Worksheet 5: Spline Interpolation Solutions

June 13, 2017

Author: Johannes Zeman

Task 5.1: Cubic and Quadratic Splines (6 points)

In this task, you will spline-interpolate the following functions on the specified domains:

Name	Definition	Domain
Sine Function	$f(x) = \sin x$	$[0, 2\pi]$
Runge Function	$g(x) = \frac{1}{1+x^2}$	$[-5, 5]$
Lennard-Jones Function	$h(x) = x^{-12} - x^{-6}$	$[1, 5]$

The python script `ws5.py` provides a demo which uses the class `scipy.interpolate.interp1d` to compute interpolating splines.

- **5.1.1** (2 points) Write a class `CubicSplineInterpolation` which implements cubic spline interpolation. As in the previous worksheet, the class shall provide a constructor `__init__(self, ...)` to initialize the interpolation and a method `__call__(self,x)` to compute the value of the interpolating function at `x`. On the boundaries, set the second derivative of the spline to zero. Use the class to create the same plots as the demo does. Note that the splines in this class will not be identical to the splines in the demo as SciPy uses different boundary conditions.

  Hint The constructor has to compute the spline coefficients by first generating the defining linear equations (see lecture notes pp. 35-36, especially the equation after equation 2.8) and then solving them using `scipy.linalg.solve`.

- **5.1.2** (2 points) Derive the equations for the *quadratic spline*, where the spline polynomial is defined as

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 \,. \tag{1}$$

  The conditions for the quadratic spline are:

  - It matches the function values at the supporting points x_j and x_{j+1}.
  - The first derivative of the spline at x_{j+1} is the same for S_j and S_{j+1}.
  - In your implementation, the first derivative should vanish at the lower boundary.

- **5.1.3** (2 points) Write a class `QuadraticSplineInterpolation` which implements the quadratic spline interpolation. Redo the same plots as in task 5.1.1 with quadratic splines.

Solution of Task 5.1.1 (Cubic Spline): Derivation of Defining Linear Equations

A cubic spline interpolation connecting n points $(x_j|y_j)$, $j \in [0, 1, ...n-1]$ can be expressed as splines $S_j(x)$ of the general form

$$S_j(x) = a_j + b_j\,(x - x_j) + c_j\,(x - x_j)^2 + d_j\,(x - x_j)^3\ , \quad j \in [0,\ 1,\ ...,\ n-1]\ , \tag{2}$$

where each segment $S_j(x)$ is the interpolating third-order polynomial on the interval $[x_j, x_{j+1}]$.

The corresponding first and second derivatives with respect to x are thus

$$\frac{d}{dx}S_j(x) = b_j + 2c_j\,(x - x_j) + 3d_j\,(x - x_j)^2 \tag{3}$$

$$\frac{d^2}{dx^2}S_j(x) = 2c_j + 6d_j\,(x - x_j)\ . \tag{4}$$

Each segment has to fulfill the following conditions:

i) $S_j(x_j) = y_j\,, \quad j \in [0,\ 1,\ ...,\ n-1]$

ii) $S_{j+1}(x_{j+1}) = S_j(x_{j+1})\,, \quad j \in [0,\ 1,\ ...,\ n-2]$

iii) $\frac{d}{dx}S_{j+1}(x_{j+1}) = \frac{d}{dx}S_j(x_{j+1})\,, \quad j \in [0,\ 1,\ ...,\ n-2]$

iv) $\frac{d^2}{dx^2}S_{j+1}(x_{j+1}) = \frac{d^2}{dx^2}S_j(x_{j+1})\,, \quad j \in [0,\ 1,\ ...,\ n-2]$

Additionally, it has to fulfill the boundary conditions

v) $\frac{d^2}{dx^2}S_0(x_0) = A$

vi) $\frac{d^2}{dx^2}S_{n-2}(x_{n-1}) = B\,.$

Here, we impose the so-called *natural* boundary conditions by requiring $A = B = 0$.

Using these conditions, the defining linear equations for the coefficients a_j, b_j, c_j, and d_j are derived in the following.

For the coefficients a_j, it follows from condition i) together with Eq. (2) that

$$a_j = y_j\,. \tag{5}$$

Applying condition iv) to Eq. (4) yields

$$\begin{aligned} 2c_{j+1} + 6d_{j+1}\underbrace{(x_{j+1} - x_{j+1})}_{=0} &= 2c_j + 6d_j\,(x_{j+1} - x_j) \\ \Rightarrow \quad d_j &= \frac{c_{j+1} - c_j}{3(x_{j+1} - x_j)}\,. \end{aligned} \tag{6}$$

Furthermore, applying conditions i) and ii) to Eq. (2) and inserting Eq. (5) and (6) yields

$$\begin{aligned} y_{j+1} &= y_j + b_j(x_{j+1} - x_j) + c_j(x_{j+1} - x_j)^2 + d_j(x_{j+1} - x_j)^3 \\ \Rightarrow b_j &= \frac{y_{j+1} - y_j}{x_{j+1} - x_j} - \frac{1}{3}(x_{j+1} - x_j)(2c_j + c_{j+1})\,. \end{aligned} \tag{7}$$

Analogously, by applying condition iii) on Eq. (3) and inserting Eq. (6), it follows that

$$b_{j+1} = b_j + (c_j + c_{j+1})(x_{j+1} - x_j)\,.$$

Plugging in Eq. (7) leads to

$$\frac{y_{j+2}-y_{j+1}}{x_{j+2}-x_{j+1}} = \frac{y_{j+1}-y_j}{x_{j+1}-x_j} + \frac{1}{3}\left(c_j(x_{j+1}-x_j) + 2c_{j+1}(x_{j+2}-x_j) + c_{j+2}(x_{j+2}-x_{j+1})\right)$$

Inserting Eq. (7), bringing all c-dependent terms to the left side, dividing by $(x_{j+2}-x_j)$ and shifting the index $j \to (j-1)$ yields the system of linear equations

$$\alpha_j c_{j-1} + 2c_j + \beta_j c_{j+1} = 3\gamma_j \tag{8}$$

with

$$\alpha_j := \frac{x_j - x_{j-1}}{x_{j+1}-x_{j-1}}\,, \quad \beta_j := \frac{x_{j+1}-x_j}{x_{j+1}-x_{j-1}}\,, \quad \text{and } \gamma_j := \frac{\frac{y_{j+1}-y_j}{x_{j+1}-x_j} - \frac{y_j-y_{j-1}}{x_j-x_{j-1}}}{x_{j+1}-x_{j-1}}\,.$$

In order to solve the equation system, we write Eq. (8) in matrix notation:

$$\begin{pmatrix} \alpha_1 & 2 & \beta_1 & & & 0 \\ & \alpha_2 & 2 & \beta_2 & & \\ & & \ddots & \ddots & \ddots & \\ 0 & & & \alpha_{n-2} & 2 & \beta_{n-2} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-2} \end{pmatrix} = \begin{pmatrix} 3\gamma_1 \\ 3\gamma_2 \\ \vdots \\ 3\gamma_{n-2} \end{pmatrix} \tag{9}$$

Yet, there are two equations missing in order to solve Eq. (9), which are the boundary conditions v) and vi). With natural boundary conditions $A = B = 0$, applying these conditions to Eq. (4) together with Eq. (6) yields

$$c_0 = c_{n-1} = 0\,.$$

Adding these equations to Eq. (9) finally leads to the complete system of linear equations:

$$\begin{pmatrix} 1 & 0 & & & & 0 \\ \alpha_1 & 2 & \beta_1 & & & \\ & \alpha_2 & 2 & \beta_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \alpha_{n-2} & 2 & \beta_{n-2} \\ 0 & & & & 0 & 1 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-2} \\ c_{n-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 3\gamma_1 \\ 3\gamma_2 \\ \vdots \\ 3\gamma_{n-2} \\ 0 \end{pmatrix} \tag{10}$$

Once the system of linear equations is solved, the coefficients b_j and d_j are obtained from Eq. (7) and (6), respectively.

For the implementation of the class `CubicSplineInterpolation`, see the Python script `ws5_solution.py`.

Solution of Task 5.1.2 (Quadratic Spline): Derivation of Defining Linear Equations

A quadratic spline interpolation connecting n points $(x_j|y_j)$, $j \in [0, 1, ...n-1]$ can be expressed as splines $S_j(x)$ of the general form

$$S_j(x) = a_j + b_j\,(x - x_j) + c_j\,(x - x_j)^2\,, \quad j \in [0,\,1,\,...,\,n-1]\,, \tag{11}$$

where each segment $S_j(x)$ is the interpolating second-order polynomial on the interval $[x_j, x_{j+1}]$.

The corresponding first derivatives with respect to x are thus

$$\frac{d}{dx}S_j(x) = b_j + 2c_j\,(x - x_j)\,. \tag{12}$$

Each segment has to fulfill the following conditions:

i) $S_j(x_j) = y_j\,, \quad j \in [0,\,1,\,...,\,n-1]$

ii) $S_{j+1}(x_{j+1}) = S_j(x_{j+1})\,, \quad j \in [0,\,1,\,...,\,n-2]$

iii) $\frac{d}{dx}S_{j+1}(x_{j+1}) = \frac{d}{dx}S_j(x_{j+1})\,, \quad j \in [0,\,1,\,...,\,n-2]$

Additionally, it has to fulfill the boundary condition

iv) $\frac{d}{dx}S_0(x_0) = A\,.$

We impose natural boundary conditions by setting $A = 0$.

Using these conditions, the defining linear equations for the coefficients a_j, b_j, and c_j are derived in the following.

For the coefficients a_j, it follows from condition i) together with Eq. (11) that

$$a_j = y_j\,. \tag{13}$$

Applying condition iii) to Eq. (12) yields

$$\begin{aligned} b_{j+1} + 2c_{j+1}\underbrace{(x_{j+1} - x_{j+1})}_{=0} &= b_j + 2c_j\,(x_{j+1} - x_j) \\ \Rightarrow \quad c_j &= \frac{b_{j+1} - b_j}{2(x_{j+1} - x_j)}\,. \end{aligned} \tag{14}$$

Furthermore, applying conditions i) and ii) to Eq. (11) and inserting Eq. (13) and (14) yields

$$\begin{aligned} y_{j+1} &= y_j + b_j(x_{j+1} - x_j) + c_j(x_{j+1} - x_j)^2 \\ \Rightarrow b_j + b_{j+1} &= 2\frac{y_{j+1} - y_j}{x_{j+1} - x_j}\,. \end{aligned} \tag{15}$$

With an index shift $j \to (j-1)$ we write this as

$$b_{j-1} + b_j = 2\gamma_j \tag{16}$$

with

$$\gamma_j := \frac{y_j - y_{j-1}}{x_j - x_{j-1}}\,.$$

In order to solve the equation system, we write Eq. (16) in matrix notation:

$$\begin{pmatrix} 1 & 1 & & & 0 \\ & 1 & 1 & & \\ & & \ddots & \ddots & \\ 0 & & & 1 & 1 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \end{pmatrix} = \begin{pmatrix} 2\gamma_1 \\ 2\gamma_2 \\ \vdots \\ 2\gamma_{n-1} \end{pmatrix} \tag{17}$$

Yet, there is one equation missing in order to solve Eq. (17), which is the boundary condition iv). With the natural boundary condition $A = 0$, applying this condition to Eq. (12) yields

$$b_0 = 0\,.$$

Adding this equation to Eq. (17) finally leads to the complete system of linear equations:

$$\begin{pmatrix} 1 & 0 & & & 0 \\ 1 & 1 & & & \\ & 1 & 1 & & \\ & & \ddots & \ddots & \\ 0 & & & 1 & 1 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{n-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 2\gamma_1 \\ 2\gamma_2 \\ \vdots \\ 2\gamma_{n-1} \end{pmatrix} \tag{18}$$