# MAT-INF4170 Oblig 4

Daniel Osen

May 3, 2017

## Problem 1

My implementation seems to have some bugs, which I cannot figure out what are. I do not understand why, but both the least-squares solution, i.e $A^T\boldsymbol{C} \approx [\boldsymbol{x}, \boldsymbol{y}]$ and the evaluations from Algorithm 2.20 seem to produce points at the origin $(0, 0, z)$ which destroys the curve. Moreover, the matrix $A^TA$ is not always singular when this happens, and there are no coefficients at this origin then. In the plots shown I have used the $p + 1$ regular knot vector with identical knots at the end, though otherwise it is a uniform knot vector on the interval $u_2 = a < ... < b = u_{m-1}$, though I have also attempted wrap the knots and control points and otherwise just used a uniform knot vector without $p + 1$ identical knots at the ends.

$$\boldsymbol{\tau} = (u_1, u_1, u_1, u_1, a, ..., b, u_m, u_m, u_m, u_m).$$

The curves produced for the data is shown below. They are closed only in the sense that I have forced $C^0$ continuity. In the case of about 20% of the length the matrix was singular for two of the cross-sections, and I resorted to numpy's built-in least squares method. The number of evaluations is 1000 for each curve.

Figure 1: 5% Cubic Spline Least Squares

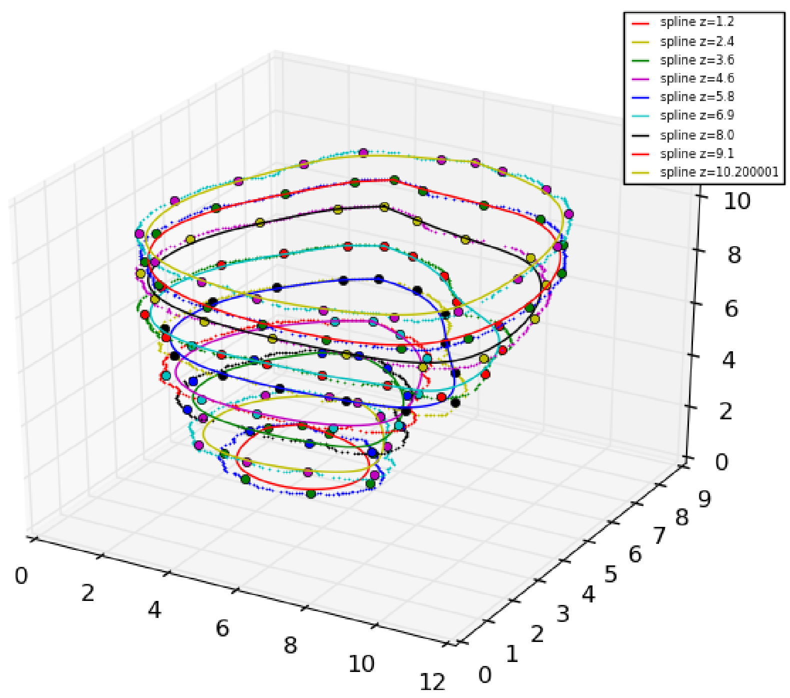


Figure 2: 10% Cubic Spline Least Squares

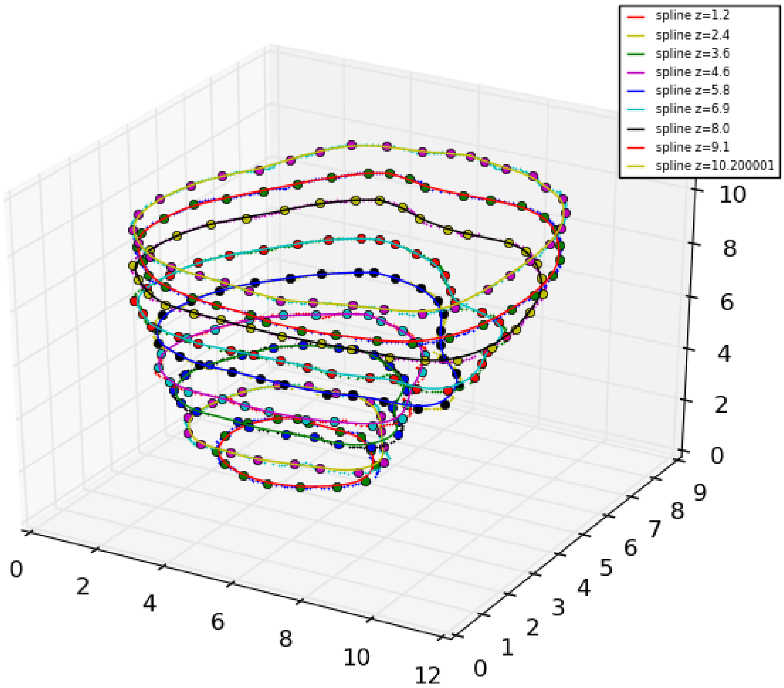
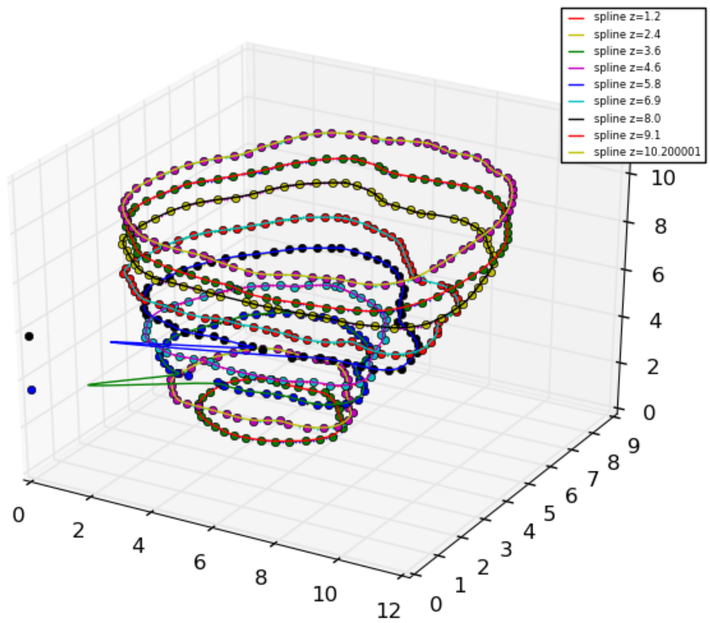



Figure 3: 20% Cubic Spline Least Squares



## Problem 2

I wasn't able to complete this problem in time. I will try to finish it this week.