Visualization Homework 4
Seren Lowy

1.1 Stream surfaces: concerns

Stream surfaces are knitted together out of stream ribbons that are placed next to each other following an adaptive strategy. Initially, we place some seeds for streamlines in a pattern so that adjacent seeds start to form stream ribbons that are tangent to the vector field. The edges of each stream ribbon are shared with adjacent stream ribbons.

There are some concerns about the shape of the surface as the ribbons get longer. The streamlines forming them may get too close together (convergence), too far apart (divergence), or start moving in opposing directions from their neighbors so that the surface becomes twisted (shear). We need to consider how to adapt the seeds during computation to mitigate these events. For convergence, we can terminate a streamline and merge the two ribbons that were sharing it into one ribbon. For divergence, we need to add a new seed point somewhere between the edges of the overly wide ribbon and start tracing a streamline from it, then split the wide ribbon into two ribbons who each get one of the outside edges of the prior ribbon and share the new streamline as inside edges. For shear, we may need to terminate streamlines on one side of the shear while adding streamlines on the other side, shifting the ribbons as appropriate.

We still need to consider a good pattern to place the starting streamline seeds. In this assignment, we are implementing an interactive widget that places all the seeds along a movable bar. When seeds are evenly spaced and aligned along a linear structure, the lines will start forming an orderly surface right away and will only have to respond to minimal disturbances from convergence, divergence and shear. We might find that the stream surface from one placement of the bar does not show enough of the shape of the data. This is why it helps to have interactive movement capability for the bar. Moving the bar to a different location and orientation, we can discover different stream surfaces which illustrate the data better.

If there are multiple stream surfaces, they should be placed at even distances and terminate when they get too close to other stream surfaces, just like stream lines. Especially when there are multiple stream surfaces, parts of the surfaces become occluded inside, like a bundled blanket or scarf. The stream surface should be made out of a virtual material with transparency so that the inner folds can be seen through the outer folds. When the shape of a stream surface is complex, it can be difficult to notice features of its shape. Applying a texture to the surface, using advection (like LIC) or accentuating the streamlines on the surface in a different color from the intervals of material, can help illustrate features.

Source: Lecture 15 (3D vector fields)

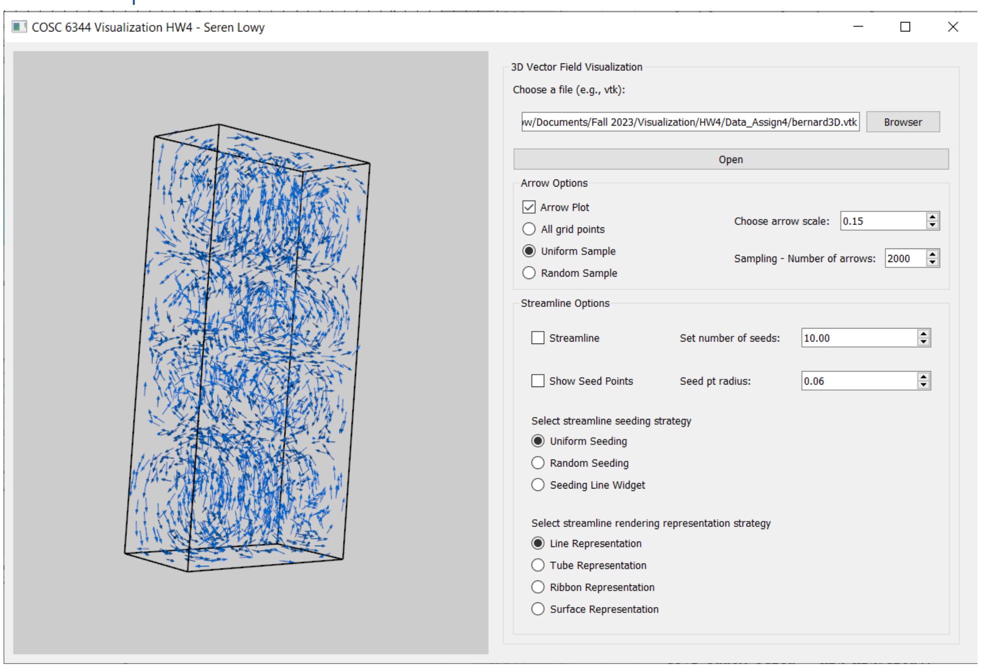
1.2 Line integral convolution in 3D

To adapt LIC to 3D, we need an extra dimension in the starting texture (noise) in order to compute convolution outputs for 3D streamlines in the 3D vector field. The starting texture is “volumetric noise”, or a series of frames of noise. Instead of tracking only x and y displacement in a 2D grid of pixels, we track these as well as z displacement according to the 3D streamlines and use voxel values from the volumetric noise array selected according to all 3 dimensions.

The resulting texture is also a volumetric texture, and contains a 3D map of voxels that arguably form patterns representing the flow field, but it is not very useful as a visualization. The method for rendering a volumetric texture is direct volume rendering (raycasting or splatting). To do DVR appropriately, we need a transfer function to guide choices of color and opacity for voxel regions. But since the volumetric texture from LIC was generated iteratively from a vector field alone, without a corresponding scalar field representing physical properties like density that could correspond to colors or opacity, transfer functions have to be chosen arbitrarily, which means it is hard to pick out meaningful features. No matter how good the transfer function is, all of the noisy brightness variations of convolution paths makes too much information heaped on each other throughout the volumetric texture, and the view through from the outside to the inside is usually too faint to see patterns in the outer layers, or too opaque to see in the inner layers, or too noisy to see the middle. The texture visualization at best, usually amounts to a cube of cut planes showing textures on the outer surface, with maybe some limited awareness of how the texture alters just below the surface.

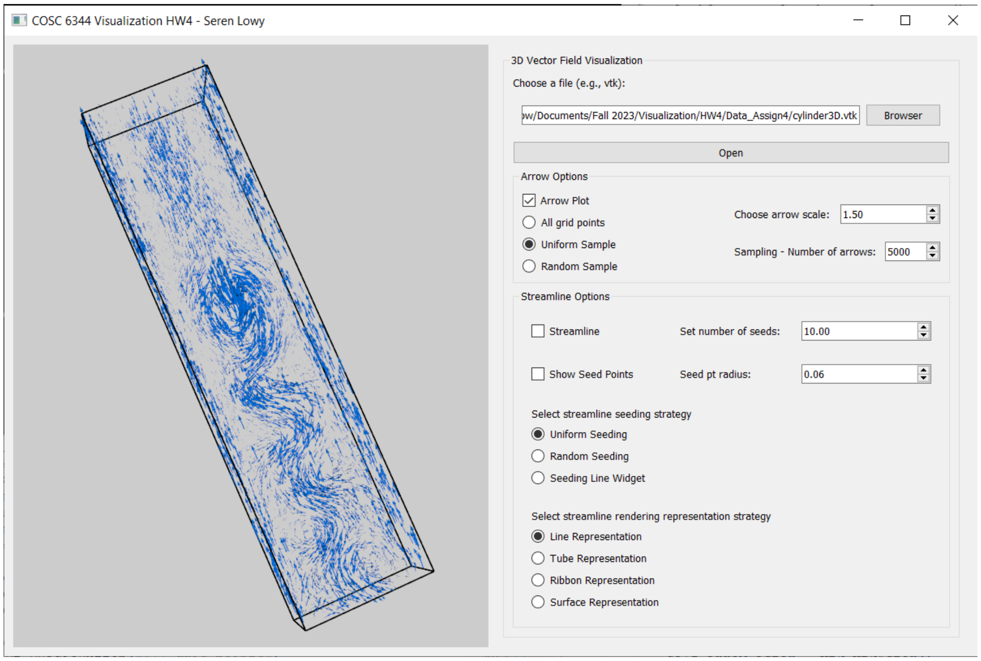
Source: Lecture 15 (3D vector fields)

2. Arrow plots in 3D

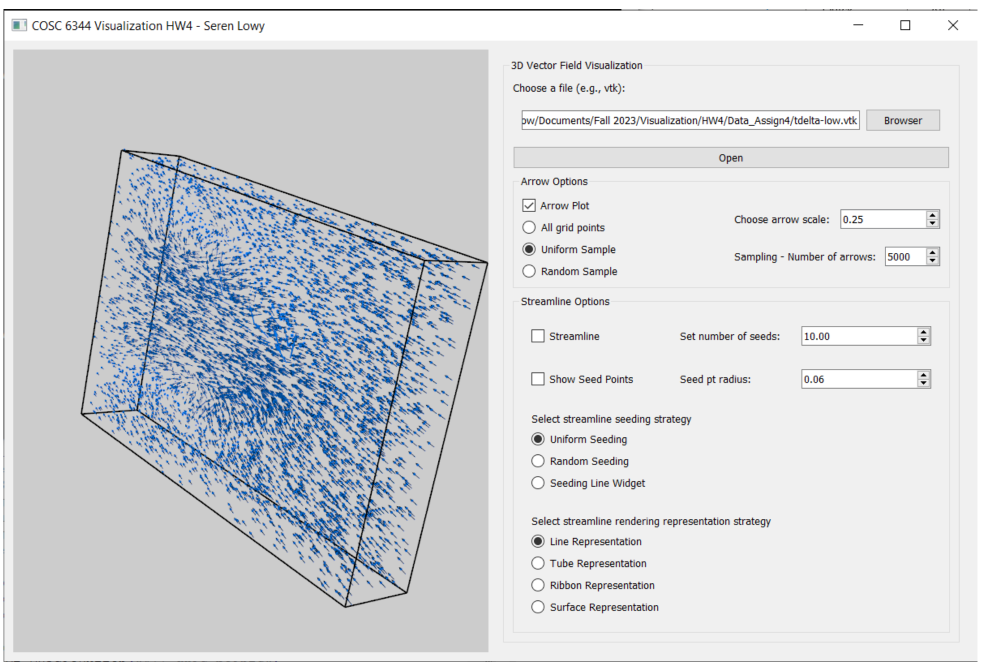


Dataset: bernard3D.vtk
Number of arrows: 2000 | Sampling type: uniform | Arrow scale: 0.15

Visualization Homework 4
Seren Lowy



Dataset: cylinder3D.vtk
Number of arrows: 5000 | Sampling type: uniform | Arrow scale: 1.50

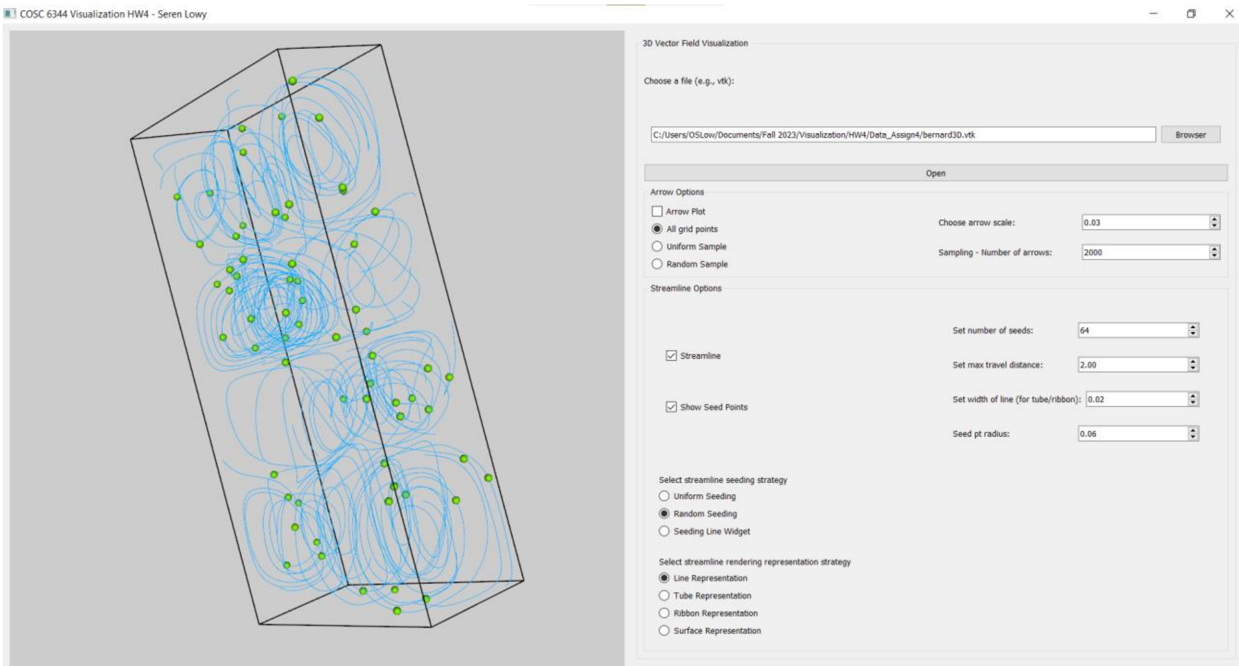


Dataset: tdelta-low.vtk
Number of arrows: 5000 | Sampling type: uniform | Arrow scale: 0.25

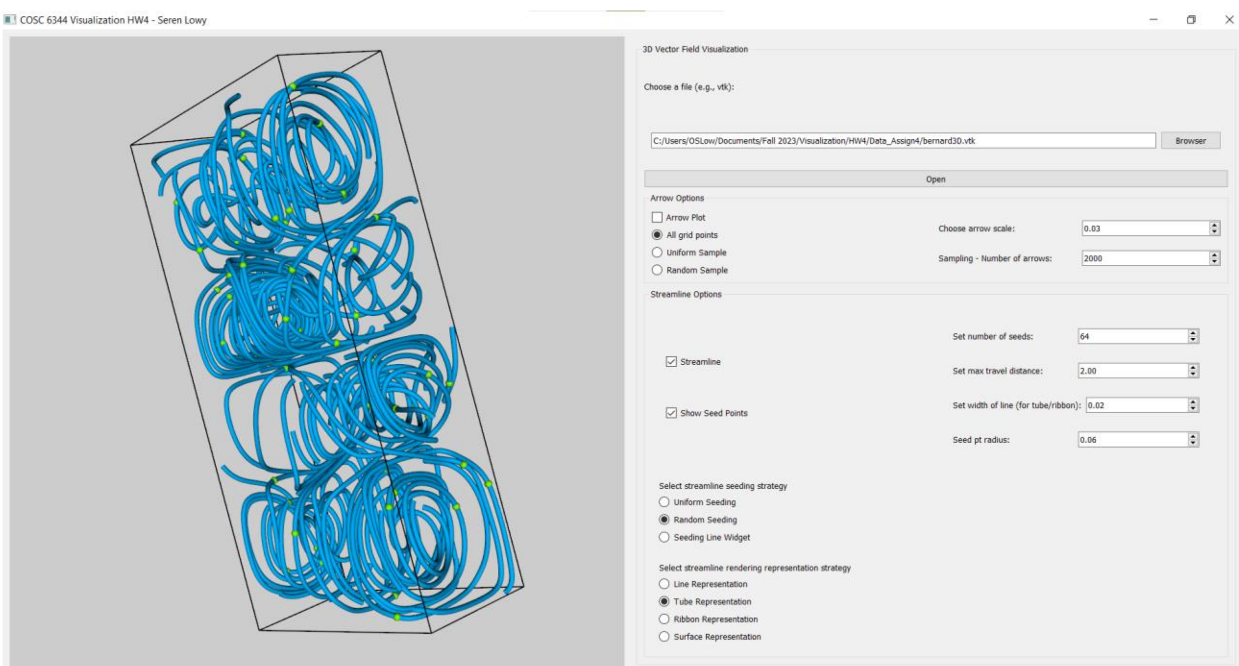
Visualization Homework 4
Seren Lowy

3. Streamlines, tubes, and ribbons

Dataset: bernard3D.vtk

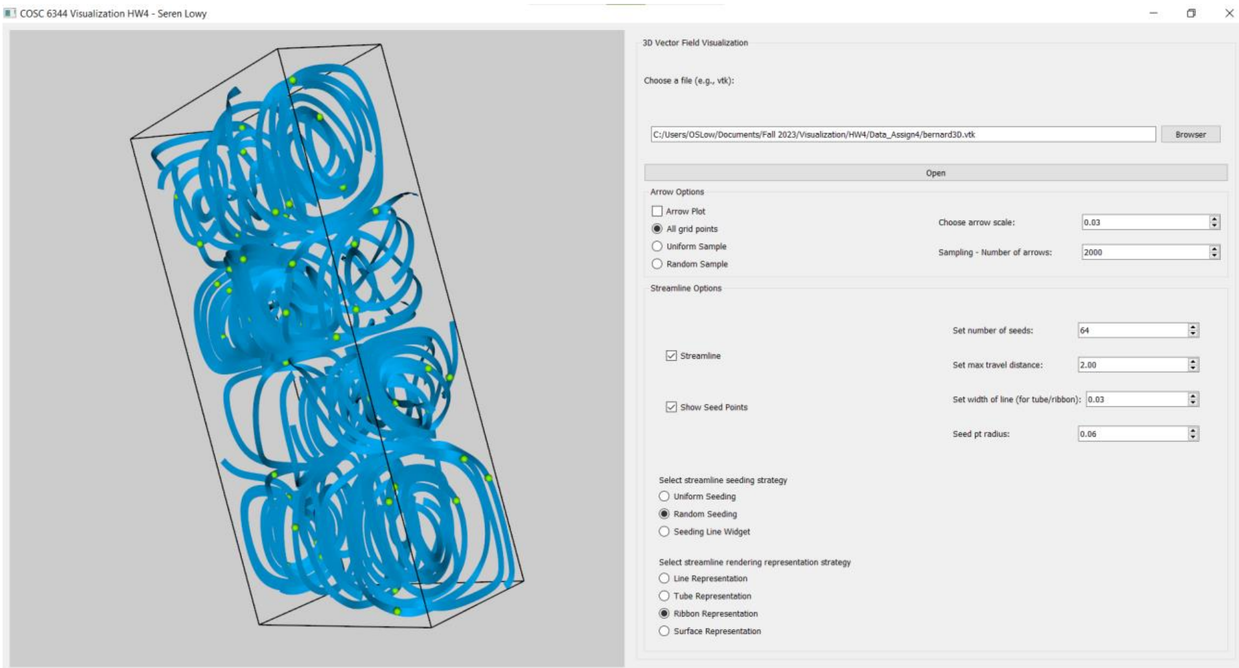


Number of streamlines: 64 | Seed placement type: random | Propagation: 2 units | Display type: line

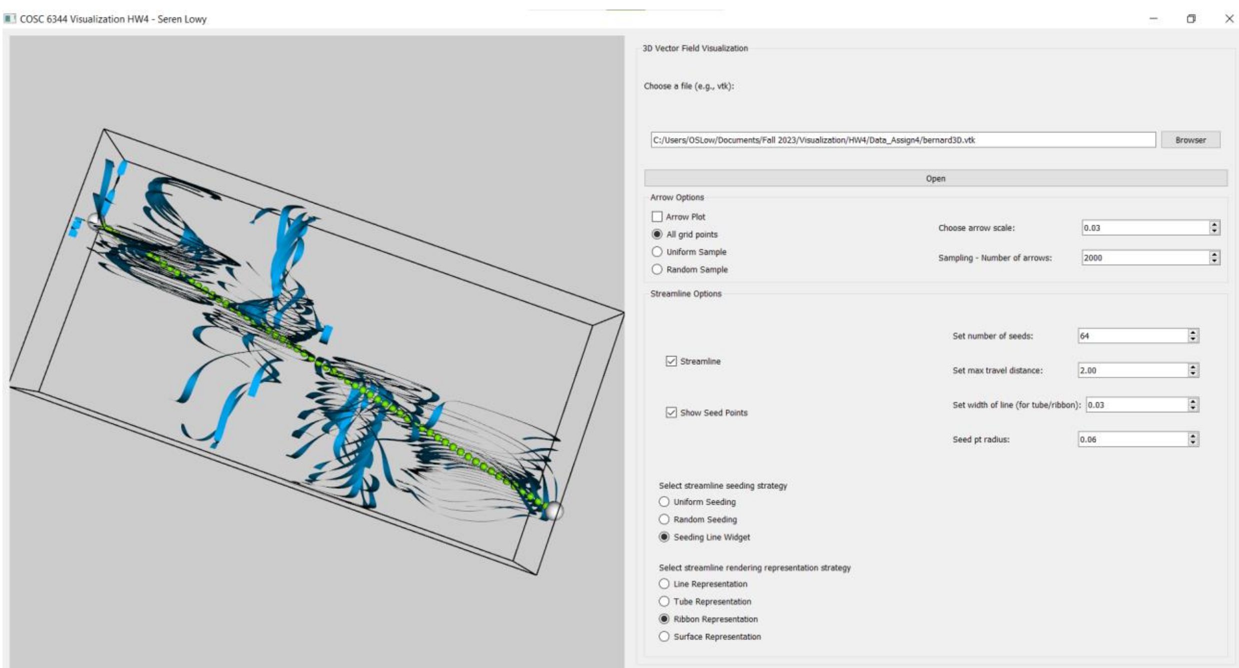


Number of streamlines: 64 | Seed placement type: random | Propagation: 2 units| Display type: tube

Visualization Homework 4
Seren Lowy



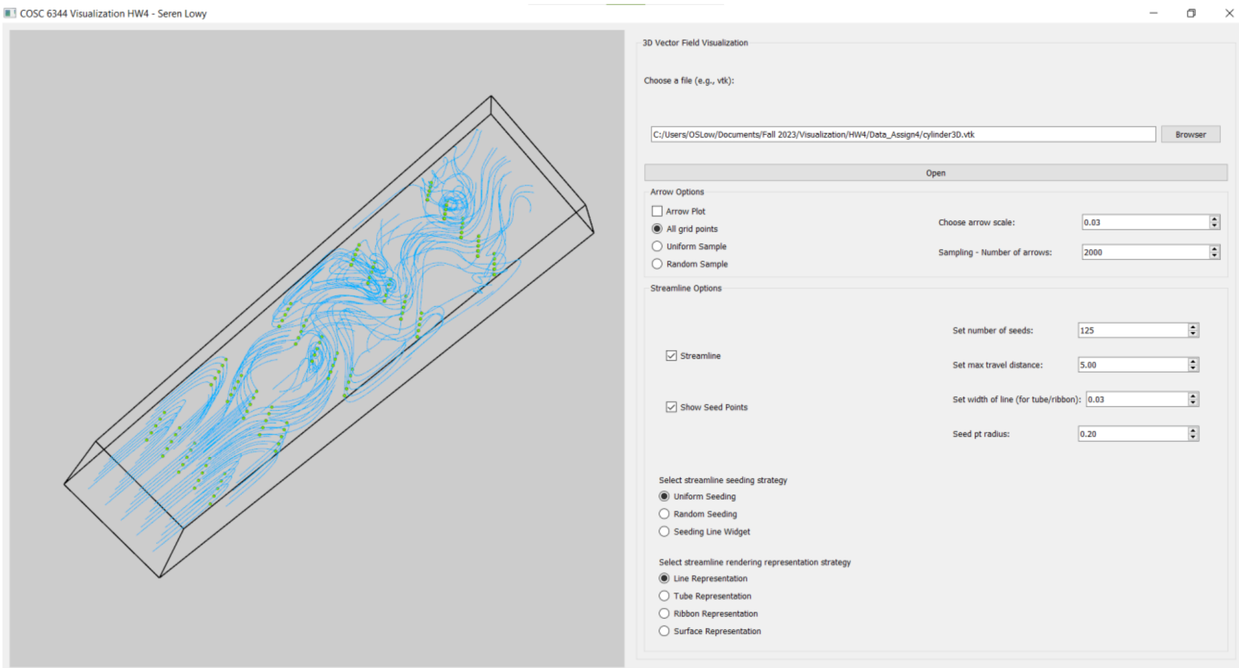
Number of streamlines: 64 | Seed placement type: random | Propagation: 2 units| Display type: ribbon



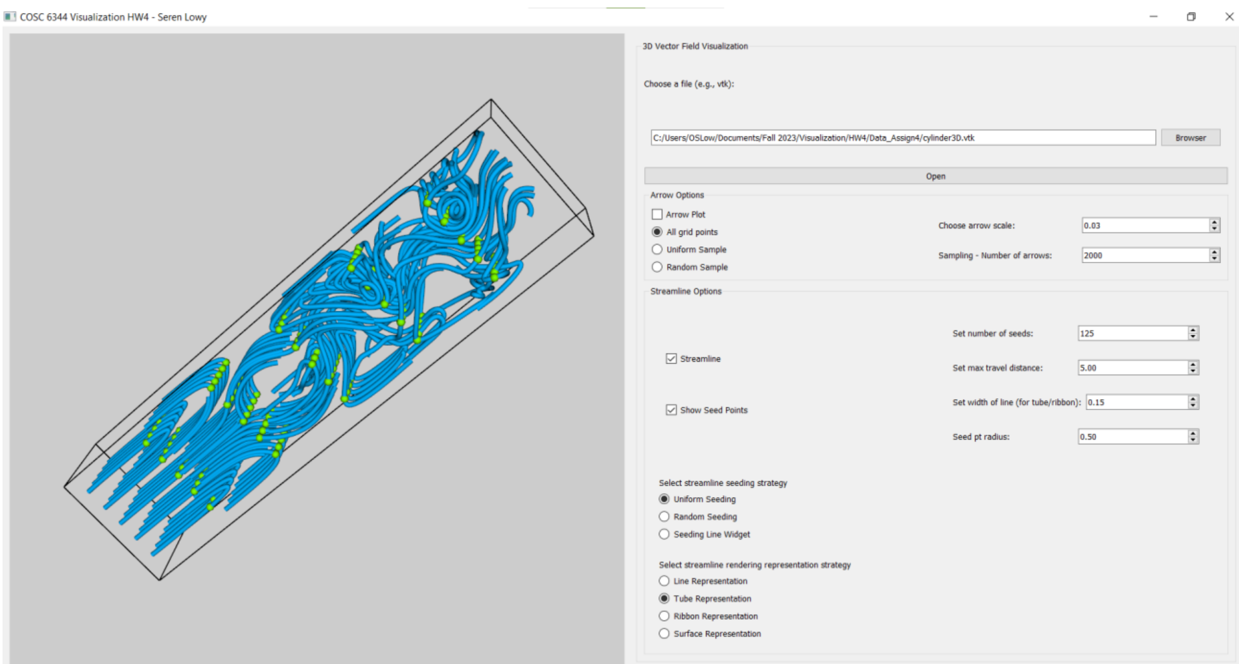
Number of streamlines: 64 | Seed placement type: Line widget | Propagation: 2 units | Display type: ribbon

Visualization Homework 4
Seren Lowy

Dataset: cylinder3D.vtk

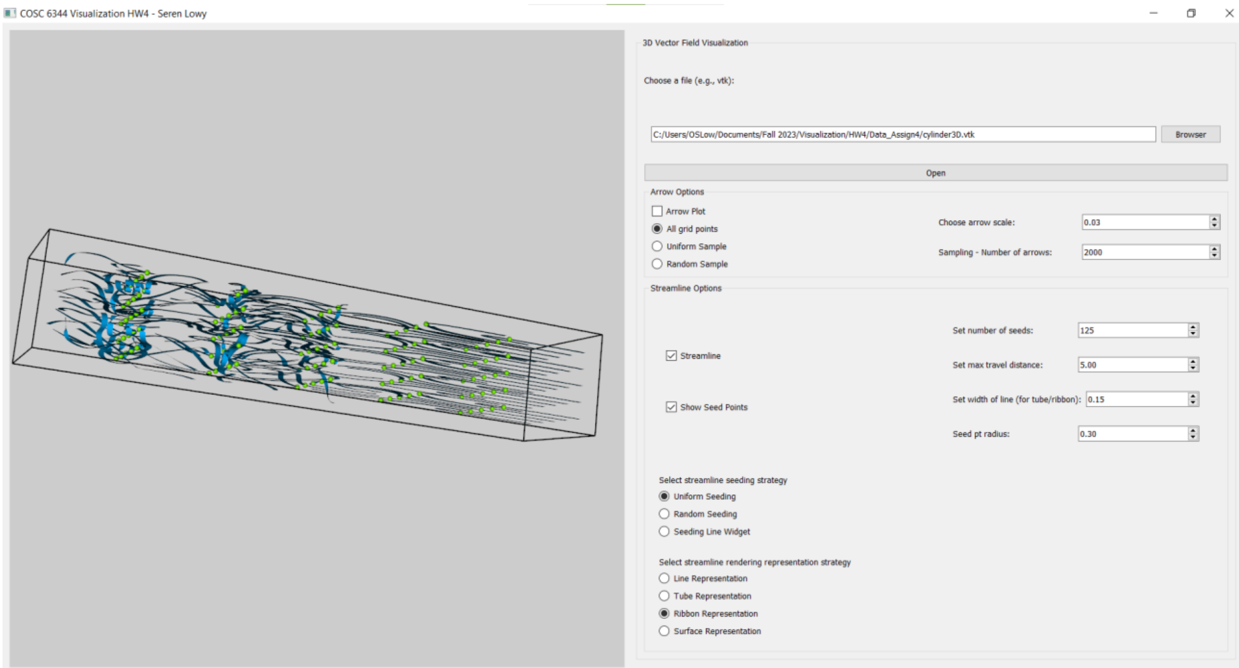


Number of streamlines: 125 | Seed placement type: uniform | Propagation: 5 units| Display type: line

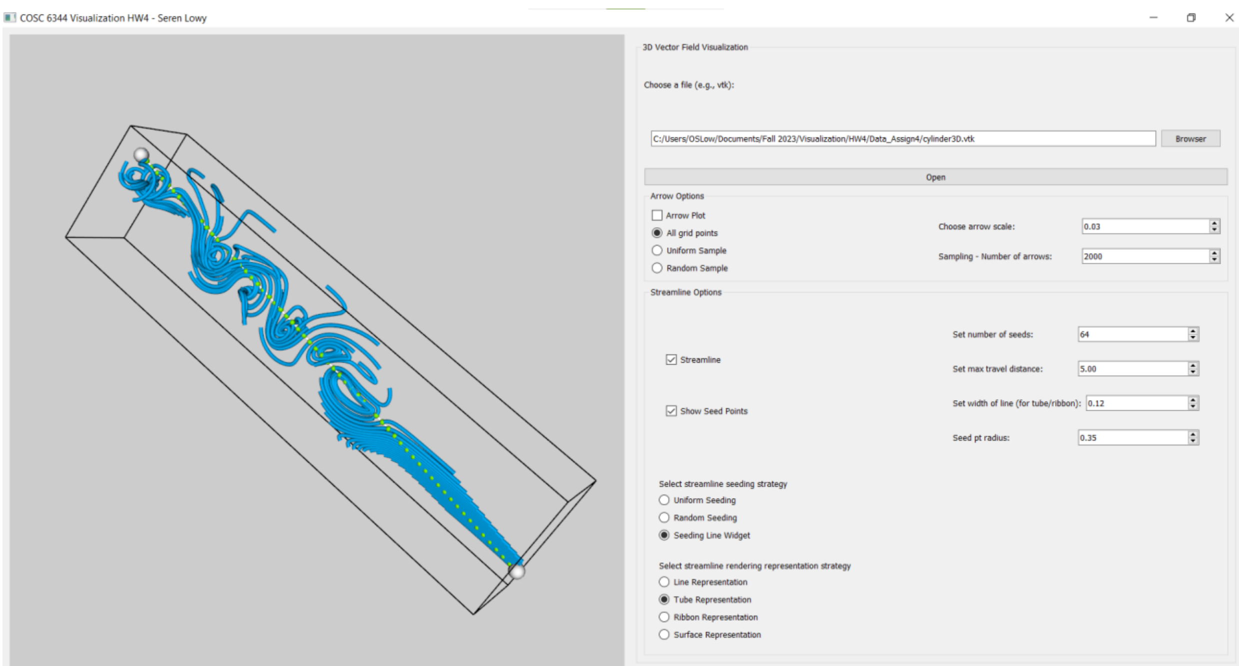


Number of streamlines: 125 | Seed placement type: uniform | Propagation: 5 units| Display type: tube

Visualization Homework 4
Seren Lowy



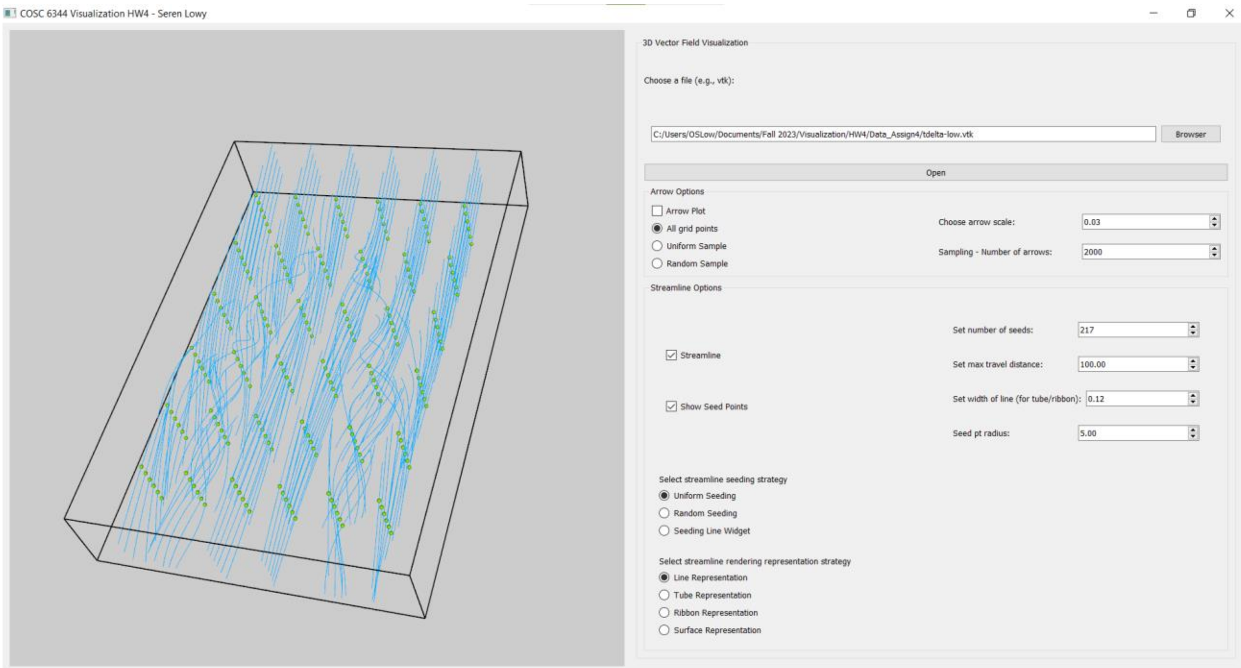
Number of streamlines: 125 | Seed placement type: uniform | Propagation: 5 units| Display type: ribbon



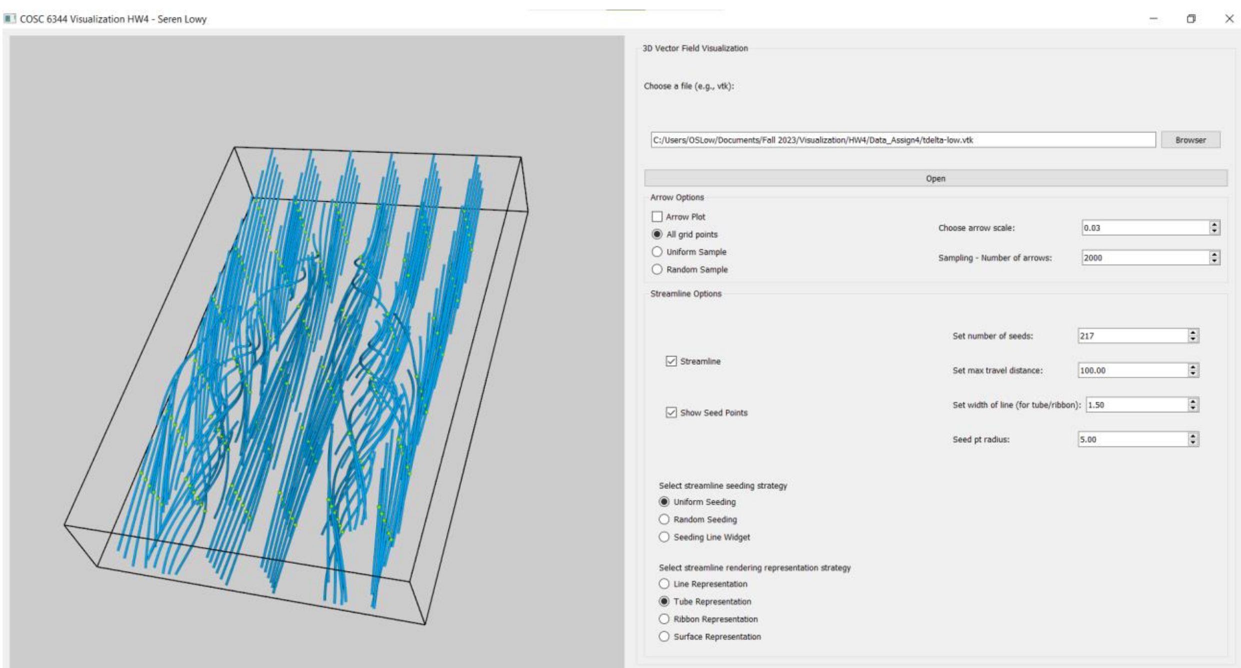
Number of streamlines: 64 | Seed placement type: line widget | Propagation: 5 units| Display type: tube

Visualization Homework 4
Seren Lowy

Dataset: tdelta-low.vtk

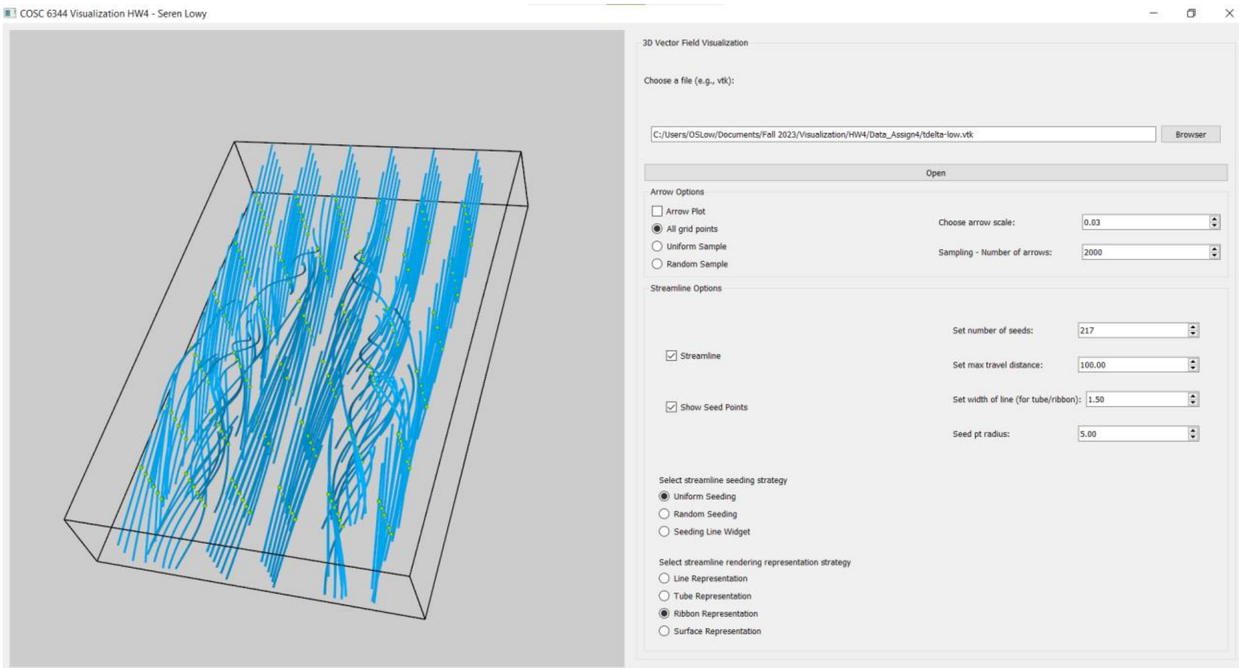


Number of streamlines: 217 | Seed placement type: uniform | Propagation: 100 units| Display type: line

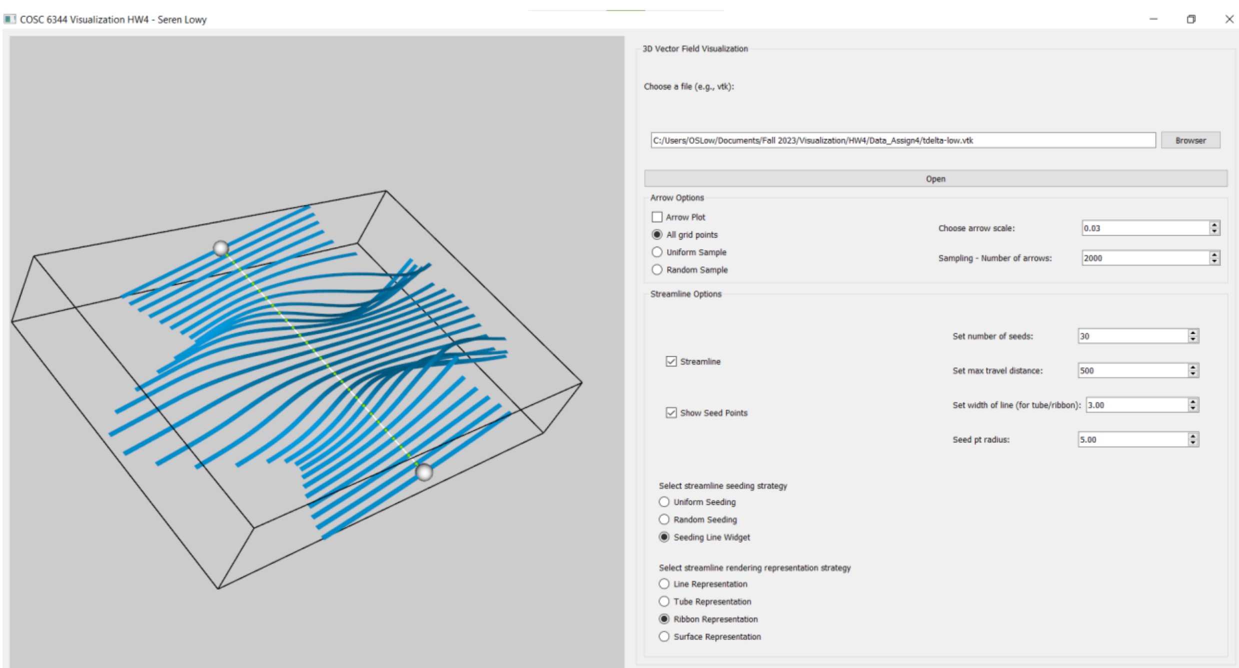


Number of streamlines: 217 | Seed placement type: uniform | Propagation: 100 units| Display type: tube

Visualization Homework 4
Seren Lowy



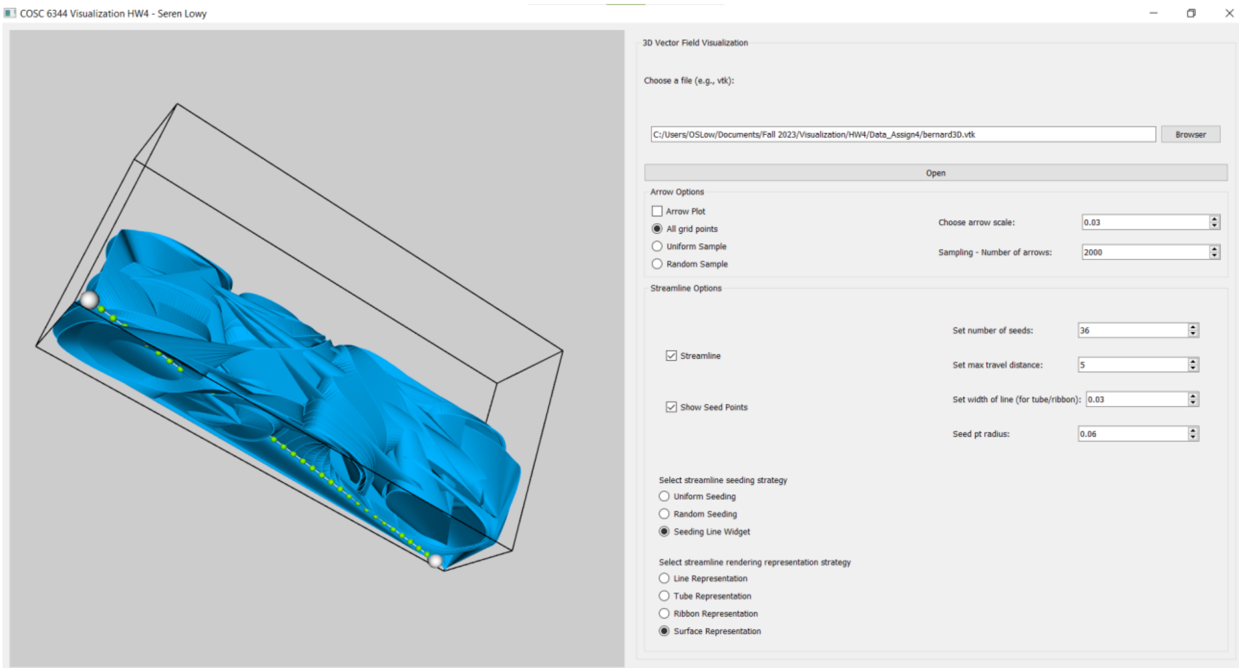
Number of streamlines: 217 | Seed placement type: uniform | Propagation: 100 units| Display type: ribbon



Number of streamlines: 30 | Seed placement type: line widget | Propagation: 500 units| Display type: ribbon

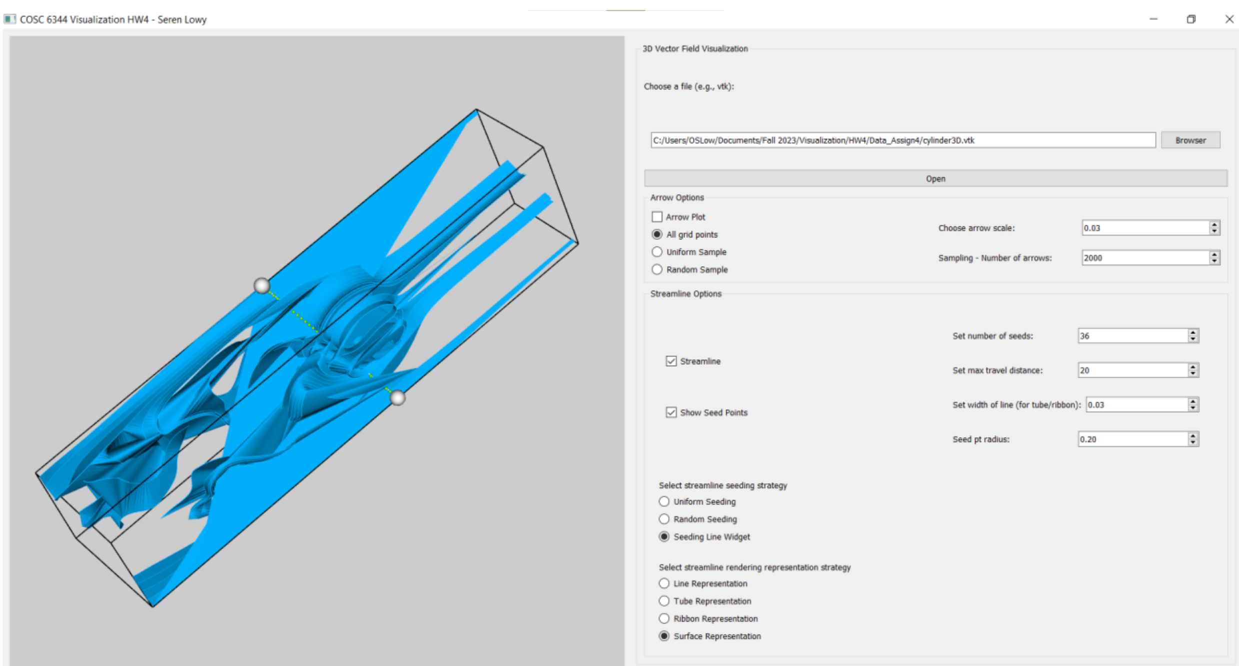
Visualization Homework 4
Seren Lowy

4. Stream surfaces



Dataset: bernard3D.vtk

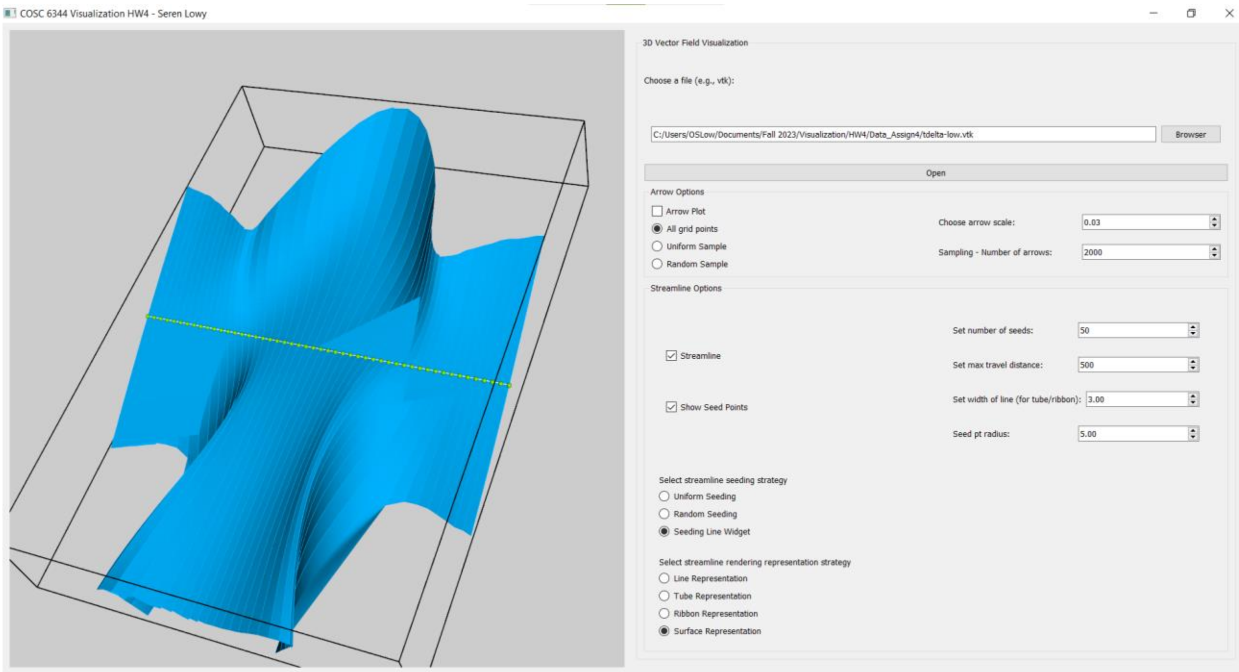
Number of streamlines: 36 | Seed placement type: line widget | Propagation: 5 units| Display type: surface



Dataset: cylinder3D.vtk

Number of streamlines: 36 | Seed placement type: line widget | Propagation: 20 units| Display type: surface

Visualization Homework 4
Seren Lowy



Dataset: tdelta-low.vtk

Number of streamlines: 50 | Seed placement type: line widget | Propagation: 500 units| Display type: surface