Assignment 4 for MAT4170
Spline methods, Spring 2022

To be completed by Tuesday 10 May. Please send your solution as a single pdf file (including plots/figures) to `michaelf@math.uio.no`.

Problem 1. Modelling cross sections of a heart. In the data directory there are nine files that contain data taken from contours of different cross sections of a heart, see Figure 1. The file `hj1.dat` is the smallest one and this cross section is taken from the bottom of the heart while the file `hj9.dat` is the largest one with a cross section taken from the top of the heart. The data were obtained by cutting a heart into thin slices and photographing each slice. The contours were then obtained via image processing techniques. Every contour is in the form $(x_i, y_i, z_i)_{i=1}^n$ where the number of points n varies between the cross sections, and z_i is constant for each cross section. Every cross section corresponds to a closed curve so the first point is identical to the last point. In this problem you are going to determine spline curve approximations to the cross sections. Before we can approximate the data with spline curves, we must determine a parameter value for each point. In other words, we want to augment the data set to $(u_i, x_i, y_i, z_i)_{i=1}^n$ where $(u_i)_{i=1}^n$ is an increasing sequence of numbers with $u_1 = 0$. In this problem we will use chord length parameterization, i.e., we use

$$u_i = u_{i-1} + \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2}$$

for $i = 2, \ldots, n$. The next step is to approximate the data by a cubic spline by using least squares approximation. For this, you need to choose a suitable knot vector. Experiment by letting the length of the knot vector be 5%, 10% and 20% of the number of data points in the cross section and distribute the knots uniformly. From earlier assignments you should have routines available for computing points on, and plotting, spline functions. These must be adapted to computing points on, and plotting, spline curves. Present your results in three 3D plots that each show all the cross sectional curves for a certain length of the knot vector.

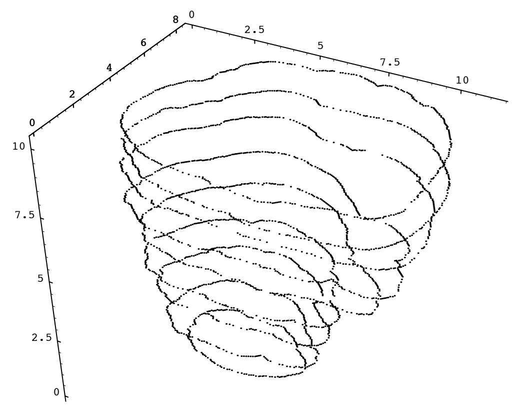

Figure 1: Data sets - closed contours sampled from a heart model

Problem 2. Tensor-product spline representation of the heart data. This problem is a continuation of Problem 1. The aim is to compute a parametric tensor-product spline surface with the property that all of the nine given data sets are close to the surface. The construction is based on the theory in Chapter 7 of the lecture notes. In order to efficiently determine a tensor-product spline surface, we need the data to be rectangular, i.e., the data must be in the form $(x_{ij}, y_{ij}, z_{ij})_{i,j=1}^{m,n}$ for suitable m and n. From this we can compute a parametric tensor-product spline surface $\mathbf{f}(u, v)$ where the u variable runs along the i index and the v variable runs along the j index.

The solution of the problem can naturally be divided into three parts:

(a) Generate a rectangular data set.

(b) Find parameter values for the data set and compute a spline surface.

(c) Plot the surface.

Let us consider each of these steps in turn.

(a) In Problem 1 you approximated the data with spline curves with knot vectors of different lengths. We will now take one of the sets of curves as a starting point, namely the one that you think best fits the nine data sets. In other words, we start with nine spline curves that provide good representations of the nine cross sections of the heart. By sampling each of the nine curves at 20 points we obtain a data set $(x_{ij}, y_{ij}, z_{ij})_{i,j=1}^{20,9}$. Here the i-index runs along the curves while the j-index runs across the curves. The question is which 20 values to pick along each curve. We will use a simple (but not always very good) strategy: we pick values with uniform parameter distance. If the j-th curve is $\mathbf{g}_j(u)$ and the parameter u varies in the interval $[0, U_j]$, we choose $(x_{ij}, y_{ij}) = \mathbf{g}_j(iU_j/19)$ for $i = 0, 1, \ldots, 19$. The third component z_{ij} is constant for curve $\mathbf{g}_j$ and explicitly known. The idea behind this is that if we now fix i, then $(x_{ij}, y_{ij}, z_{ij})_{j=1}^{9}$ will be points on the nine cross-sectional curves that should be naturally linked by a curve across these curves. Generate the 180 points as described above.

(b) In order to compute a tensor-product spline surface we need two knot vectors. Because of the way in which we obtained the data, it is natural to use uniform parameterization so that the point (x_{ij}, y_{ij}, z_{ij}) is assigned the parameter value $(u_i, v_j) = (i/19, j/8)$. This will give you enough data to compute the three components of a parametric, bicubic spline surface with the least squares method, as explained in Section 7.2 in the notes. Use a uniform 4-regular knot vector with 7 interior knots in the u direction and a uniform 4-regular knot vector with 1 interior knot in the v direction. Determine the spline surface.

(c) Using standard spline procedures you can compute points on the spline surface that you determined in (b), as shown in Section 7.1 of the lecture notes. Do this (try with a uniform 30×30 grid) and plot the result.

Extra challenge (optional): The data is sampled from closed curves. Make the curves and surfaces closed and C^2 continuous by either posing appropriate boundary conditions, or making the parameter domain periodic.