Splines Cúbics

Quan es tracta d'aproximar una funció $f(x)$ per interpolació a partir d'un nombre molt elevat de punts, el polinomi corresponent haurà de ser de grau molt elevat. Ja hem vist que això pot donar lloc a errors molt grans, sobretot als extrems de l'interval (fenomen de Runge). Llavors, el que es fa en aquests casos es dividir l'interval en subintervals i, a cada subinterval, aproximar per polinomis de grau baix. Aquesta tècnica es coneix com *aproximació per splines* i va ser introduïda per Schoenberg en 1946 (veure [2, 3]). De manera més precisa:

Definició 1 (Bonet et al. [1])

Suposem que tenim un conjunt de punts ordenats $x_1 < x_2 < \cdots < x_N$. Una funció spline de grau p amb nodes en els punts x_i, $i = 1, \ldots, N$ és una funció $s(x)$ amb les propietats següents:

(i) En cada interval $[x_i, x_{i+1}]$, $i = 1, 2, \ldots, N-1$, $s(x)$ és un polinomi de grau p.

(ii) $s(x)$ i les seves $p-1$ primeres derivades són contínues a $[x_1, x_N]$.

El splines més utilitzats són els splines cúbics ($p = 3$). Es tracta doncs de buscar una un polinomi "a trossos", $s(x)$, de la forma

$$s(x) = \begin{cases} s_1(x), & x_1 \le x \le x_2 \\ s_2(x), & x_2 \le x \le x_3 \\ \vdots & \\ s_{N-1}(x), & x_{N-1} \le x \le x_N \end{cases} \tag{1}$$

(veure figura 1). Aquí $s_i(x)$, $i = 1, 2, \ldots, N-1$, són polinomis de grau 3, verificant les condicions següents:

1. s *interpola* la funció f, i.e., als nodes $x_1, x_2, \ldots, x_N$, els valors de s i f coincideixen:

$$s(x_i) = f(x_i), \tag{2}$$

   per $i = 1, 1, \ldots, N$.

2. s és de classe C^2 en l'interval $[x_1, x_N]$. Això implica:

$$\begin{aligned} s_i(x_{i+1}) &= s_{i+1}(x_{i+1}), \\ s_i'(x_{i+1}) &= s_{i+1}'(x_{i+1}), \\ s_i''(x_{i+1}) &= s_{i+1}''(x_{i+1}), \end{aligned} \tag{3}$$

   per $i = 1, 2, \ldots, N-2$.

3. Finalment, s'acostuma a imposar l'anul·lació de les derivades segones als extrems, i.e.:

$$s''(x_1) = s''(x_N) = 0. \tag{4}$$

Els splines cúbics que satisfan aquesta condició s'anomenen *splines cúbis naturals* i es demostra que aquests minimitzen $\int_{x_1}^{x_N} (s(x))^2 \, \mathrm{d}x$ sobre tots els splines cúbics que satisfan (2) i (3).

Notem que, per definir $s(x)$ necessitem fixar $4N-4$ coeficients: els corresponents als $N-1$ polinomis $s_1(x), s_1(x), \ldots, s_{N-1}(x)$, de grau 3. Ara bé:

- La condició 1 ens proporciona N equacions.
- La condició 2 ens proporciona $3(N-2)$ equacions.
- La condició 3 ens proporciona 2 equacions addicionals.

Amb la qual cosa resulten

$$N + 3(N-2) + 2 = 4N - 4$$

equacions *lineals* en els coeficients dels polinomis. Es demostra que el sistema que s'obté és compatible determinat. Aleshores els coeficients de $s_1(x), s_1(x), \ldots, s_{N-1}(x)$ vénen determinats de manera única. A la literatura, la construcció dels splines s'acostuma a dur a terme mitjançant l'anomenat *mètode dels moments*, que no explicarem en aquestes notes. El lectors interessats poden consultar, per exemple, el Capítol 4 de [1].

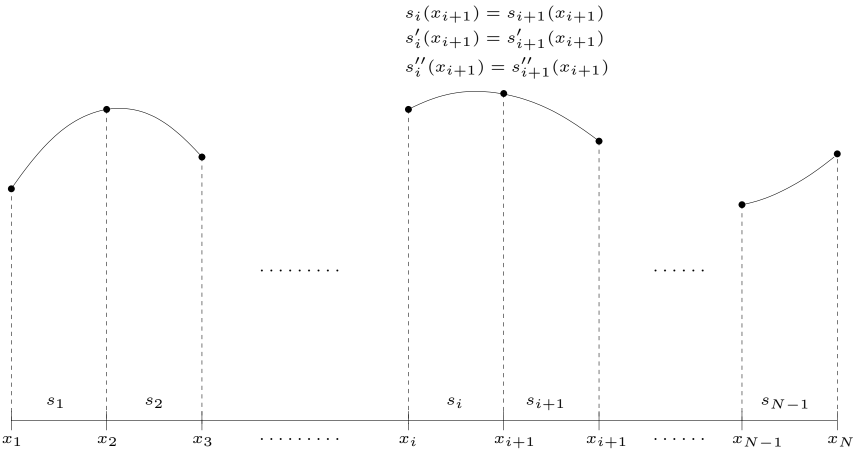

Figura 1 : Splines cúbicos

Spines cúbics amb Matlab/Octave

En Matlab/Octave, la funció `spline` calcula l'aproximació d'una funció $f(x)$ pel seu spline cúbic natural, $s(x)$:

```
>> sx = spline(x,f,xx)
```

- INPUT:
  - `x=[x(1),...,x(N)]` són les N abscisses (o *nodes*) que donen lloc a $N-1$ intervals: $[x_1, x_2], \ldots, [x_{N-1}, x_N]$. A cadascun d'ells, f s'aproxima per un polinomi de grau 3. Veure (1).
  - `f=[f(1),...,f(N)]` són els valors de f a `x(1),...,x(N)` respect.
  - `xx=[xx(1),...,xx(M)]` són M punts de l'interval $[x_1, x_N]$.
- OUTPUT:
  - `sx=[sx(1),...,sx(M)]` són els valors de $s(x)$ a `xx(1),...,xx(M)`.

Podeu consultar l'ajut del Matlab/Octave per obtenir més informació sobre l'ús de la funció `spline` escrivint a la línea de comandes:

```
>> help spline
```

Exemple 1

Si volem aproximar la funció:

$$f(x) = \frac{1}{1+30x^2}$$

per splines cúbics a l'interval $[-1,1]$ fent servir 11 nodes equiespaiats, podem escriure un fitxer (un *script*) Matlab/Octave amb les comandes:

```
clearvars
close all
f=@(x) 1./(1+30*x.^2); %f(x)=1/(1+30*x^2)
x=-1:0.2:1; xx=-1:0.01:1;
fx=f(x); fxx=f(xx);
sx=spline(x,fx,xx); %dona s(x) a xx(1),...,xx(m)
plot(x,fx,'og',xx,fxx,'b-',xx,sx,'r-');
hold on
xlabel('x'); ylabel('y');
hleg1=legend('Points','f(x)','s(x)','Location','NorthEast');
hold off
```

guardar-ho, per exemple, com `splines_script.m` i després excecutar-ho des de Matlab/Octave:

```
>> splines_script
```

Llavors resulta la gràfica de la Figura 2.

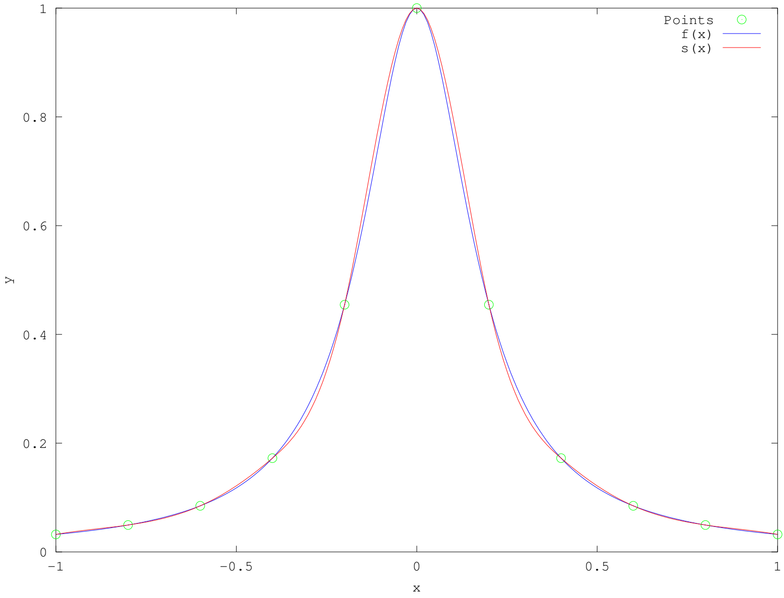

Figura 2 : Aproximació per splines cúbics de la funció $f(x) = \frac{1}{1+30x^2}$.

Splines lineals

Si en comptes d'aproximar per un polinomi de grau 3 aproximem, a cada subinterval $[x_i, x_{i+1}]_{i=0,1,\dots,n-1}$, la funció $f(x)$ per una recta i imposem continuïtat als extrems dels intervals, obtenim l'aproximació per *splines lineals* de $f(x)$. Geomètricament, estem aproximant la funció $f(x)$ per una *poligonal*, $\ell(x)$, que té per vèrtexs els punts coneguts de la funció $f(x)$: $(x_i, f(x_i))$, $i = 0, 1, \dots, n$.

En Matlab/Octave, la funció `interp1` permet trobar el valor de l'aproximació per splines lineals de $f(x)$:

```
>> Lx=interp1(x,f,xx)
```

- INPUT:
  - `x=[x(1),...,x(N)]` són les n abscisses (o *nodes*) que donen lloc a $N-1$ intervals: $[x_1, x_2], \dots, [x_{N-1}, x_N]$. A cadascun d'ells, f s'aproxima per un polinomi de grau 1, imposant continuïtat als extrems.
  - `f=[f(1),...,f(N)]` són els valors de f a `x(1),...,x(N)` respect.
  - `xx=[xx(1),...,xx(M)]` són M punts de l'interval $[x_1, x_N]$.

- OUTPUT:
  - `Lx=[Lx(1),...,Lx(M)]` són els valors de l'spline lineal $\ell(x)$ als punts `xx(1),...,xx(M)`.

Podeu consultar l'ajut del Matlab/Octave per obtenir més informació sobre l'ús de la funció `interp1` escrivint a la línea de comandes:

```
>> help interp1
```

Exercici: repetiu els càlculs de l'exemple 1 per trobar ara l'aproximació per splines lineals.

Referències

C. Bonet, A. Jorba, T. M-Seara, J. Masdemont, M. Ollé, T. Susín, and M. València.
Càlcul Numèric.
Edicions UPC, 1994.

I. J. Schoenberg.
Contributions to the problem of approximation of equidistant data by analytic functions. Part A. On the problem of smoothing or graduation. A first class of analytic approximation formulae.
Quart. Appl. Math., 4:45–99, 1946.

I. J. Schoenberg.
Contributions to the problem of approximation of equidistant data by analytic functions. Part B. On the problem of osculatory interpolation. A second class of analytic approximation formulae.
Quart. Appl. Math., 4:112–141, 1946.