In-class Exercise 4.2
Polynomial fits and splines
AMATH 301
University of Washington
Jakob Kotas

1. Given the dataset:

x	0	2	4	6
y	9	-17	-3	3

   (a) Plot the data.

   (b) Find the interpolating polynomial by solving a linear system of equations.

$$\begin{aligned}
f(x_0) &= a_3x_0^3 + a_2x_0^2 + a_1x_0 + a_0 &= y_0 \\
f(x_1) &= a_3x_1^3 + a_2x_1^2 + a_1x_1 + a_0 &= y_1 \\
f(x_2) &= a_3x_2^3 + a_2x_2^2 + a_1x_2 + a_0 &= y_2 \\
f(x_3) &= a_3x_3^3 + a_2x_3^2 + a_1x_3 + a_0 &= y_3
\end{aligned}$$

$$\underbrace{\begin{bmatrix} 1 & x_0 & x_0^2 & x_0^3 \\ 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 1 & x_3 & x_3^2 & x_3^3 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}}_{a} = \underbrace{\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}}_{y}$$

   (c) Find the interpolating polynomial by using the Lagrange form.

$$L_{n,k} = \prod_{0\le m\le n, m\ne k} \frac{x - x_m}{x_k - x_m}$$

$$p_n(x) = \sum_{k=0}^{n} y_k L_{n,k}(x)$$

$$L_{3,0}(x) = \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} = \frac{(x-2)(x-4)(x-6)}{(0-2)(0-4)(0-6)} = \frac{(x-2)(x-4)(x-6)}{-48}$$

$$L_{3,1}(x) = \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} = \frac{(x-0)(x-4)(x-6)}{(2-0)(2-4)(2-6)} = \frac{x(x-4)(x-6)}{16}$$

$$L_{3,2}(x) = \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} = \frac{(x-0)(x-2)(x-6)}{(4-0)(4-2)(4-6)} = \frac{x(x-2)(x-6)}{-16}$$

$$L_{3,3}(x) = \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} = \frac{(x-0)(x-2)(x-4)}{(6-0)(6-2)(6-4)} = \frac{x(x-2)(x-4)}{48}$$

$$\begin{aligned}
p_3(x) = 9\left(\frac{(x-2)(x-4)(x-6)}{-48}\right) - 17\left(\frac{x(x-4)(x-6)}{16}\right) \\
-3\left(\frac{x(x-2)(x-6)}{-16}\right) + 3\left(\frac{x(x-2)(x-4)}{48}\right) \\
= -x^3 + 11x^2 - 31x + 9
\end{aligned}$$

(d) Plot the interpolating polynomial on the same axes as the dataset.

(e) Compare the result to the built-in function `KroghInterpolator`. You'll need `from scipy.interpolate import KroghInterpolator`.

2. Given the dataset:

x	0	1	2	3	4
y	1	3	0	1	-2

(a) Plot the data.

(b) The spline interpolation is defined as:

$$S_k(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3, \qquad x_k \le x \le x_{k+1}$$

under the conditions:

$$\begin{gathered} S_k(x_k) = S_{k+1}(x_k) = y_k \quad \text{for all} \quad k = 0, 1, 2, \cdots, n \\ S'_k(x_k) = S'_{k+1}(x_k) \quad \text{for all} \quad k = 1, 2, \cdots, n-1 \\ S''_k(x_k) = S''_{k+1}(x_k) \quad \text{for all} \quad k = 1, 2, \cdots, n-1 \\ S''(x_0) = S''(x_n) = 0 \end{gathered}$$

Find the interpolating spline by solving a linear system of equations.

$$\begin{gathered} S_0(x) = a_0 + b_0 x + c_0 x^2 + d_0 x^3, \qquad 0 \le x < 1 \\ S_1(x) = a_1 + b_1(x-1) + c_1(x-1)^2 + d_1(x-1)^3, \qquad 1 \le x < 2 \\ S_2(x) = a_2 + b_2(x-2) + c_2(x-2)^2 + d_2(x-2)^3, \qquad 2 \le x < 3 \\ S_3(x) = a_3 + b_3(x-3) + c_3(x-3)^2 + d_3(x-3)^3, \qquad 3 \le x \le 4 \end{gathered}$$

$$\underbrace{\left[\begin{array}{cccc|cccc|cccc|cccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\hline
0 & 1 & 2 & 3 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & -1 & 0 & 0 \\
\hline
0 & 0 & 2 & 6 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & 6 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 6 & 0 & 0 & -2 & 0 \\
\hline
0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 6
\end{array}\right]}_{M}
\underbrace{\left[\begin{array}{c} a_0 \\ b_0 \\ c_0 \\ d_0 \\ \hline a_1 \\ b_1 \\ c_1 \\ d_1 \\ \hline a_2 \\ b_2 \\ c_2 \\ d_2 \\ \hline a_3 \\ b_3 \\ c_3 \\ d_3 \end{array}\right]}_{c}
=
\underbrace{\left[\begin{array}{c} 1 \\ 3 \\ 3 \\ 0 \\ 0 \\ 1 \\ 1 \\ -2 \\ \hline 0 \\ 0 \\ 0 \\ \hline 0 \\ 0 \\ 0 \\ \hline 0 \\ 0 \end{array}\right]}_{b}$$

(c) Compare the result to the built-in function `CubicSpline`. You'll need `from scipy.interpolate import CubicSpline`.

(d) Also plot the interpolating polynomial on the same axes and compare the two.

3. Generate a dataset where $x = -10, -9, -8, \cdots, 10$ and y are randomly sampled from a standard normal distribution. Plot the interpolating polynomial (using `KroghInterpolator`) and interpolating spline (using `CubicSpline`). Compare the results.