UNIVERSITÄT BERN

Introduction to Computer Graphics
Assignment 10 – Freeform curves

Handout date: 29.11.2019

Submission deadline: 06.12.2019 12:00

Late submissions are not accepted

This assignment is a pure "pencil and paper" exercise on Bézier curves. In the first two tasks, you are asked to find the control points of the Bézier curves to reproduce given functions; in the third task, you will compute a single point on a given Bézier curve using de Calsteljau algorithm; in the forth task, you will subdivide a given Bézier curve into two pieces; and in the fifth task, you will look at the maxima of Bernstein polynomials.

Task 1: Control points 1

Let $f(x) = \sqrt{x}$ in the interval $[0, 1]$, as plotted in Figure 1. Represent the graph of this function as a Bézier curve of lowest possible degree. What is the degree of this Bézier curve? Compute the exact coordinates of all the control points and draw them in the image.

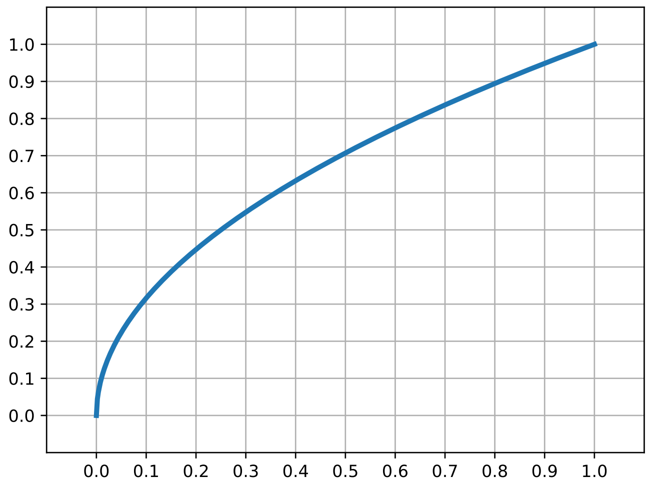

Figure 1: $y = \sqrt{x}$.

Task 2: Control points 2

Let $f(x) = \sqrt[3]{x^2}$ in the interval $[0, 1]$, as plotted in Figure 2. Represent the graph of this function as a Bézier curve of lowest possible degree. What is the degree of this Bézier curve? Compute the exact coordinates of all the control points and draw them in the image.

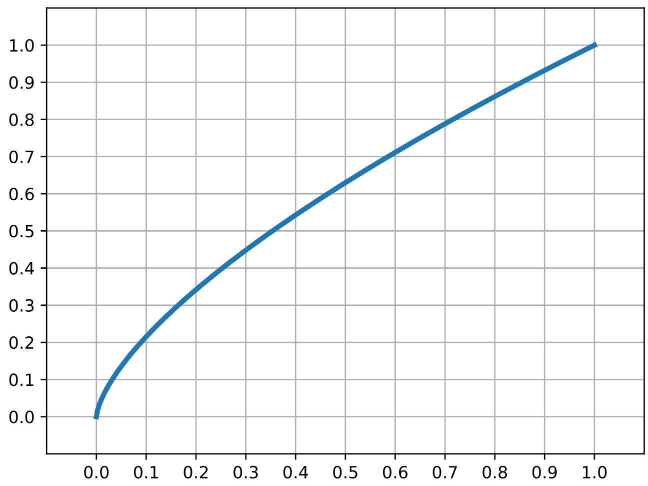

Figure 2: $y = \sqrt[3]{x^2}$

Task 3: The de Casteljau algorithm

You are given a set of 5 control points $\mathbf{b}_0 = \begin{bmatrix}1\\1\end{bmatrix}$, $\mathbf{b}_1 = \begin{bmatrix}3\\5\end{bmatrix}$, $\mathbf{b}_2 = \begin{bmatrix}6\\3\end{bmatrix}$, $\mathbf{b}_3 = \begin{bmatrix}7\\9\end{bmatrix}$, $\mathbf{b}_4 = \begin{bmatrix}10\\4\end{bmatrix}$. Using the de Casteljau algorithm, find the coordinates of the point $\mathbf{c}(t)$ on the corresponding Bézier curve at $t = 0.25$. Do this in two ways:

- **Arithmetically:** Give the intermediate values $\mathbf{b}_i^k$ after each recursion step.
- **Graphically:** Draw the points into a coordinate system, use a ruler to perform the affine interpolation steps. Label the points with the corresponding intermediate variables $\mathbf{b}_i^k$.

Task 4: Maxima of Bernstein polynomials

Prove that Bernstein polynomial B_i^n reaches its maximum at $t = i/n$. Hint: the derivative of a Bernstein polynomial is given by the expression $\frac{\mathrm{d}}{\mathrm{d}t} B_i^n(t) = n(B_{i-1}^{n-1}(t) - B_i^{n-1}(t))$.

What does this result mean in the context of Bézier curves?

Task 5: Subdivision

You are given a set of 4 control points $\mathbf{b}_0 = \begin{bmatrix}2\\2\end{bmatrix}$, $\mathbf{b}_1 = \begin{bmatrix}2\\9\end{bmatrix}$, $\mathbf{b}_2 = \begin{bmatrix}7\\2\end{bmatrix}$, $\mathbf{b}_3 = \begin{bmatrix}7\\9\end{bmatrix}$ defining the Bézier curve $\mathbf{c}(t)$ of order $n = 3$ and are asked to subdivide this curve at $t_s = 0.25$ into two Bézier curves $\mathbf{c}_l(t)$ and $\mathbf{c}_r(t)$ of the same order $n = 3$. Do this by computing the control points of each sub-curve such that evaluating $\mathbf{c}_l$ on the interval $[0, 1]$ traces out the same curve as evaluating $\mathbf{c}$ on $[0, t_s]$, and similarly $\mathbf{c}_r$ on $[0, 1]$ coincides with $\mathbf{c}$ on $[t_s, 1]$.

An example of how the Bézier curve is subdivided into two for the parameter $t_s = 0.6$ is shown in Figure 3.

We can find the control points of the new curves using the **blossoming principle**. You can think of it as

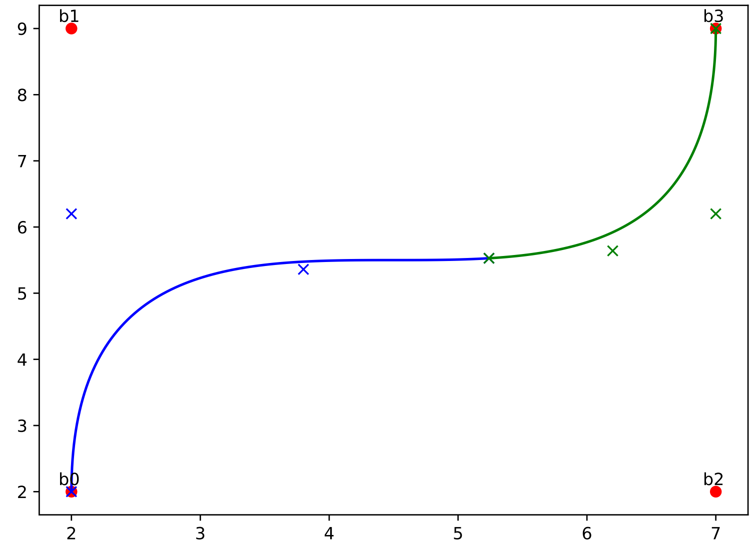

Figure 3: Subdivision at $t_s = 0.6$. The original control points are denoted by red dots, the new control points of each subcurve are denoted by blue and green crosses.

the adjusted de Casteljau algorithm. In the traditional de Casteljau algorithm, each recursion step k uses the same parameter t, i.e.

$$\mathbf{b}_i^k = (1-t)\mathbf{b}_i^{k-1} + t\mathbf{b}_{i+1}^{k-1}.$$

Blossoming is essentially the same procedure, except a distinct parameter t_k can be used in each recursion step:

$$\mathbf{b}_i^k = (1-t_k)\mathbf{b}_i^{k-1} + t_k\mathbf{b}_{i+1}^{k-1}.$$

It turns out that feeding a suitable combination of coefficients t_k into the blossoming algorithm can extract the control point coordinates of a Bézier curve (or its subdivided curves).

Let us denote by $dcj_P(t) : \mathbb{R} \to \mathbb{R}^2$ the de Casteljau algorithm — the function evaluating the Bézier curve for given control points $P \in \mathbb{R}^{(n+1)\times 2}$ at the scalar input parameter t. Similarly, we will denote the blossoming algorithm as function $blossom_P(\mathbf{t}) : \mathbb{R}^k \to \mathbb{R}^2$, where $\mathbf{t}$ is the k-dimensional vector of coefficients t. Furthermore, the blossoming can be used not only on the interval $[0,1]$ but on a general interval $[a,b]$.

The following examples should provide a notion of how blossoming behaves in the case of our cubic Bézier curve and how it can be used to extract control point locations:

$$\begin{aligned} blossom_P([t,t,t]) &= dcj_P(t) \\ blossom_P([0,0,0]) &= dcj_P(0) = \mathbf{b}_0 \\ blossom_P([0,0,1]) &= \mathbf{b}_1 \end{aligned}$$

Notice that if you choose the control points P_l such that the following condition holds for all t_0, t_1, t_2,

$$blossom_{P_l}([t_0,t_1,t_2]) = blossom_P(t_s * [t_0,t_1,t_2]),$$

then these control points define a Bézier curve that, when evaluated on $[0,1]$, traces out the $[0,t_s]$ segment of the curve defined by control points P.

Grading

The scores for this assignment are broken down as follows:

1. Control points 1: 15%
2. Control points 2: 15%
3. The de Casteljau algorithm: 20%
4. Maxima of Bernstein polynomials: 20%
5. Subdivision: 30%

What to hand in

Electronic version of the document (.pdf), either created electronically or hand-written and scanned, and a `readme.txt` file with each team member's full name, and brief comments of difficulties encountered.