# Computergraphik 2 - Exercise Sheet 2

Technische Universität Berlin - Computer Graphics

**Handout** 11th May 2023, **Due** 31th May 2023

Markus Worchel, Lars Munaf, Ugo Finnendahl, Prof. Dr. Marc Alexa

## Exercise 1: Approximation of height fields (5 + 1 points)

The framework developed in the first exercise can be used in this exercise to approximate the height field function of point data. Points are given as triplets $(x_i, y_i, z_i)$ and the approximated height field function $z(x, y)$ can be evaluated on a regular grid represented by quads or triangles. The implementation includes the following tasks (one point per task):

1. Loads data points $(x_i, y_i, z_i)$ from an `.off` file. Implement a weighted least squares (WLS) approximation $z(x, y)$ of arbitrary points $x, y \in \mathbb{R}$ given the input points, with $z(x_i, y_i) = z_i$. For the approximation use a quadratic (bivariate) polynomial and a suitable weighting function with radius $r$ (e.g. Wendland's function). Note that all distances need to be calculated in the input space $\mathbb{R}^2$.
2. Evaluate the resulting moving least squares (MLS) surface on a discrete $km \times kn$ grid, for $n, m, k \in \mathbb{Z}_{>0}$, that covers all input points. The parameter $k$ is currently redundant but will be used in the next task. Visualize the determined $z$ values of all grid points in a mesh (e.g. as a wireframe).
3. Instead of building a MLS surface we can use control points on a regular grid to define a Bézier tensor product surface. The $z$-values in the $m \times n$ grid points calculated using WLS are considered the control points of the Bézier tensor product surface. Additional points on a finer regular grid $km \times kn$ are to be added using de Casteljau's algorithm for tensor product surfaces. Visualize the surface as a triangle mesh.
4. Use the Casteljau's algorithm efficiently to also estimate the surface normals in the points. Visualize the normals as line segments or as shading normals.
5. Approximate the surface normals for the MLS surface. For computing the normals see the last theory question.
6. Bonus: Calculate the exact normals for the MLS surface.

## Exercise 2: Theory (5 + 2 points)

1. How do the basis functions of a first order (= degree 1) B-Spline look like with knot vector $(-1, 0, 1, 2, 3, 4, 5)$? Given the control points $(0, 0), (1, 3), (2, 1), (0, 3)$ and $(4, 1)$, draw the first order B-Spline. Explain the **connection** between degree, continuity, and support of B-Splines at this example. (1 Points)
2. Let a curve be defined as the affine combination of control points $p_i$, where the weights are given by a set of basis functions. Show that the curve is invariant under affine transformations if the basis functions are a partition of unity in the interval defining the curve. (1 Point)
3. There are multiple algorithm variations of the de Casteljau algorithm possible for the evaluation of a Bézier tensor product surface. The two presented in the lecture are (1) row-first evaluation using the 1D de Casteljau algorithm and (2) column-first evaluation using the 1D de Casteljau algorithm. Compare the number of basic operations (addition, multiplication) needed to evaluate a single point $(u, v)$ of the surface with respect to the number of column control points $n_c$ and row control points $n_r$. (1 point)
4. Is the tensor product of two linear polynomials a flat surface? Explain your answer. (1 point)
5. As a simple approximation, the tangents of a local polynomial approximation can be used as the tangents of the MLS surface (computed from the continuously moving weighted least squares approximation). Show that the tangents of the local polynomial are not necessarily the tangents of the MLS surface (1 point). Derive an exact equation for the tangents (2 bonus points).