A static mixing reactor

Scientific Visualization & Virtual Reality

Eva van Weel (10244743) and Jaromir Camphuijsen (6042473)

September 28, 2015

1 Introduction

In order to mix (very viscous) fluids in an industrial setting, a static mixing reactor can be used. The flow in such a reactor can be visualized to obtain better understanding of whether or not the fluids mix and in what way. In this assignment the flow in such a reactor is visualized and discussed, as well as the geometry of the reactor. The dataset[1] that is used is represented as a regular structured grid consisting of two data arrays: *scalars* that represent the geometry of the mixer and *vectors* that represent the simulation of flow through the mixing reactor in terms of a static 3D vectorfield.

2 Approach

In Figure 1 the VTK pipeline that produced the visualization that will be discussed in the following section is show. The pipeline visualizes the approach taken and will be discussed in further detail below.

The first step in the process is to read in the data, that is the regular structured grid, that represents the fluid flow and the geometry of the reactor. This is achieved by calling the class *vtkStructuredPointsReader()*. The data forms the input for the contour filter that produces isosurfaces which show the geometry of the reactor (*vtkContourFilter*), the outline filter that shows the extent of the data set (*vtkOutlineFilter*) and the streamline filter that integerates the vectorfield to generate streamlines (*vtkStreamTracer()*). The latter filter also requires another input, namely a source from which particles are release which form the starting points of the streamlines. This source is created with a class called *vtkProgrammableSource()*, which allows the user to generate source particles by a self defined function. After these first data filters are used, a set of different filters is used to filter the data even further. The class *vtkPolyDataNormals()* is used to create normals on top of the isosurfaces of the reactor's geometry, which allows for smooth surface shading of the isosurface. The streamlines are also filtered with a second filters which wraps a tube around the streamlines to make the lines more prominent in the visualization (*vtkTubeFilter*).

After filtering the input data, a mapper is used to map the polydata to graphics primitives (*vtkPolyDataMapper*). Subsequently, the output of these mappers is used to create actor objects (*vtkActor()*) which can be shown in the rendered scene that will form the visualization. These actors are then added to a renderer (*vtkRenderer*), which is subsequently added to a window (*vtkTkRenderWidget*) which shows the user the visualization.

[1]The dataset can be downloaded from: `http://bit.ly/1uESaMC`

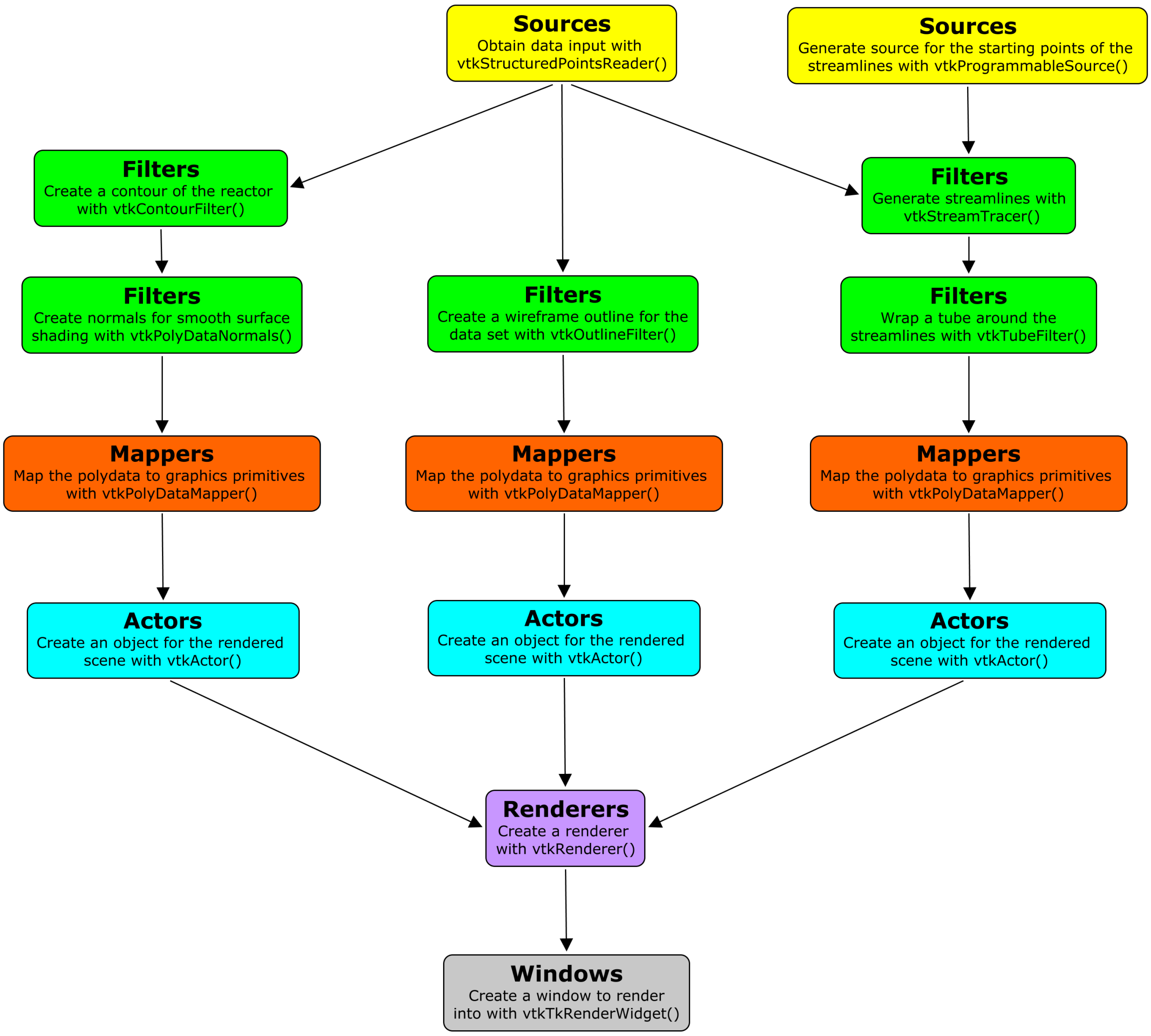

Figure 1: The VTK pipeline that produced the visualization

3 Findings

Figure 2 and Figure 3 show snapshots of the produced visualization with two different initial states. The test particles are created in rectangular areas on one side of the reactor. The orientation of the rectangles clearly determines the amount of mixing that occurs. When the two rectangles are placed vertically and aside eachother, the fluids will not mix due to the orientation of the diagonally placed bars in the reactor. It is clearly seen in figure 2 that as the fluids start of separated, they remain that way during their way through the reactor. However when placed horizontally and on top of eachother, a certain mixing will take place due to this reactor geometry. The diagonally placed bars will guide the top fluid downward and the bottom fluid upward. This results in mixing of the two fluids. However this mixing is very limited as can be seen from figure 3. It shows a clear structure in the mixing of the two fluids. Random mixing of two fluids in a static reactor remains a problem. To get a more homogeneous mix, the reactor geometry could be repeated in different orientations.

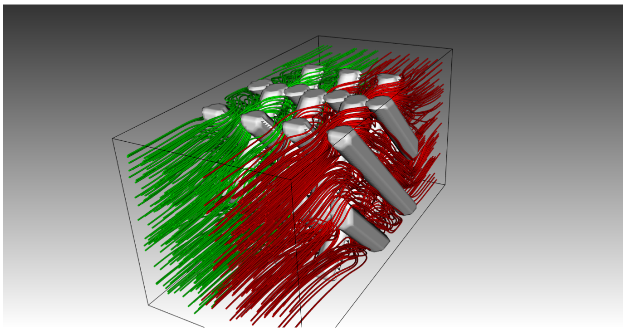

Figure 2: When the two fluids are simulated starting next to eachother, they end up that way as well.

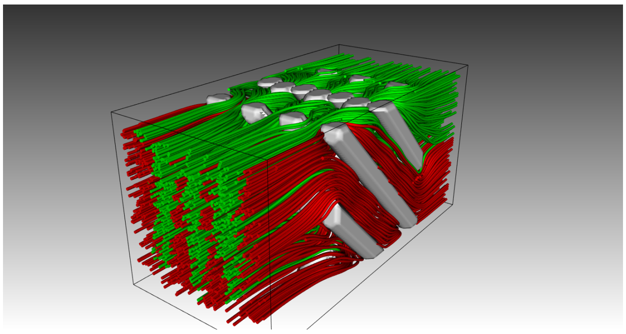

Figure 3: When simulated on top of eachother, the fluids will mix. Not at random, but in a very distinct pattern.

Paraview was used to visualize the magnitude of the velocity vectors and map it to a colorscale as can be seen in figure 4. While an extensive search in the vtk documentation for the same functionality did not result in succes, we can still analyze the obtained picture. In this visualization, absolute speed of the test particle is mapped to the color of the tube. For lower speed, tubes are drawn in blue and for higher speed in red. We can see a clear increase in speed for all streamlines in between the reactor tubes. This is as expected, since part of the reactor volume is filled up with the diagonal rods. As the fluid maintains a steady current on both sides of the reactor, due to continuity, the speed in between the rods has to increase.

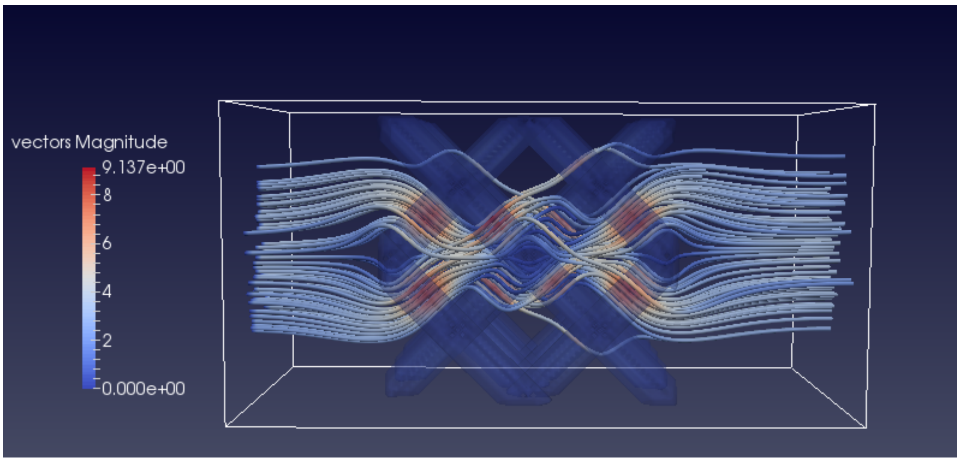

Figure 4: Visualization of the magnitude of the flow velocity in the reactor.