Exam assignment: Implementing Akima spline

Jedrzej Jawor

19/06/2018

1 Introduction

This report seeks to implement the Akima sub spline and an another cubic sub-spline and compare them with the cubic spline.
This is a combination of the "Akima spline" and "Yet another cubic sub-spline" exercises from the available exam exercises. I chose to do both, as "Akima spline" (which I originally did) turned out not to be a part of the valid exam exercises.

2 Theory

Interpolation is a process where one constructs a smooth function to fit a set of discrete data points. This is often done in order to approximate experimental values at points where no data is available.
A spline, S(x), is a piecewise polynomial used for interpolation between two points in a discrete data set:

$$S(x) = S_i(x), x \in [x_i, x_{i+1}], \tag{1}$$

for i=[1,n-1] being the index of the tabulated points. S_i is a polynomial of order k.
Splines for a discrete data set are calculated by demanding the continuity of the splines and their derivatives at the tabulated points [1].
The efficiency of splines as interpolating functions varies as splines are prone to wiggling.
One of the more common choices is a cubic spline (k=3). Cubic splines (cplines) usually have high precision but they can also be prone to wiggling if the interpolated set has a large discontinuity or outliers.

If higher precision near such point is desired, a sub-spline can be used. Sub splines are variations on the spline method that modify a boundary condition of the spline in order to achieve some form of improvement.
One such sub spline is the Akima sub-spline (aspline). Asplines are especially useful for data sets with discontinuities as they wiggle much less than csplies.

However, they come with drawbacks, sacrificing maximal differentiability by treating one of the coefficient as a free variable used for wiggle reduction instead of demanding continuity of the second derivative.
The method and equations for calculating coefficients for csplines and asplines, on which the code used in this *project is based on, come from [1].

2.1 Cubic sub-spline

The other spline that will be used in this project is a cubic sub-spline (cs-spline). This spline requires that we know the derivatives of the tabulated function. So we need a data set:

$$x_i, y_i, p_i | i = 1...n. \tag{2}$$

Where p_i is the derivative at the i'th point.
The i'th cs-spline has the same form as a cubic spline:

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

Where $h = (x - x_i)$. In order to find the proper coefficients (a_i, b_i, c_i, d_i) for the spline, we require that four boundary conditions be satisfied.

1. $S_i(x_i) = y_i \Rightarrow a_i = y_i$
2. $S'(x_i) = p_i \Rightarrow b_i = p_i$
3. $S(x_{i+1}) = y_{i+1} \Rightarrow c_i = \frac{dy_i - b_i dx_i - d_i dx_i^3}{dx_i^2}$
4. $S'(x_{i+1} = p_{i+1} \Rightarrow d_i = \left(\frac{dp_i}{3dx_i^2} - \frac{1}{3dx_i^3}(dy_i - b_i dx_i)\right)\left(1 + \frac{1}{3}\right)^{-1}$

Here I used: $dx_i = x_{i+1} - x_i$, $dy_i = y_{i+1} - y_i$ and $dp_i = p_{i+1} - pi$, S' is the derivative of S with respect to x.
The four conditions are sufficient for finding all coefficients as c_i can be found after finding d_i. It is therefore possible to find all cs-splines using only the tabulated function and its derivative.

3 Results and Discussion

Using the equations for asplies and csplines from [1], an aspline for a manually tabulated data set with two large outliers is constructed. The derivatives at each point are evaluated using a quadratic spline and used for constructing a cs-spline. A cspline is also constructed and plotted so that the sub-splines may be compared to it.
The discrete points and the splines are shown in figure 1. It can be seen that the akima spline wiggles much less that the cubic spline. The cs-spline on the other hand seems to wiggle much more. I think this is a result of the quadratic spline giving a poor estimate of the derivative.

In order to check if the derivative is the reason for the poor quality of the cs-spline, I make another data set where the derivatives are manually tabulated together with the x,y points. Again, aspline, cspline and cs-spline are constructed, see figure 2.
The manual estimate of the derivative makes the cs-spline wiggle much less. It wiggles less than the cspline.
The Akima spline is clearly the superior option if one is in need of a spline for a data set with outliers. The cs-spline on the other hand seems only to be useful if one has a precise estimate of the derivatives at the tabulated points.

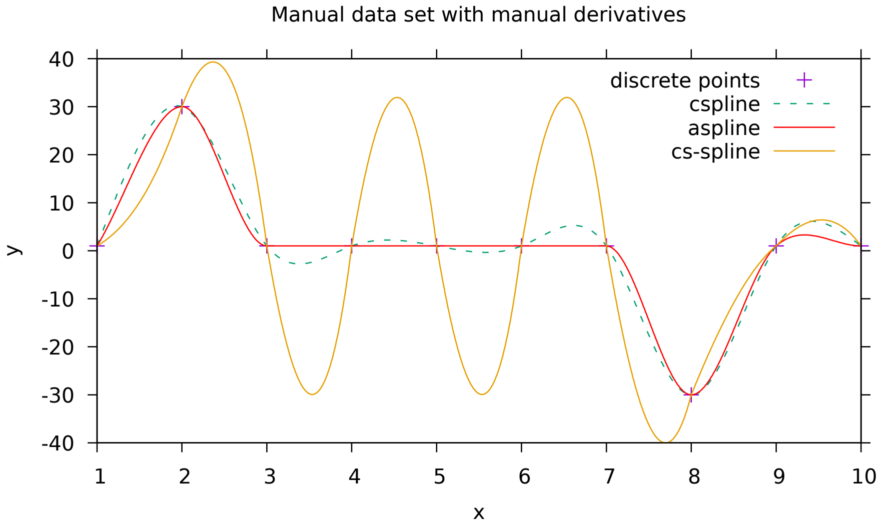

Figure 1: Aspline and cs-spline for a manually tabulated data set with derivatives found using a quadratic spline.

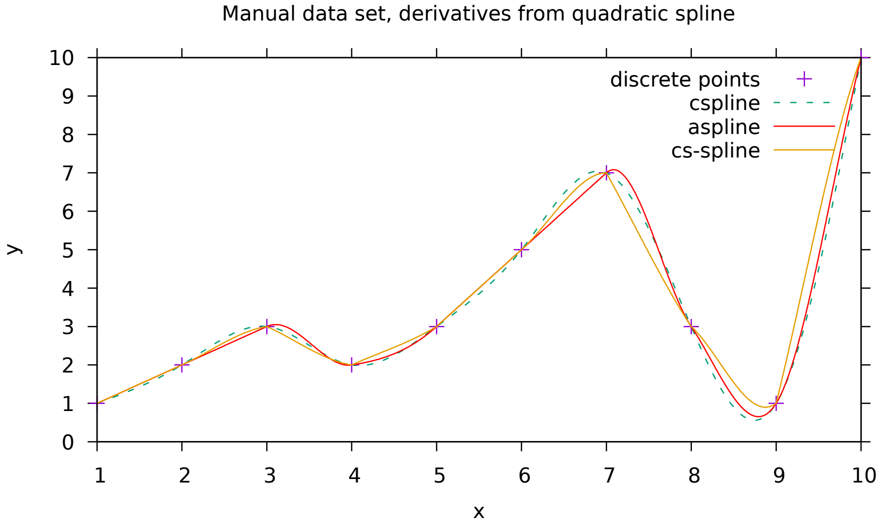

Figure 2: Aspline and cs-splne for a manually tabulated data set with manually tabulated derivatives.

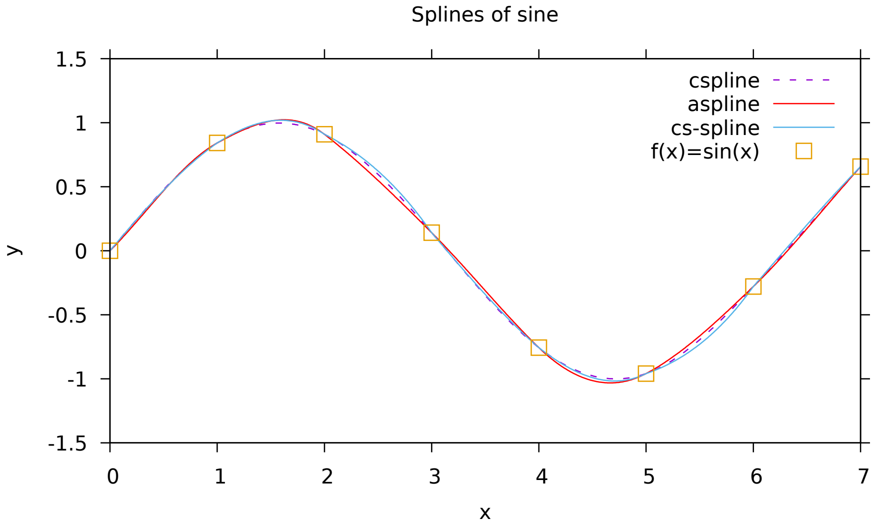

Figure 3: Aspline and cs-spline for $f(x)$.

I also want to test the derivatives and integrals of the aspline and cs-spline. For this purpose a sine function

$$f(x) = sin(x)$$

is tabulated in discrete steps. The derivative is tabulated using the analytic expression $\frac{d}{dx}f(x) = cos(x)$. The sine was chosen as I can compare the derivatives and integrals with the analytic solutions.
The a- and cs-splines for $f(x)$ are constructed and their derivative and the second derivative are found by differentiation. The splines are plotted in figure 3 while the plots of the derivatives are shown in figures 4 and 5. The analytic derivatives are also plotted.
The first derivative seems to be quite bad for both the aspline and cs-spline. They both follow the general trend but seem to behave oddly close to the tabulated points. The second derivatives are even worse. The aspline and cs-spline both become discontinuous.
In both cases, it seems that the cspline is the better option if one is interested in finding the derivatives of the tabulated points.

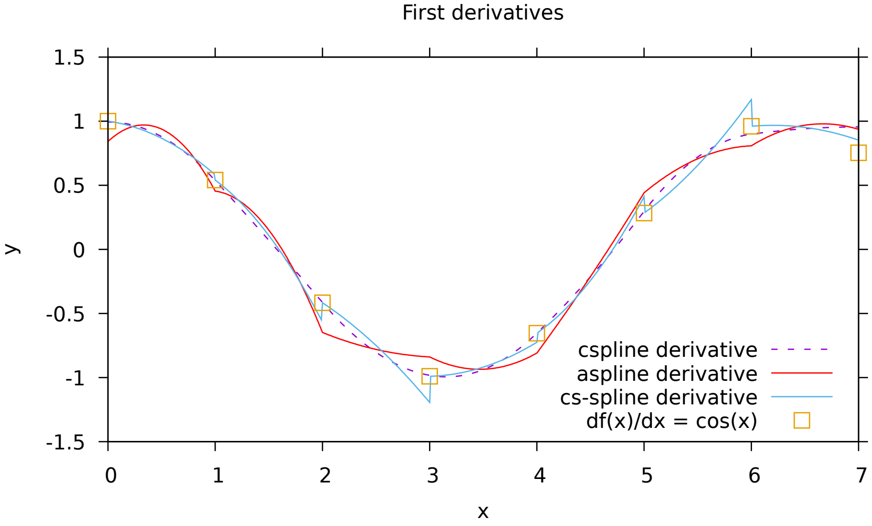

Figure 4: Derivative of aspline and cs-spline for $f(x)$.

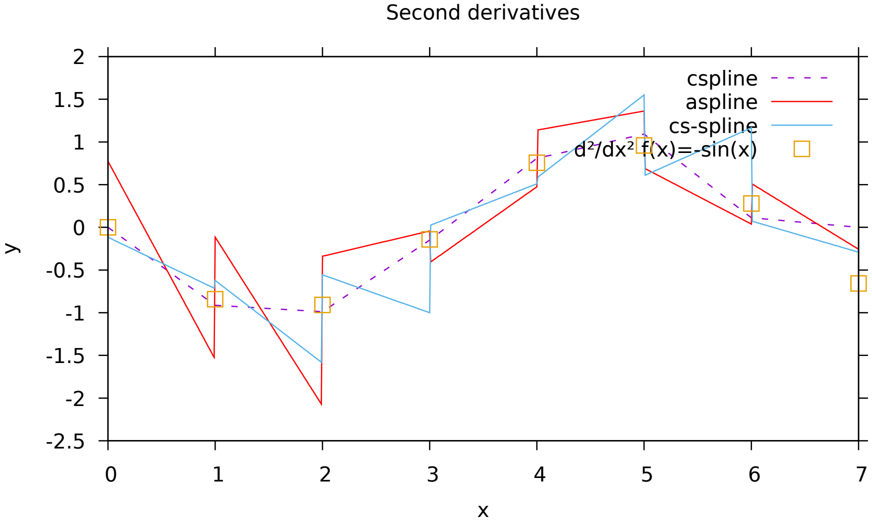

Figure 5: Second derivative of cspline and cs-spline for $f(x)$.

Now the integration of the aspline and cs-spline is tested. Again using the sine function $f(x)$. The analytical integral of $f(x)$ from 0 to 4 is found to be:

$$\int_0^4 f(x) = 1.65364. \tag{4}$$

The aspline integration yields:

$$\int_0^4 aspline = 1.6510. \tag{5}$$

The cspline integration yields:

$$\int_0^4 cspline = 1.65076. \tag{6}$$

The cs-spline integration yields:

$$\int_0^4 \text{cs-spline} = 1.68317. \tag{7}$$

So the integration seems to work quite well for both the cs-spline and aspline (at least for a nice analytic function like sine). The cs-splines seems to be the worst one, but the error is quite small at 0.02.

References

[1] Practical programming course book on interpolation.
PrakProg: Interpolation.pdf,
`http://86.52.112.181/~fedorov/numeric/book/interpolation.pdf`