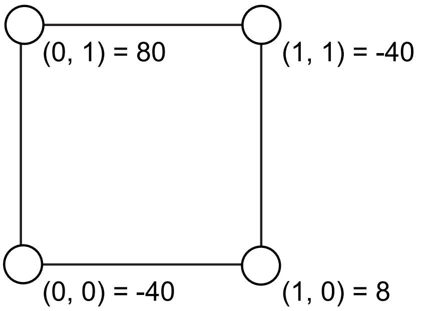
Thomas Ertl, Steffen Frey *Stuttgart, 2019-5-31*

Scientific Visualization (Assignment 8)

Exercise 8. 1 [3 Points] Asymptotic Decider

Determine a solution for the following ambiguous case of the Marching Squares algorithm using the *Midpoint Decider*. Calculate the asymptotes and the function value at their intersection. The coordinates of the grid points are given in parentheses along with their function values.



Exercise 8. 2 [3 Points] Octrees

Consider a full octree where each leaf node represents a voxel for isosurface extraction. The volume to be visualized is a rectangular grid with $500 \times 500 \times 200$ voxels at 32-bit floating point precision. Calculate the estimated memory consumption...

(a) ...of the full octree in bytes.

(b) ...assuming that each node stores the minimum and maximum values of its subtree.

Exercise 8. 3 [6 Points] Color Mapping with Shaders

Prerequisites Obtain the skeleton shader (`colormap2D.glsl`) from ILIAS and familiarize yourself with ShaderToy (`https://www.shadertoy.com/new/`). ShaderToy only needs a WebGL-compatible browser, such as Google Chrome, Mozilla Firefox, or Microsoft Edge. To get started, just copy the skeleton shader into the source text box on the right and press the play button on the lower left of the source box. Before you have implemented the following task, the result image on the left should be white.

ShaderToy essentially gives you a pre-configured fragment shader you can work with. Opposed to standard fragment shaders from OpenGL, for example, some additional input parameters like the current time, mouse coordinates, or sound processing capabilities are available.

Your task is to implement a function modeling a 2D time-varying scalar field and visualize it using a color map (cf. Figure 1):

- Map 2D texture coordinates $(\phi_x, \phi_y) \in [0,1]^2$ to a scalar value $\mathbb{R}$ using:

  $$f(t, \rho_x, \rho_y, \phi_x, \phi_y) = |(\sin(t) + 1) \cdot \sin(\rho_x \cdot \phi_x + \cos(t)) \cdot \cos(\rho_y \cdot \phi_y + \sin(t))|$$

  where t is playback time, (ρ_x, ρ_y) is resolution, and (ϕ_x, ϕ_y) are texture coordinates.

- Map f to a color. To do so, define a transfer function using linear interpolation based on the following support points:
  $f = 0.0$: blue
  $f = 0.5$: white
  $f = 1.0$: red

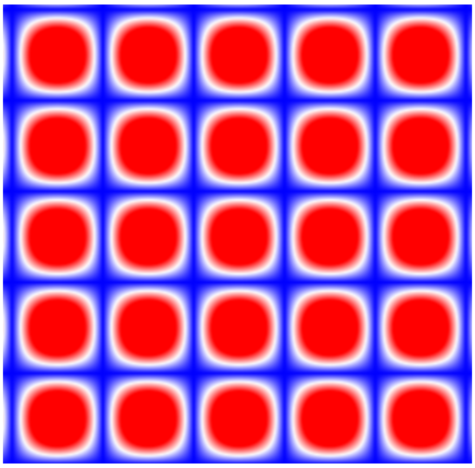


Figure 1: Example frame of the resulting animated texture.

Please submit your solution (`colormap2D.glsl`) along with two examples screenshots!

Due to the Pentecost holidays the discussion of the exercise will be in the week starting from 2019-6-17.

Submission Deadline: 2019-6-7, 23:55