UNIVERSITÄT BERN

Digital 3D Geometry Processing
Exercise 3 – Geometry Representations

Handout date: 05.03.2019

Submission deadline: 12.03.2019, 13:00 h

What to hand in

A .zip compressed file renamed to `Exercisen-Group.zip` where n is the number of the current exercise sheet. It should contain:

- Hand in **only** the files you changed (headers and source). It is up to you to make sure that all files that you have changed are in the zip.
- A `readme.txt` file containing a description on how you solved each exercise (use the same numbers and titles) and the encountered problems. Indicate what fraction of the total workload each project member contributed.
- Other files that are required by your `readme.txt` file. For example, if you mention some screenshot images in `readme.txt`, these images need to be submitted too.
- For the theory exercise, put `TheoryExercise.pdf` with your solutions in the same .zip you submit for the code.
- Submit your solutions to ILIAS before the submission deadline. Late submissions will receive 0 points! The total points of this homework is 16.

1 Theory Exercise (8 pt)

1.1 (2 pt) Derive a signed distance function in 2D for a line L of the form $y = x$

1.2 (2 pt) Define a planar curve that has a sharp corner (discontinuity of normal vector) using only polynomials as coordinate functions. Please give a parametric representation.

1.3 (2 pt) Denote $\{(x(\frac{k}{N}), y(\frac{k}{N}))|k = 0, \dots, N\}$ as a uniform sampling of a 2D curve $(x(t), y(t)), t \in [0, 1]$. As the number of sampling N increases, does chord length monotonically increase (non-decrease). If yes, give a proof; if no, give a counterexample.

1.4 (2 pt) Find a sequence of 2D curves $\mathbf{C}_i(t), t \in [0, 1]$ which satisfy:

$\lim_{i\to\infty} \mathbf{C}_i(t) = \mathbf{L}(t), \forall t \in [0, 1]$ where $\mathbf{L}(t)$ is a straight line segment in$[0, 1]$

the length of $\mathbf{C}_i(t)$ does not converge to the length of $\mathbf{L}(t)$ in $[0, 1]$

Submit your solutions in a file named `TheoryExercise.pdf`.

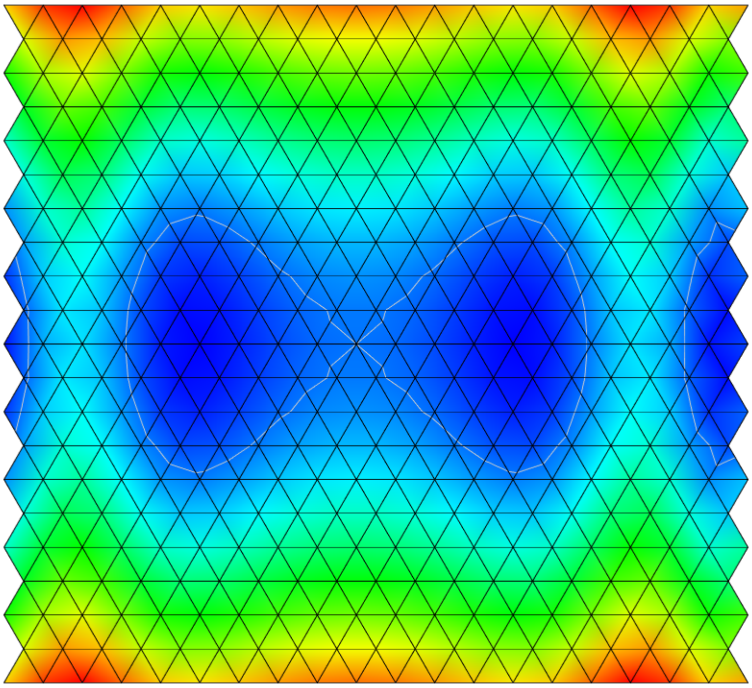

Figure 1: Final result for the *Regular Mesh 2* and function $F(x,y) = y^2 - \sin(x^2) = 0$.

2 Coding Exercise (8 pt)

Download `dgp2019-exercise3.zip` from ILIAS. Extract and replace the files in **Plugin-DGPExercise** with the new ones. For this exercise you will need to fill in the missing code in the file `IsoContouring.cc`. You are given a triangular mesh with a list of vertex coordinates stored in `v_positions` and a list of vertex indices forming each triangle stored in `triangle_ids`. For each vertex third coordinate is zero, since the meshes are all in 2D.

The goal of this exercise is to implement a contouring method like *marching squares algorithm* but on triangles rather than squares. For each vertex you can compute a scalar iso-value from an implicit function. Then for each triangle, check if the signs of the vertex iso-values are not all the same. If the signs are different use linear interpolation to compute the edge that passes through that triangle. Add two end-points of that edge in a vector `segment_points`. Here `segment_points` is a vector of points, so add the two end-points one after the other.

Test the following implicit functions:

- $F(x,y) = \sqrt{x^2 + y^2} - 1 = 0$
- $F(x,y) = y^2 - \sin(x^2) = 0$
- $F(x,y) = \sin(2x + 2y) - \cos(4xy) + 1 = 0$
- $F(x,y) = (3x^2 - y^2)^2 y^2 - (x^2 + y^2)^4 = 0$

After correctly implementing the functions, you should be able to test any of them by picking in the combobox.
To see the result on different meshes, simply load the provided meshes from the `data` folder. One correct output is shown in Figure 1.