+1}''(x_{i+1})$.

• On impose en outre : $p_0''(a) = 0$ et $p_{n-1}''(b) = 0$.

Quelques calculs préliminaires :

Pour tout $x \in [x_i,\,x_{i+1}]$, on a :

$$Spline(x) = p_i(x) = \sigma_i + \sigma_i'\,(x - x_i) + \sigma_i^{(2)}\,\frac{(x-x_i)^2}{2} + \sigma_i^{(3)}\,\frac{(x-x_i)^3}{6}$$

où $\sigma_i' = p_i'(x_i)\,;\, \sigma_i^{(2)} = p_i''(x_i)$ et $\sigma_i^{(3)} = p_i'''(x_i)$.

On peut montrer successivement :

pour $0 \leq i \leq n-1$, $\sigma_i^{(3)} = \dfrac{\sigma_{i+1}^{(2)} - \sigma_i^{(2)}}{h}$, en convenant que $\sigma_n^{(2)} = 0$;

pour $0 \leq i \leq n-1$, $\sigma_i' = \dfrac{\sigma_{i+1} - \sigma_i}{h} - \dfrac{h}{6}\left(\sigma_{i+1}^{(2)} + 2\sigma_i^{(2)}\right)$;

pour $1 \leq i \leq n-1$, $\sigma_{i+1}^{(2)} + 4\sigma_i^{(2)} + \sigma_{i-1}^{(2)} = \dfrac{6}{h^2}\,(\sigma_{i+1} - 2\sigma_i + \sigma_{i-1})$, en convenant que $\sigma_0^{(2)} = 0$.

On en déduit que ces dernières relations peuvent se réécrire sous la forme d'un système linéaire $Ax = b$ avec :

$$A = \begin{pmatrix} 4 & 1 & 0 & & \cdots & & 0 \\ 1 & 4 & 1 & 0 & & & \\ 0 & 1 & 4 & 1 & & & \vdots \\ & 0 & 1 & 4 & \ddots & & \\ \vdots & & & \ddots & \ddots & 1 & 0 \\ & & & & 1 & 4 & 1 \\ 0 & & \cdots & & 0 & 1 & 4 \end{pmatrix} ; x = \begin{pmatrix} \sigma_1^{(2)} \\ \sigma_2^{(2)} \\ \\ \vdots \\ \\ \\ \sigma_{n-1}^{(2)} \end{pmatrix} ; b = \frac{6}{h^2} \begin{pmatrix} \sigma_2 - 2\sigma_1 + \sigma_0 \\ \sigma_3 - 2\sigma_2 + \sigma_1 \\ \\ \vdots \\ \\ \\ \sigma_n - 2\sigma_{n-1} + \sigma_{n-2} \end{pmatrix}$$

Travail demandé :

Construire la fonction *Spline* et afficher sa représentation graphique.

Pour tester l'approximation, on peut partir d'une fonction f définie explicitement ;

par exemple, $f(x) = \cos(x)$ sur $[0\,;\,2\pi]$.

1. Construire la matrice A et le second membre b.
2. Résoudre le système linéaire $Ax = b$.
3. Calculer les nombres σ_i' et $\sigma_i^{(3)}$.
4. En déduire $spline(x)$ en repérant dans quel intervalle se situe x.
5. Afficher la fonction *Spline*.

Script Scilab

```
// Spline cubique d'interpolation
// -------------------------------------------
// définition de la fonction à approcher
function y=f(x)
    y=cos(x)
endfunction
// -------------------------------------------
// intervalle [a;b] et nombre n de subdivisions
utilisateur=x_mdialog('intervalle [a;b] et nombre n de subdivisions',...
['a';'b';'n'],['0';'6';'10']);
a=evstr(utilisateur(1));
b=evstr(utilisateur(2));
n=evstr(utilisateur(3));
h=(b-a)/n; // pas de la subdivision
// vecteur points d'interpolation yi=f(xi)
Y=[f(a:h:b)];
// construction de la matrice A
v=zeros(1,n-3);
A=toeplitz([4,1,v]);
// construction du second membre b
B=zeros(n-1,1);
for i=1:n-1
    B(i,1)=Y(i)-2*Y(i+1)+Y(i+2);
end
B=(6/h^2).*B;
// résolution du système AX=B
X=A\B;// remarque : on n'utilise pas ici le fait que A est tridiagonale
// ---------------------------------------------------------------
Y_2=[0,X',0]; // dérivées secondes aux points xi
// détermination du polynôme d'interpolation nommé spline(x)
function y=spline(x)
    // on détermine la subdivision contenant x
    if (x==b) then
        k=n-1;
    else k=floor((x-a)/h);
    end
    xk=a+k*h; // x appartient à l'intervalle [x_k ; x_k+1]
    // -------------------------------------------------------------
    Y_1=zeros(1,n); // dérivées aux points xi
    Y_3=zeros(1,n); // dérivées troisièmes aux points xi
    for i=1:n
        Y_1(i)=(Y(i+1)-Y(i))/h-(h/6)*(Y_2(i+1)+2*Y_2(i));
        Y_3(i)=(Y_2(i+1)-Y_2(i))/h;
    end
    // construction du polynôme d'interpolation p(x)
    s=poly(0,'s');
    t=s-xk;
    p=Y(k+1)+t*Y_1(k+1)+t^2*(Y_2(k+1)/2)+t^3*(Y_3(k+1)/6);
    y=horner(p,[x]);
endfunction
// -------------------------------
// graphique
// -------------------------------
x=[a:0.1:b];
P=[];
F=[];
for i=1:length(x)
    P=[P,spline(x(i))];
    F=[F,f(x(i))];
end
clf;
plot2d(x,F,1); // affiche la fonction f
plot2d(x,P,2); // affiche la spline cubique
```