Advanced Topics in Computer Graphics II

Universität Bonn
Institut für Informatik II
14.10.2025

Winter term 2025
Prof. Dr. Reinhard Klein
Domenic Zingsheim
zingsheim@cs.uni-bonn.de

Sheet G01 - Bézier Curves

Solutions for the theoretical and practical part via eCampus by Su, 19.10.2025, **23:59**.

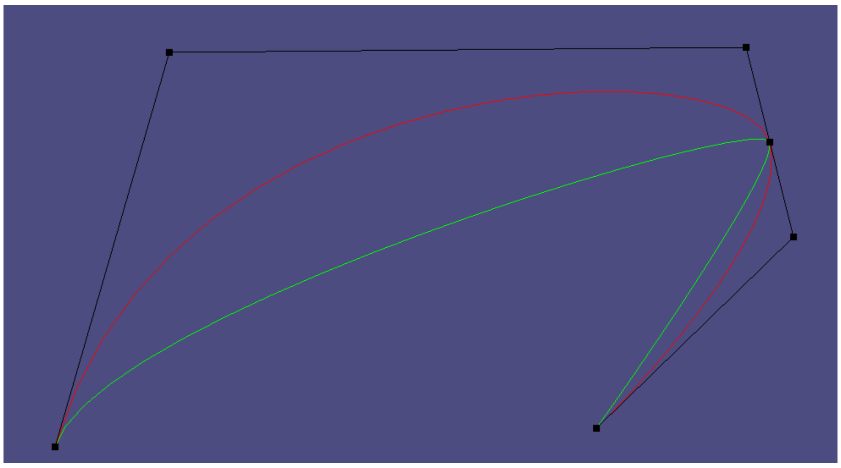
Figure 1: Result of the Bezier (red) and Hermite curve (green).

Practical Part

Assignment 1) *(0Pts)*

First, we will setup the exercise framework to make sure it works for everyone. Download the file `atcg_framework.zip` from eCampus and unpack it.

To build the project, follow the instructions on the README.md file. If everything worked, you should be able to execute: `./bin/<config>/exercise00Intro` where `<config>` is either `Release` or `Debug`, or (depending on your system) empty (`./bin/exercise00Intro`). A window that renders the suzanne mesh should open. You can

- **Press Left Mouse Button + Drag** to rotate the mesh.
- **Press Right Mouse Button + Drag** to move the mesh.
- **Drag and Drop other mesh files** from the `res` folder to open another mesh.
- **Scroll** to zoom.
- **Use the settings menu** to only display edge or vertex information.

Make yourself familiar with the used libraries, especially OpenMesh.

Look into `exercises/exercise00Intro/source.cpp` and understand the structure of the framework for the next exercises! If you have any questions or problems building the framework, please contact me!

Assignment 2) *(2Pts)*

Download the archive `framework_g01.zip` and unpack it into the corresponding folder inside your `atcg_framework` (exercises should go into the `exercise` folder and shader into `shader`). From the root directory of the project, run `cmake build` to add the new project to the build settings. After that, you can build the project like usual (for example `cmake --build build`). This should create an exercise01Bezier executable inside your `bin` folder. If you run this file you should see the viewer from last time but without any geometry (yet). You can now draw a control polygon by adding points via **Shift + Left Click**. Your task will be to implement different types of curves into these control polygons. If you have any problems building the framework, please let me know!

Inside `shader/{bezier,hermite}.gs` you will find the exercises for this task.

- Please implement the basis functions for a cubic Bezier curve.
- Please implement the basis functions for a cubic Hermite curve.

Submit this assignment by uploading both shader files onto eCampus!.
If you implemented everything correctly, the results should look like in Figure 1. You can control the render settings inside the sub-menu found on the menu bar.

Assignment 3) *(2Pts)*

This exercise is located inside `exercises/exercise01Bezier/source.cpp`. Inside the function `onMousePressed()`, add a point to the control polygon such that the resulting curve will be $\mathcal{C}_1$-continuous.
Submit this assignment by uploading the source file onto eCampus!

Theoretical Part

Please hand in a sheet-G01-*lastname*.pdf sheet written in latex via eCampus!

Assignment 4) *(4Pts)*

We saw in the lecture that we can represent Bezier Curves in matrix form using the polynomial monomials, a coefficient matrix M and a control points matrix P.

$$p(t) = \begin{bmatrix} 1 & t & t^2 \end{bmatrix} \cdot M \cdot P$$

a) Please derive this representation for *quadratic* Bézier curves.

Splitting a curve into two parts at a point can easily be done using the matrix representation. We want to split a quadratic Bézier curve at a point $t = z$ by first separating out the actual point on the curve:

$$p(t) = \begin{bmatrix} 1 & (z \cdot t) & (z \cdot t)^2 \end{bmatrix} \cdot M \cdot P = \begin{bmatrix} 1 & t & t^2 \end{bmatrix} \cdot Z \cdot M \cdot P$$

where M is the coefficient matrix from the previous exercise and P is the vector of control points.

b) Please derive the matrix representation of Z.

c) We need make the following transformation to get from this representation to the standard matrix form of a Bézier curve:

$$\begin{bmatrix} 1 & t & t^2 \end{bmatrix} \cdot Z \cdot M \cdot P = \begin{bmatrix} 1 & t & t^2 \end{bmatrix} \cdot M \cdot Q \cdot P$$

Please derive the matrix representation of the matrix Q.

Hint: $M \cdot M^{-1} = Id$

This gives us the subcurve from $t = 0$ to $t = z$. How can we get the second subcurve? The approach is similar. The second curve should go from $t = z$ to $t = 1$.

d) Please derive the matrix representation of the second curve. Explicitly state the Q matrix.

Hint: Reparameterize $t = z + (1 - z) \cdot t$

Assignment 5) *(6Pts)*

Given three control points on the xy-plane $(-1, 0)$, $(0, 1)$ and $(2, 0)$, do the following:

a) Write down its Bézier curve equation.

b) Expand this equation to its equivalent conventional form.

c) Since there are three control points, there are three Bézier coefficients. Write down their equations and sketch their graphs.

d) Use your calculator to find enough number of points using the conventional parametric form and sketch the curve.

e) Find points on the curve that correspond to $u = 0, 0.25, 0.5, 0.75$ and 1 with the conventional form.

f) Use de Casteljau's algorithm to find points on the curve corresponding to $u = 0, 0.25, 0.5, 0.75$ and 1.

g) Subdivide the Bézier curve at $u = 0.4$ and list the control points of the resulting curve segments.

h) Increase the degree of this curve to three and list the new set of control points. Then, increase the degree to four and list the new set of control points.

Good luck!