Cubic splines

1 Cubic spline interpolation in 1D

Using cubic spline interpolation comes to the same thing that decomposing g on a cubic spline basis, and writting it for all point $r \in [r_0, r_{N_r}]$ like :

$$g(r) = \sum_{\nu=-1}^{N_r+1} c_\nu \Lambda_\nu(r)$$

where Λ_ν are piecewise cubic polynomials (represented in figure (1)), twice continuously differentiable, which are defined by :

$$\Lambda_\nu(r) = \frac{1}{6h^3} \begin{cases} (r - r_{\nu-2})^3 & \text{if } r_{\nu-2} \leq r \leq r_{\nu-1} \\ h^3 + 3h^2(r - r_{\nu-1}) + 3h(r - r_{\nu-1})^2 & \\ \quad -3(r - r_{\nu-1})^3 & \text{if } r_{\nu-1} \leq r \leq r_\nu \\ h^3 + 3h^2(r_{\nu+1} - r) + 3h(r_{\nu+1} - r)^2 & \\ \quad -3(r_{\nu+1} - r)^3 & \text{if } r_\nu \leq r \leq r_{\nu+1} \\ (r_{\nu+2} - r)^3 & \text{if } r_{\nu+1} \leq r \leq r_{\nu+2} \\ 0 & \text{otherwise} \end{cases}$$

, where $h = |r_{N_r} - r_0| / N_r$.

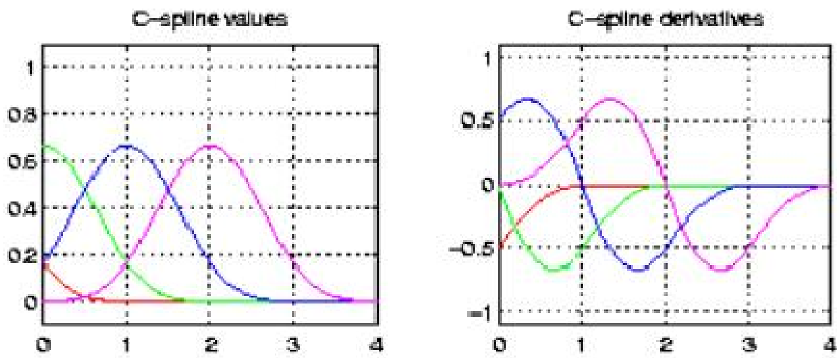

Figure 1: Piecewise cubic polynomials vector of the cubic spline basis and its first derivatives

The c_ν coefficients are computed like the solution of the following equation system

$$g(r_i) = \sum_{\nu=-1}^{N_r+1} c_\nu \Lambda_\nu(r_i) \qquad i = 0, \cdots, N_r \tag{1}$$

This system is differently solved, according to the boundary conditions we have to treat. Indeed we have two kinds of conditions : non-periodic and periodic, because we suppose non-periodic boundary conditions in r and $v_{\parallel l}$ directions and periodic conditions in the two others directions.
The equation system (1) can be written in the following matricial form :

$$[\Lambda]_{(N_r+1)\times(N_r+1)} \begin{pmatrix} c_{-1} \\ \vdots \\ c_{N_r+1} \end{pmatrix} = \begin{pmatrix} g(r_0) \\ \vdots \\ g(r_{N_r}) \end{pmatrix}$$

There are so (N_r+1) equations and (N_r+3) unknowns.

1.1 Non-periodic boundary conditions :

In the non-periodic case, two supplementary equations are needed. Let s be the function defined by :

$$s(r) = \sum_{\nu=-1}^{N_r+1} c_\nu \Lambda_\nu(r)$$

The solution is to impose conditions on the first or second derivates like :

$$\begin{cases} g'(r_0) = s'(r_0) \\ g'(r_{N_r}) = s'(r_{N_r}) \end{cases} \quad \text{or} \quad \begin{cases} g''(r_0) = s''(r_0) \\ g''(r_{N_r}) = s''(r_{N_r}) \end{cases}$$

According to the following table :

r	$r_{\nu-2}$	$r_{\nu-1}$	r_ν	$r_{\nu+1}$	$r_{\nu+2}$
$\Lambda_\nu(r)$	0	1	4	1	0
$\Lambda'_\nu(r)$	0	$3/h$	0	$-3/h$	0
$\Lambda''_\nu(r)$	0	$6/h^2$	$-12/h^2$	$6/h^2$	0

the (N_r+3, N_r+3) matricial system to be solved, becomes in the first case :

$$\begin{pmatrix} -3/h & 0 & 3/h & & & \\ 1 & 4 & 1 & & 0 & \\ & 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & & \\ 0 & & 1 & 4 & 1 & \\ & & -3/h & 0 & 3/h & \end{pmatrix} \times \begin{pmatrix} c_{-1} \\ c_0 \\ \vdots \\ \vdots \\ c_{N_r} \\ c_{N_r+1} \end{pmatrix} = \begin{pmatrix} g'(r_0) \\ g(r_0) \\ \vdots \\ \vdots \\ g(r_{N_r}) \\ g'(r_{N_r}) \end{pmatrix} \tag{2}$$

For the case with the conditions on the second derivates, only the first and the last rows are respectively replaced by :

$$\begin{pmatrix} 6/h^2 & -12/h^2 & 6/h^2 & 0 & \cdots & 0 \end{pmatrix} \times \vec{c} = g''(r_0) \quad \text{and}$$
$$\begin{pmatrix} 0 & \cdots & 0 & 6/h^2 & -12/h^2 & 6/h^2 \end{pmatrix} \times \vec{c} = g''(r_{N_r})$$

, where $\vec{c} = \begin{pmatrix} c_{-1} & c_0 & \cdots & c_{N_r} & c_{N_r+1} \end{pmatrix}^t$.

If we permute this system to keep the boundary conditions in the two last rows, then (2) can be written like :

$$\tilde{A}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix} \qquad \text{where} \begin{cases} x = (c_0, \cdots, c_{N_r})^t \\ y = (c_{N_r+1}, c_{-1})^t \\ u = (g(r_0), \cdots, g(r_{N_r}))^t \\ v = (g'(r_{N_r}), g'(r_0))^t \end{cases}$$

and

$$\tilde{A} = \left(\begin{array}{c|cc} A & \multicolumn{2}{c}{\gamma} \\ \hline \lambda & \begin{matrix} \xi_1 \\ \xi_3 \end{matrix} & \begin{matrix} \xi_2 \\ \xi_4 \end{matrix} \end{array}\right)$$

where

$$\begin{cases} \cdot\, A \text{ is the } (N_r+1)\times(N_r+1) \text{ tridiagonal symmetric matrix : } \begin{pmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{pmatrix}, \\ \cdot\, \lambda \text{ is equal to the } 2\times(N_r+1) \text{ matrix : } \begin{pmatrix} 0 & \cdots & 0 & -3/h & 0 \\ 0 & 3/h & 0 & \cdots & 0 \end{pmatrix}, \\ \cdot\, \gamma \text{ is equal to the } (N_r+1)\times 2 \text{ matrix : } \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}^t \text{ and} \\ \cdot\, \delta = \begin{pmatrix} \xi_1 & \xi_2 \\ \xi_3 & \xi_4 \end{pmatrix} = \begin{pmatrix} 3/h & 0 \\ 0 & -3/h \end{pmatrix} \end{cases}$$

Besides, $\tilde{A}$ can be factorized in the LU form, like :

$$\tilde{A} = \begin{pmatrix} A & 0 \\ \lambda & \bar{\delta} \end{pmatrix} \times \begin{pmatrix} I & A^{-1}\gamma \\ 0 & I \end{pmatrix} \qquad \text{with} \qquad \bar{\delta} = \delta - \lambda A^{-1}\gamma$$

This LU factorization is used to solve, by forward and backward substitutions, the (2) equivalent system :

$$\begin{pmatrix} A & 0 \\ \lambda & \bar{\delta} \end{pmatrix} \times \begin{pmatrix} I & A^{-1}\gamma \\ 0 & I \end{pmatrix} \times \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix}$$

That means, at first, solving :

$$\begin{pmatrix} A & 0 \\ \lambda & \bar{\delta} \end{pmatrix} \times \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix}$$

and then

$$\begin{pmatrix} I & A^{-1}\gamma \\ 0 & I \end{pmatrix} \times \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix}$$

So the interpolation coefficient computation can be summarized in the following steps :

1. Initialization :
   (a) Factorize and store A,
   (b) Compute and store $A^{-1}\gamma$ and
   (c) Compute and store the (2×2) matrix $\bar{\delta} = \delta - \lambda A^{-1}\gamma$.
2. Time loop :
   (a) Compute and store $x' = A^{-1}u$ using the previously computed factorization of A,
   (b) Assemble $v - \lambda A^{-1}u$,
   (c) Solve the (2×2) system $\bar{\delta}y' = v - \lambda A^{-1}u$ using the Cramer formulae for $\bar{\delta}$ inverse computation $\bar{\delta}^{-1} = \frac{1}{\det(\bar{\delta})} \begin{pmatrix} \bar{\xi_4} & -\bar{\xi_2} \\ -\bar{\xi_3} & \bar{\xi_1} \end{pmatrix}$ and
   (d) Compute x using the previous storage of $A^{-1}\gamma$ by $x = x' - A^{-1}\gamma y$, where y is trivially equal to y'.

This involves a first tridiagonal symmetric system resolution in the initialization and then one at each time loop. These systems are solved with LAPACK library subroutines using a LDL^t factorization for A.

1.2 Periodic boundary conditions :

In the periodic case, we have the $N_r + 1$ equations :

$$g(r_i) = \sum_{\nu=0}^{N_r} c_\nu \Lambda_\nu(r_i) \qquad i = 0, \cdots, N_r \tag{3}$$

but we need two others equations, because of the $N_r + 3$ unknowns. We cannot used the equation $g(r_0) = g(r_{N_r})$ because it is linearly dependent of the previous system. So we use the first and second derivatives continuity property of the cubic splines :

$$\begin{cases} g'(r_0) = g'(r_n) \\ g''(r_0) = g''(r_n) \end{cases}$$

which give respectively :

$$-\frac{3}{h}c_{-1}+\frac{3}{h}c_1+\frac{3}{h}c_{N_r-1}-\frac{3}{h}c_{N_r+1}=0$$

and

$$\frac{6}{h^2}c_{-1}-\frac{12}{h^2}c_0+\frac{6}{h^2}c_1-\frac{6}{h^2}c_{N_r-1}+\frac{12}{h^2}c_{N_r}-\frac{6}{h^2}c_{N_r+1}=0$$

The equivalent $(N_r+3)\times(N_r+3)$ matricial system becomes :

$$\begin{pmatrix} 4 & 1 & 0 & & \cdots & \cdots & 0 & 0 & 1 \\ & 1 & 4 & 1 & & & & \vdots & 0 \\ & & & \ddots & \ddots & \ddots & 1 & 0 & \vdots \\ & & & & & 1 & 4 & 1 & 0 \\ 0 & \frac{3}{h} & 0 & \cdots & 0 & \frac{3}{h} & 0 & -\frac{3}{h} & -\frac{3}{h} \\ -\frac{12}{h^2} & \frac{6}{h^2} & 0 & \cdots & 0 & -\frac{6}{h^2} & \frac{12}{h^2} & -\frac{6}{h^2} & \frac{6}{h^2} \end{pmatrix} \times \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{N_r-1} \\ c_{N_r} \\ c_{N_r+1} \\ c_{-1} \end{pmatrix} = \begin{pmatrix} g(r_0) \\ g(r_1) \\ \vdots \\ g(r_{N_r-1}) \\ g(r_{N_r}) \\ 0 \\ 0 \end{pmatrix} \tag{4}$$

which can be solved with the same method than the previous matricial system ; where the sole differences are :

$$\begin{cases} \cdot\, v=(0,0)^t, \\ \cdot\, \lambda=\begin{pmatrix} 0 & 3/h & 0 \cdots 0 & 3/h & 0 \\ -12/h^2 & 6/h^2 & 0 \cdots 0 & -6/h^2 & 12/h^2 \end{pmatrix} \text{ and} \\ \cdot\, \delta=\begin{pmatrix} \xi_1 & \xi_2 \\ \xi_3 & \xi_4 \end{pmatrix}=\begin{pmatrix} -3/h & -3/h \\ -6/h^2 & 6/h^2 \end{pmatrix} \end{cases}$$