Interactive Visual Data Analysis

Assignment 9 – *Stream Lines and Ribbons*

This week, we will enhance the Vis tool to support characteristic lines as an additional flow visualization primitive next to particles.

9.1. Stream Lines

Implement stream lines: The user should be able to select the number and length (=number of vertices) of the lines, as well as the integration step size. The starting positions should be distributed randomly within a probe, just like for the particles. Allocate a buffer to hold all the line vertices, and implement the integration in a compute shader, running one thread per line. After each integration step, write the current position into the buffer.

The stream lines need to be recomputed every time the vector field changes, which means every frame while the vector field is playing back. If you haven't done so already, create a Play/Pause button in the UI so it is possible to look closely at the stream lines.

Render the stream lines using line primitives. It should be possible to turn the stream lines on/off independently of the particles for each probe.

9.2. Stream Ribbons

Enhance your stream lines to also support rendering as ribbons. Either extend your stream line integration shader or create a new one which also tracks the ribbon orientation by integrating additional particle positions. Use either two particles to represent the left and right edge of the ribbon, or three particles: a central particle for the position plus a left and right "ghost particle" to track the orientation. In either case, make sure the distance between the particles stays constant by adjusting their positions after each integration step. In the three-particle case, adjust only the ghost particles and also make sure they stay symmetric with respect to the center particle.

You'll need to store either pairs of positions (left/right edge) or position+direction (center and sideways vector) in the buffer. The initial orientation of each ribbon you can choose arbitrarily (one option is to align the ribbons to the vorticity direction, but a fixed direction is OK as well).

Render the ribbons with a user-defined width, using two triangles per segment (disable back face culling!). You can either create an appropriate index buffer or use the geometry shader. Apply Phong lighting to the ribbons.

9.3. Visualize

Like every week, take a few screenshots of some pretty renderings generated by your tool and commit them to a folder called `screenshots/assignment9/` outside of your solution directory.

The working solution must be committed until **December 18, 09:00am**. If anything is not working as described here or if you want a specific SVN revision to be rated, explain yourself in the `readme.txt` file within your `solution` directory.