MATINF4160 Fall 2016
Mandatory Assignment 3

Ivar Haugaløkken Stangeby

Due: November 17th, 2016

Problem 1

We wish to find control points $\mathbf{c}_0, \dots, \mathbf{c}_5$ such that the Bézier curve

$$\mathbf{p}(t) = \sum_{j=0}^{5} \mathbf{c}_j B_j^n(u)$$

interpolates a given function $\mathbf{f}$ and its ith derivatives in the points a and b for $i = 0, 1, 2$. That is, we want $\mathbf{p}^{(i)}(a) = \mathbf{f}^{(i)}(a)$ for $i = 0, 1, 2$. We have in general that the control points $\mathbf{c}_j$ can be found by the following two formulas:

$$\mathbf{c}_j = \sum_{j=0}^{i} \binom{i}{j} \mathbf{b}_j, \qquad \mathbf{c}_{n-i} = \sum_{j=0}^{i} (-1)^j \binom{i}{j} \mathbf{b}_{n-j}$$

where we define

$$\mathbf{b}_i \coloneqq \frac{(n-i)!}{n!}(b-a)^i \mathbf{f}^{(i)}(a),$$

for the leftmost control points, and define

$$\mathbf{b}_{n-i} \coloneqq \frac{(n-i)!}{n!}(b-a)^i \mathbf{f}^{(i)}(b).$$

for the rightmost control points. We start by computing the $\mathbf{b}_i$ and $\mathbf{b}_{n-1}$. This yields

$$\begin{aligned} \mathbf{b}_0 &= f(a) & \mathbf{b}_1 &= \frac{1}{5}(b-a)\mathbf{f}'(a) \\ \mathbf{b}_2 &= \frac{1}{20}(b-a)^2\mathbf{f}''(a) & \mathbf{b}_3 &= \frac{1}{20}(b-a)^2\mathbf{f}''(b) \\ \mathbf{b}_4 &= \frac{1}{5}(b-a)\mathbf{f}'(b) & \mathbf{b}_5 &= \mathbf{f}(b) \end{aligned}$$

It then follows that the control points are given by

$$\begin{aligned} \mathbf{c}_0 &= \mathbf{b}_0 & \mathbf{c}_1 &= \mathbf{b}_0 + \mathbf{b}_1 \\ \mathbf{c}_2 &= \mathbf{b}_0 + 2\mathbf{b}_1 + \mathbf{b}_2 & \mathbf{c}_3 &= \mathbf{b}_5 - 2\mathbf{b}_4 + \mathbf{b}_3 \\ \mathbf{c}_4 &= \mathbf{b}_5 - \mathbf{b}_4 & \mathbf{c}_5 &= \mathbf{b}_5. \end{aligned}$$

Problem 2

In this problem we consider an implementation of a C^2 cubic spline curve interpolation. We interpolate two data sets, $\mathbf{x}_i = (x_i, y_i)$ with

$$(x_0, \ldots, x_9) = (261, 261, 283, 287, 280, 281, 302, 319, 335, 277)$$
$$(y_0, \ldots, y_9) = (703, 738, 718, 723, 735, 736, 731, 748, 737, 682)$$

and $\mathbf{x}_i = (\cos(s_i), \sin(s_i))$ where

$$(s_0, \ldots, s_{12}) = (0, 0.24, 0.39, 0.78, 1.03, 1.18, 1.56, 1.81, 1.96, 1.96, 2.34, 2.59, 2.74, 3.12).$$

The general idea when looking for a Hermite spline interpolation is finding the suitable knot vector, where we want sufficient resolution where the $\mathbf{x}_i$ varies a lot. To each pair of data points we fit a cubic Bézier curve with control points tweaked for C^2 continuity. In this assignment we consider the parametrization

$$t_{i+1} - t_i = \|\mathbf{x}_{i+1} - \mathbf{x}_i\|^{\mu},$$

for $\mu = 0, 0.5, 1$ and examine how it affects the interpolation. The interpolations can be computed and visualized with the following code:

Interpolating datasets

```python
from hermite_interpolation import *
import numpy as np

x_values = np.array([(261, 703), (261, 738),\
    (283, 718), (287, 723), (280, 735), \
    (281, 736), (302, 731), (319, 748),\
    (335, 737), (277, 682)])

for i, mu in enumerate([0, 0.5, 1]):
    curve = HermiteInterpolation(x_values,\
        order_of_continuity=2,\
        label='$\\mu = %0.2f$' % mu, mu=mu)
    curve.plot(display=True, filename='one_%d.pdf' % i)
```

Listing 1: spline_interpolation.py

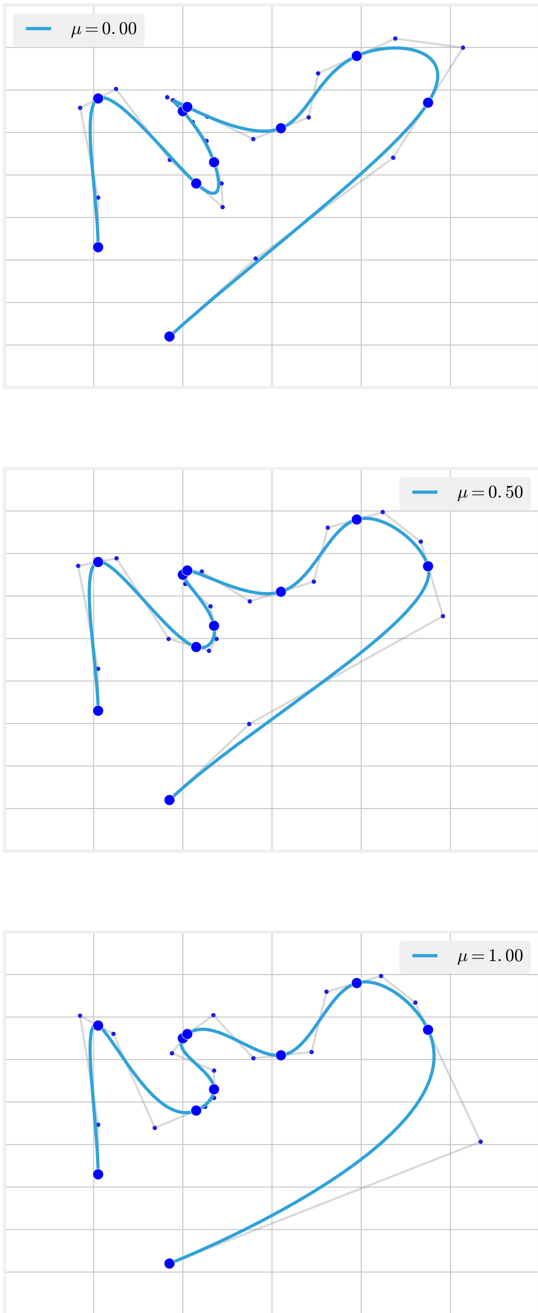

Figure 1: The first dataset visualized for three different values of μ. The interpolating curve is visualized in light blue, the large dark blue dots are the original data points.

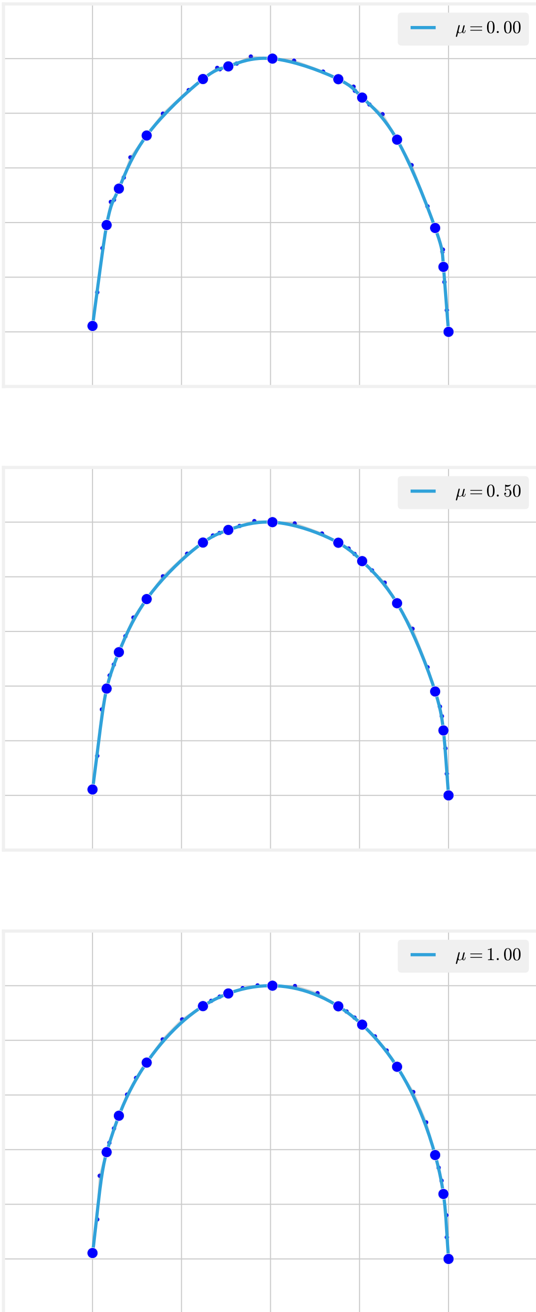

Figure 2: The second dataset visualized for three different values of μ. The interpolating curve is visualized in light blue, the large dark blue dots are the original data points.