TRINITY COLLEGE DUBLIN, SCHOOL OF MATHEMATICS

Assignment #2, Module: MA5633

Gustavo Ramirez

November 9, 2016

1 PART 2 OF ASSIGNMENT: INTERPOLATION BY SPLINES

A brief presentation is made here, of all the mathematical tools needed to implement the natural and complete spline approximations, in the general case.

2 General Construction

Purpose: *obtain a cubic interpolating spline, a function s, such that $s(x_i) = y_i$ for given values y_i, where $a = x_0 < x_1 < ... < x_m = b$. s_i is a cubic polynomial in the interval* $[x_i, x_{i+1}]$. There are $m+1$ points, and therefore m sub-intervals.

Restrictions on the system: restrictions are imposed over the values of the second derivative at the end points of the total interval: $s''(x_0)$ and $s''(x_m)$, for the case of the natural spline, but restrictions on the continuity of the first derivative are imposed in the case of complete spline. Those two restrictions will lead to two additional equations, contributing to find the total solution of the system. There are two options for imposing these restrictions on the second derivative:

1. natural splines: $s''_0(x_0) = s''_{m-1}(x_m) = 0$
2. complete splines: $s'_0(x_0) = f'(x_0)$ and $s'_{m-1}(x_m) = f'(x_m)$

System of equations:
If each of the m cubic functions s_i have the form: $s(x_i) = a_i(x-x_i)^3 + b_i(x-x_i)^2 + c_i(x-x_i) + d_i$, then there are a total of $4m$ parameters to be determined. Then, $4m$ conditions are needed. These conditions are:

1. $s(x_i) = y_i$, $\quad$ i for all points but the last
2. $s_{m-1} = y_m$, $\quad$ last point
3. Boundary conditions on the function, derivative and second derivative; these three type of bounday conditions lead to $m-1$ each, because there are $m-1$ internal points (bounded by the m internal intervals).

Is clear that for the first condition, is necessary to avoid the last point, because s_i is defined over $[x_i, x_{i+1}]$, so there is no s_m spline. For the last point, the second condition applies the restriction.
Therefore, the number of conditions is: $m + 1 + 3(m-1) = 4m - 2$

And the last two conditions necessary for completing the $4m$ conditions, are the restrictions imposed on the second derivatives at the edges.
Using an equidistant spacing $h = x_{i+1} - x_i$, and combining all the $4m$ restrictions, the resulting system of equations is (re-labeling the second derivatives on s as new σ variables):

- $\sigma_{i-1} + 4\sigma_i + \sigma_{i+1} = 6\left(\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}\right), \qquad i = 1, ..., m-1$
- $c_i = \frac{y_{i+1} - y_i}{h} - h\frac{2\sigma_i + \sigma_{i+1}}{6}$
- $a_i = \frac{\sigma_{i+1} - \sigma_i}{6h}$
- $b_i = \frac{\sigma_i}{2}$
- $d_i = y_i$

and in the last four equations: $i = 0, ..., m-1$.

3 Natural Splines

Restrictions: $\sigma_0 = \sigma_m = 0$, which leads to the matrix formulation:

$$\begin{bmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & & ... & & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ ... \\ \sigma_{m-2} \\ \sigma_{m-1} \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} y_2 - 2y_1 + y_0 \\ y_3 - 2y_2 + y_1 \\ ... \\ y_{m-1} - 2y_{m-2} + y_{m-3} \\ y_m - 2y_{m-1} + y_{m-2} \end{bmatrix} \tag{3.1}$$

It is important to note that this previous matrix is of size $(m-1) \times (m-1)$.

4 Complete Splines

In this second case, the matrix on the left side is expanded, in such a manner that contains the matrix of natural case, but three of the external walls in the matrix are modified.
This is done by adding two more equations to the total system of equations, resulting in a system of $m+1$ equations, and eventually in a expanded matrix of size $(m+1) \times (m+1)$.
In the implemented Python code, the procedures are the same, with just two simple if statements added to add some more entries in the case of the complete spline.

5 On the results

It is very clear that the cubic splines are supposed to generate (in general) better results when approximating functions, i.e. decrease the errors, when compared to the polynomial approximation. This is because of the restrictions taken. In the polynomial case, the idea is to minimize the correlation, and from there conditions are stablished for the parameters of the piecewise polynomial defined by regions over the independent variable.

On the other hand, the cubic spline is richer in information, in the sense that the conditions for the parameters are obtained from a more varied set of conditions, those corresponding to specific features of the original function, e.g. first derivative, second derivative, etc.

As can be seen from the plots in the directory ./plots/, the error is decreased considerably when compared to the polynomial approximations studied in class.

When comparing the natural with the complete approximation, forcing the first derivative to coincide in the extremes in the complete case, increases the error on both sides of an specific approximation. But otherwise, in general, both approximations turned out better than the polynomial approximation, and both are very similar in shape, up to the point that at a grid value of 0,0625 they are almost equal (see figures ./plots/SplineFit_complete_h0.0625.png and ./plots/SplineFit_natural_h0.0625.png for this).