

Worksheet 5: Spline Interpolation

May 24, 2017

General Remarks

- The deadline for handing in the worksheets is **Monday, May 29th, 2017, 12:00 noon**.
- This is a short worksheet for which you can achieve a maximum of 6 points.
- To hand in your solutions, send an email to your tutor:
 - Johannes Zeman zeman@icp.uni-stuttgart.de (Tue 15:45–17:15)
 - Michael Kuron mkuron@icp.uni-stuttgart.de (Wed 15:45–17:15)
 - Kai Szuttor kai@icp.uni-stuttgart.de (Thu 14:00–15:30)
- Please try to only hand in a single file that contains your program code for all tasks. If you are asked to answer questions, you should do so in a comment in your code file. If you are asked for graphs or figures, it is sufficient if your code generates them. You may as well hand in a separate PDF document with all your answers, graphs and equations.
- The worksheets are to be solved in groups of two or three people.

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

$$P_i(x) = y_i + m_i(x - x_i) + M_i(x - x_i)^2. \quad (1)$$

The conditions for the quadratic spline are:

- It matches the function values at the supporting points x_i and x_{i+1} .
 - The first derivative of the spline at x_{i+1} is the same for P_i and P_{i+1} .
 - In your implementation, the first derivative should vanish at the boundaries.
- **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.