Cubic spline interpolation

July 26, 2021

Suppose you are after a girl, and you want to impress her with your mathematical knowledge. Do some research about the 'heart curve', and we are going to do something similar - express some romantic image with a mathematical function[1]. In this tutorial, I will show you how this can be done in 3 steps:

- First, find an image of her favourite cartoon or anime, e.g. a portrait of Monkey·D·Luffy. Alternatively, find a figure that has a special meaning between you and her, e.g. a token of love, if you have any.

- Second, record some coordinates of points on the boundary of your chosen image.

- Third, find an mathematical curve that goes through all those points on the boundary.

The first two steps do not sound very hard, but what about the final step? Well, what we want to do is actually called interpolation.

Consider a set of points (x_i, y_i), where x_i a strictly increasing sequence. We want to find a smooth curve that goes through all points (x_i, y_i). Surely we can connect neighbouring points with straight line sections, but we won't get a smooth curve after all, and your girl probably won't like it too much because of those many kinks. Instead, let's connect consecutive points with cubic polynomials, and let's call those curves 'splines'.
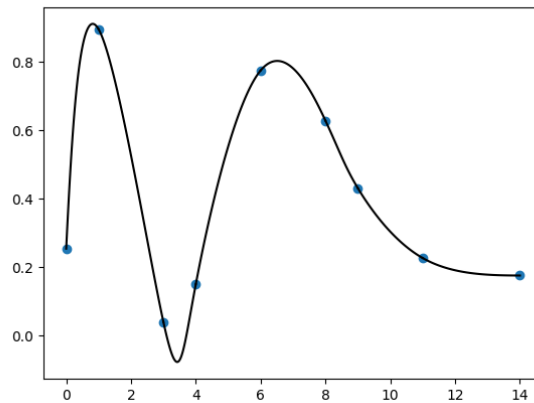

Figure 1: An example of interpolation 9 points using cubic splines.

A cubic polynomial has 4 coefficients:

$$s_i(x) = a_i + b_i x + c_i x^2 + d_i x^3. \tag{1}$$

If we want to connect $N+1$ points, we need N cubic polynomials, which means in total $4N$ coefficients. From high school we know that at least $4N$ equations are needed to solve for $4N$ unknowns.

---

[1]If the girl likes your maths function, it's probably because she already likes you, and vice versus.

For the ith spline $s_i(x)$, its left and right ends must go through (x_{i-1}, y_{i-1}) and (x_i, y_i) respectively:

$$s_i(x_{i-1}) = y_{i-1},$$
$$s_i(x_i) = y_i. \tag{2}$$

This will contributes 2 equations every spline, so in total $2N$ equations. In order to achieve 'smoothness', we require further that the 1st and 2nd derivatives of neighbouring splines agree at the nodes:

$$s_i'(x_i) = s_{i+1}'(x_i),$$
$$s_i''(x_i) = s_{i+1}''(x_i). \tag{3}$$

This condition will impose another $2N - 2$ equations (why $2N - 2$? Since for N splines, there are only $(N - 1)$ joints!). What about the last 2 equations? we can make up at the ends, e.g.:

$$s_1''(x_0) = 0$$
$$s_N''(x_N) = 0. \tag{4}$$

Now we have enough information to solve for those cubic polynomials. Let's differentiate eq.[1] twice to get the 2nd derivatives of splines:

$$s_i''(x) = 2c_i + 6d_i x. \tag{5}$$

The expression of $s_i''(x)$ indicates that it is a straight line connecting $(x_{i-1}, s_i''(x_{i-1}))$ and $(x_i, s_i''(x_i))$. Denote $h_i = x_i - x_{i-1}$, and $\sigma_i = s_i''(x_i)$, we obtain an expression for $s_i''(x_i)$,

$$s_i''(x) = \frac{x_i - x}{h_i}\sigma_{i-1} + \frac{x - x_{i-1}}{h_i}\sigma_i. \tag{6}$$

You can easily verify that eq.[6] does go through $(x_{i-1}, s_i''(x_{i-1}))$ and $(x_i, s_i''(x_i))$, and is a straight line.

Next, we integrate eq.[6] twice:

$$s_i(x) = \frac{(x_i - x)^3}{6h_i}\sigma_{i-1} + \frac{(x - x_{i-1})^3}{6h_i}\sigma_i + \alpha_i(x - x_{i-1}) + \beta_i(x_i - x), \tag{7}$$

where the last two terms are from the integration constants. Substitute eq.[7] into eq.[2]:

$$\frac{(x_i - x_{i-1})^3}{6h_i}\sigma_{i-1} + \frac{(x_{i-1} - x_{i-1})^3}{6h_i}\sigma_i + \alpha_i(x_{i-1} - x_{i-1}) + \beta_i(x_i - x_{i-1}) = y_{i-1}$$

$$\frac{1}{6}\sigma_{i-1}h_i^2 + h_i\beta_i = y_{i-1} \tag{8}$$

$$\beta_i = \frac{y_{i-1} - \frac{1}{6}\sigma_{i-1}h_i^2}{h_i}$$

$$\frac{(x_i - x_i)^3}{6h_i}\sigma_{i-1} + \frac{(x_i - x_{i-1})^3}{6h_i}\sigma_i + \alpha_i(x_i - x_{i-1}) + \beta_i(x_i - x_i) = y_i$$

$$\frac{1}{6}\sigma_i h_i^2 + h_i\alpha_i = y_i \tag{9}$$

$$\alpha_i = \frac{y_i - \frac{1}{6}\sigma_i h_i^2}{h_i}.$$

So far we haven't used the continuity of 1st derivatives. Integrate eq.[6] once and substitute eq.[8] and eq.[9],

$$s_i'(x) = -\frac{(x_i - x)^2}{2h_i}\sigma_{i-1} + \frac{(x - x_{i-1})^2}{2h_i}\sigma_i + \alpha_i - \beta_i \tag{10}$$

$$= -\frac{(x_i - x)^2}{2h_i}\sigma_{i-1} + \frac{(x - x_{i-1})^2}{2h_i}\sigma_i + \frac{y_i - \frac{1}{6}\sigma_i h_i^2}{h_i} - \frac{y_{i-1} - \frac{1}{6}\sigma_{i-1}h_i^2}{h_i}. \tag{11}$$

Similarly for $s'_{i+1}(x)$:

$$s'_{i+1}(x) = -\frac{(x_{i+1}-x)^2}{2h_{i+1}}\sigma_i + \frac{(x-x_i)^2}{2h_{i+1}}\sigma_{i+1} + \frac{y_{i+1}-\frac{1}{6}\sigma_{i+1}h_{i+1}^2}{h_{i+1}} - \frac{y_i-\frac{1}{6}\sigma_i h_{i+1}^2}{h_{i+1}}. \qquad (12)$$

Use the continuity of 1st derivatives $s'_i(x_i) = s'_{i+1}(x_i)$, and you may skip all the algebra below...

$$\frac{(x_i-x_{i-1})^2}{2h_i}\sigma_i + \frac{y_i-\frac{1}{6}\sigma_i h_i^2}{h_i} - \frac{y_{i-1}-\frac{1}{6}\sigma_{i-1}h_i^2}{h_i} = -\frac{(x_{i+1}-x_i)^2}{2h_{i+1}}\sigma_i + \frac{y_{i+1}-\frac{1}{6}\sigma_{i+1}h_{i+1}^2}{h_{i+1}} - \frac{y_i-\frac{1}{6}\sigma_i h_{i+1}^2}{h_{i+1}}$$

$$\frac{1}{2}h_i\sigma_i + \frac{y_i}{h_i} - \frac{1}{6}\sigma_i h_i - \frac{y_{i-1}}{h_i} + \frac{1}{6}\sigma_{i-1}h_i = -\frac{1}{2}h_{i+1}\sigma_i + \frac{y_{i+1}}{h_{i+1}} - \frac{1}{6}\sigma_{i+1}h_{i+1} - \frac{y_i}{h_{i+1}} + \frac{1}{6}\sigma_i h_{i+1}$$

$$3h_i\sigma_i - \sigma_i h_i + \sigma_{i-1}h_i + 3h_{i+1}\sigma_i + \sigma_{i+1}h_{i+1} - \sigma_i h_{i+1} = 6\left(\frac{y_{i+1}-y_i}{h_{i+1}} - \frac{y_i-y_{i-1}}{h_i}\right)$$

$$h_i\sigma_{i-1} + 2(h_{i+1}+h_i)\sigma_i + h_{i+1}\sigma_{i+1} = 6\left(\frac{y_{i+1}-y_i}{h_{i+1}} - \frac{y_i-y_{i-1}}{h_i}\right). \qquad (13)$$

Look at the final expression. This reminds us of writing it down as a matrix form $\mathbf{A}\sigma = \mathbf{b}$, where $\mathbf{A}$ is a tridiagonal matrix, with diagonals $2(h_{i+1}+h_i)$ and sub-diagonals h_i and h_{i+1}. The vector σ are made of σ_is and the vector $\mathbf{b}$ are made of $6\left(\frac{y_{i+1}-y_i}{h_{i+1}} - \frac{y_i-y_{i-1}}{h_i}\right)$s.

We then solve $\mathbf{A}\sigma = \mathbf{b}$ for σ, and substitute σs into eq.[7] to get our cubic splines!

Advanced topic (challenging for pre-uni students!)

Previously we only allow strictly increasing sequence x_i, which suggests that we are only able to interpolate a single-valued function $f(x_i) = y_i$. A closed curve is, however, multivalued. We must improve our method in order to interpolate a closed curve.

The theory is simple, we introduce a new parametric variable, e.g. the indices i of the boundary points, which is strictly increasing. We find the cubic spline interpolations of the two sets of points (i, x_i) and (i, y_i) using the method above. The 0th, 1st and 2nd derivatives of x w.r.t. i at the nodes agree to one another, so are that for y. By using chain rules, the 0th, 1st and 2nd derivatives of y w.r.t. x also agree at nodes, which means the parametric functions $(x(i), y(i))$ are indeed cubic spline interpolation of our closed curve!

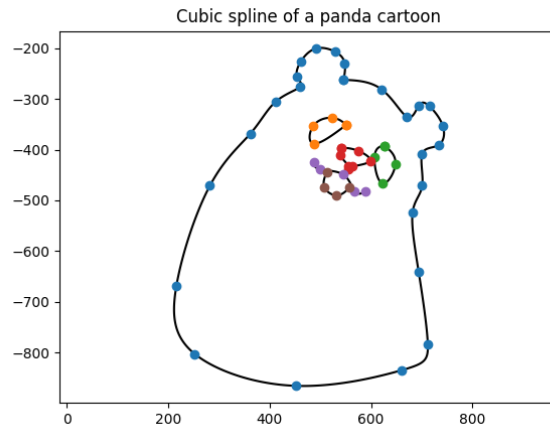The example code can be found on https://github.com/racksa/CubicSplineInterpolation

Figure 2: An example of interpolation a panda cartoon.