

## 1.1 LIC algorithm

Line integral convolution pseudocode:

1. Define stream line seed point at the center of pixel  $(x, y)$ 
  - a. Seed point  $P_0 = (x + 0.5, y + 0.5)$  where  $x$  and  $y$  are coordinates of the pixel
2. (Advection) Compute local stream lines from seed point, forward and backward
  - a. Iteratively compute the sequence of pixels a stream line travels through
  - b. The next point is given by a vector defined as follows, added or subtracted relative to the current point.
    - i. direction = the vector field value at the current pixel (measured at the integer floor of the current point)
    - ii. magnitude = the minimum needed to cross a pixel boundary in this direction from the current point
    - iii. Add this vector to the current point for forward advection
    - iv. Subtract it for backward advection
  - c. Save the magnitude values and point values of all the advection steps.
  - d. Keep computing for  $n$  many steps forward and  $n$  many steps backward.
3. Define a convolution kernel, such as a Hanning function.
4. Compute a finite integral of the kernel  $2 * n$  many times over different segments of the domain.
  - a. The widths of the segments are the magnitude values from advection.
  - b. Iterate, starting from 0 on each of two passes, once for the forward advection magnitudes and once for the backward advection magnitudes
  - c. Start the next segment where the previous segment ended. Move in order along the lists of advection magnitudes.
  - d. Compute the integral of each segment and store in two lists, one for the forward advection and one for the backward advection.
5. Use the point values from the advection steps to look up image brightness values in the source image data (typically an image of white noise) and compute a weighted average (convolution) of brightness values over the variable-stepped pixel traversal path
  - a. Use the integral values as weights in weighted sums of the pixel brightness values
  - b. Divide by the sum of all the integral values
  - c. Produce a new pixel with brightness equal to the convolution result and place it in the output image at the same  $x, y$  position as the current center pixel in the input.
6. Repeat 1-5 for every pixel in the image.

Source: Cabral, B. and Leedom, C. (1993). "Imaging Vector Fields Using Line Integral Convolution." Lawrence Livermore National Laboratory.

## 1.2 Placing streamlines

Streamlines suffer from a problem where they bunch together into thick lines following roughly the same paths and disrupt the readability of the plot. We need to place the starting positions (seed points) of streamlines iteratively, with respect to existing lines in the plot, not just according to some distribution determined all at once such as uniform or simple random. That is, even if the starting points are far enough from each other, the rest of the line's bodies may converge anyway and become closer, so we need to consider the whole body of the line when placing starting points. Any starting point of a new line

## Visualization Homework 3

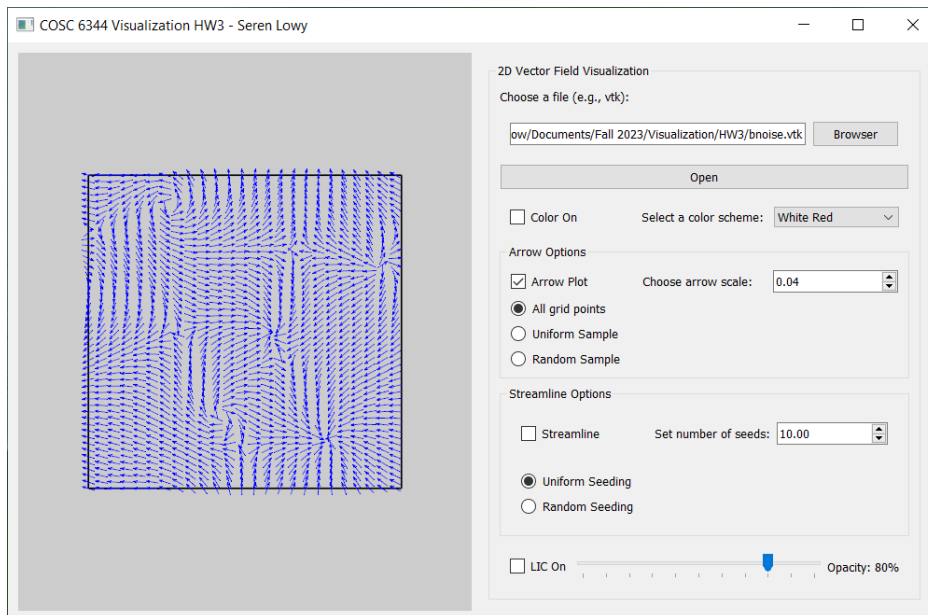
Seren Lowy

should be a minimum distance away from all parts of all previously drawn lines. This concern also applies to drawing the line after placing the initial start point. If a streamline gets too close to any portion of previously drawn lines, we need to quit drawing early.

Yet, if streamlines are too far from each other, the plot may fail to reveal some features of the vector field in good detail. We need to populate the plot with more streamlines if many of the streamlines placed so far are ending while they are still short, so that streamlines can trace over a fairer percentage of the plot's area. We need to scatter streamlines over all regions of the plot. If there are areas that streamlines starting from other regions tend to avoid, such as saddles or sources, we need to place more starting points in these regions. This could be addressed by measuring the density of streamlines in cells of a grid over the plot area, and erasing lines from dense cells while placing new ones in sparse cells.

## 2. Arrow plots

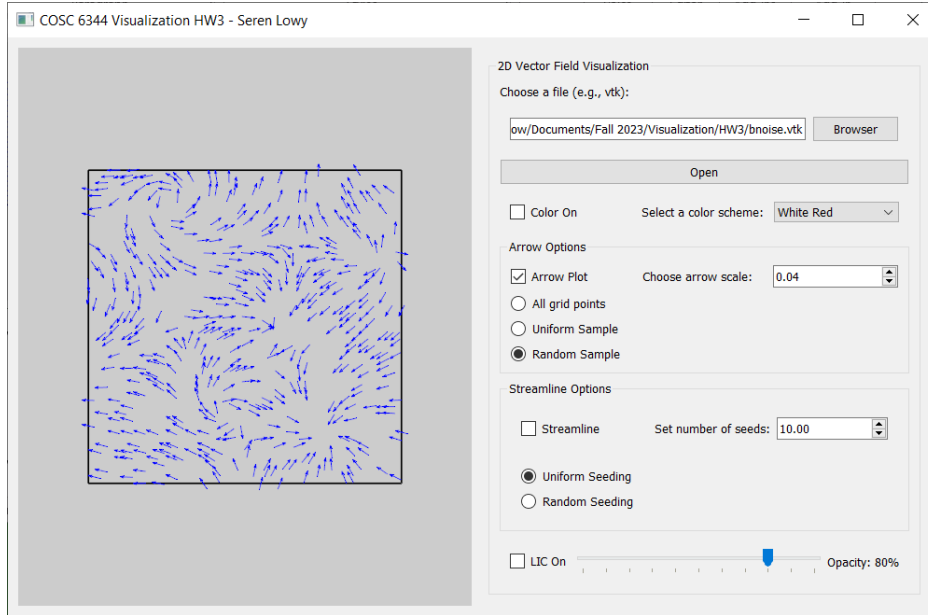
bnoise.vtk



All arrows

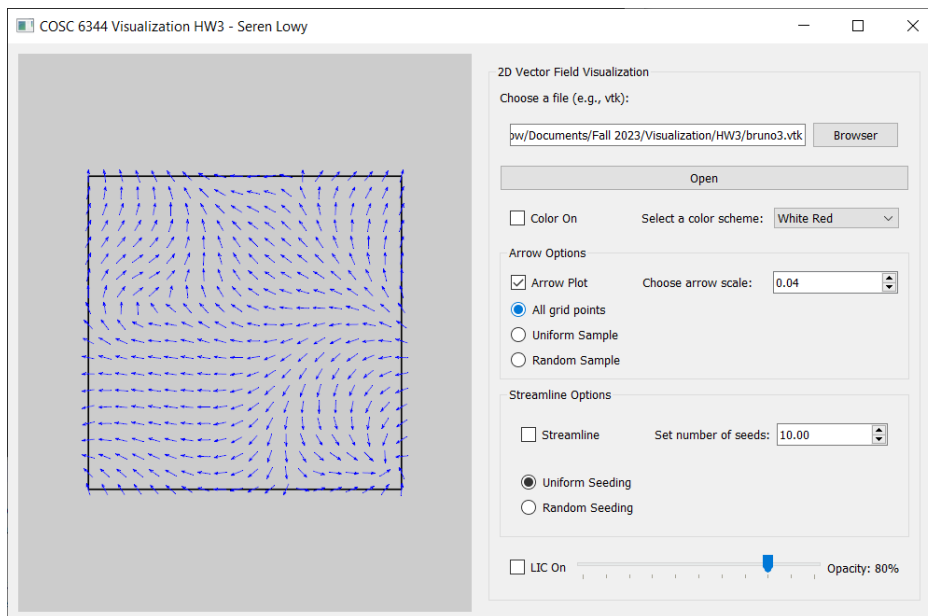
## Visualization Homework 3

Seren Lowy



Random down sample (500 arrows)

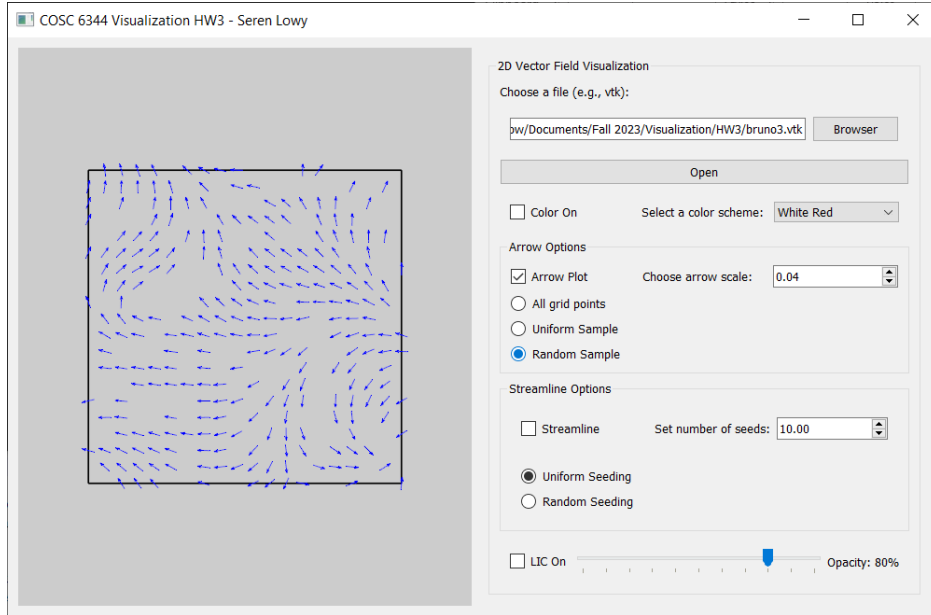
bruno3.vtk



All arrows

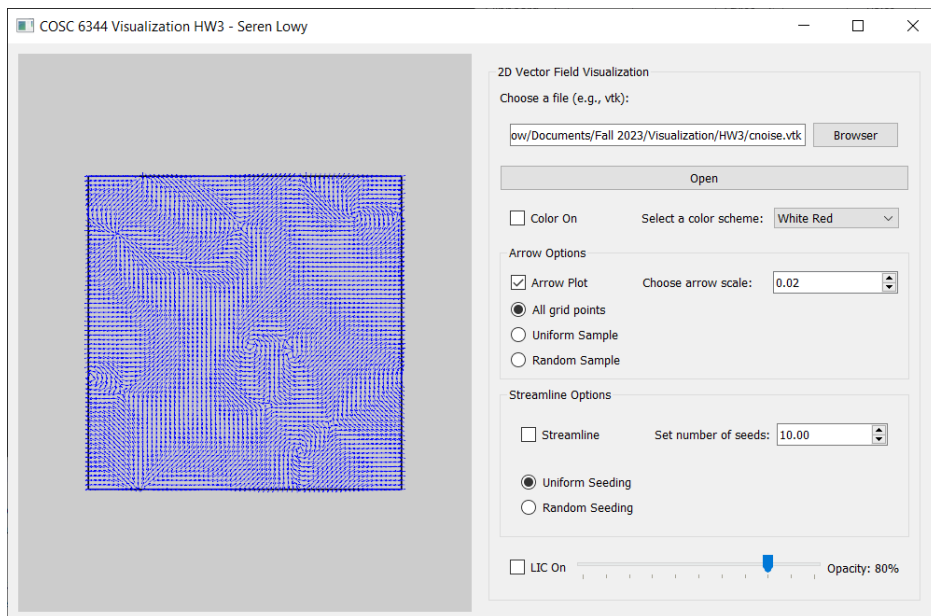
## Visualization Homework 3

Seren Lowy



Random down sample (500 arrows)

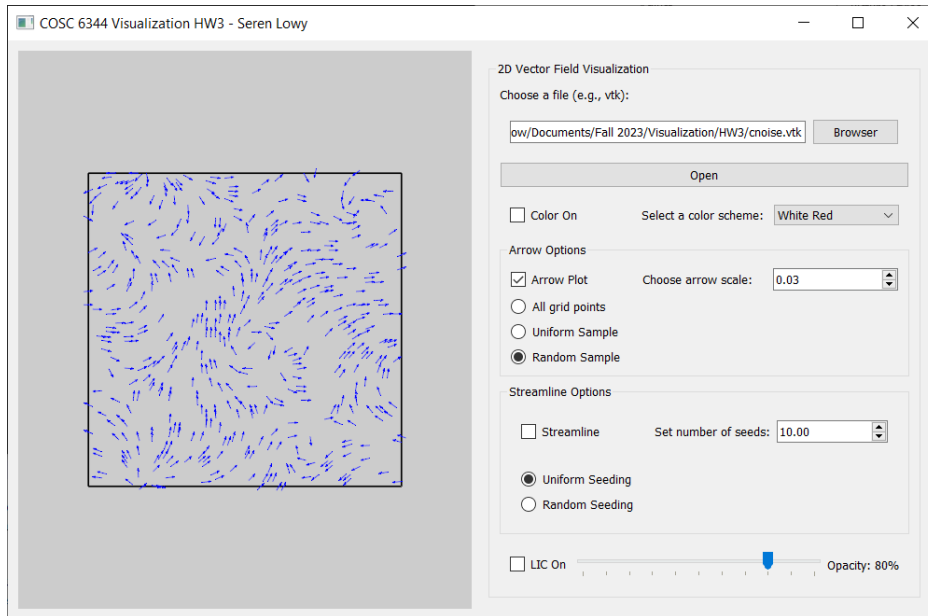
cnoise.vtk



All arrows

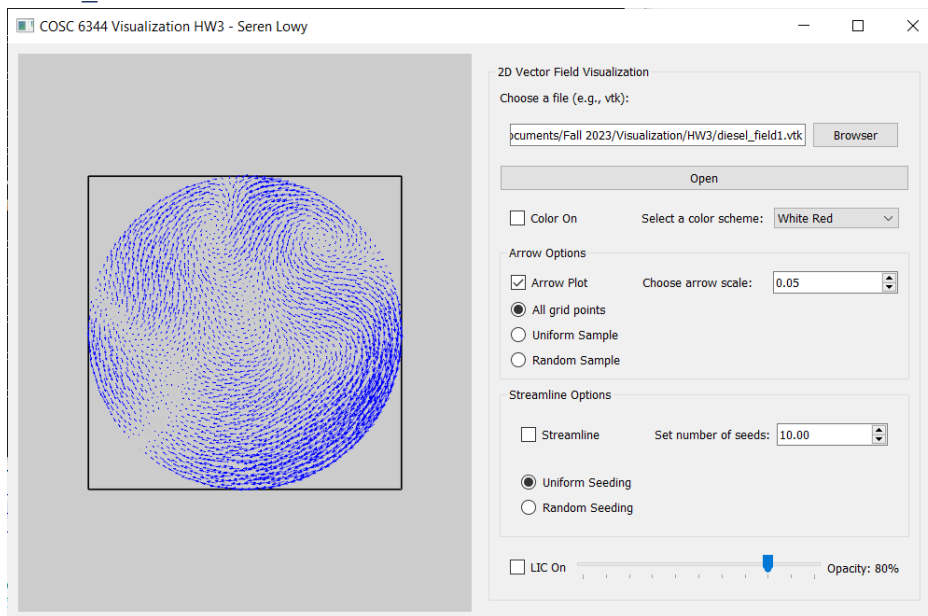
## Visualization Homework 3

Seren Lowy



Random down sample (500 arrows)

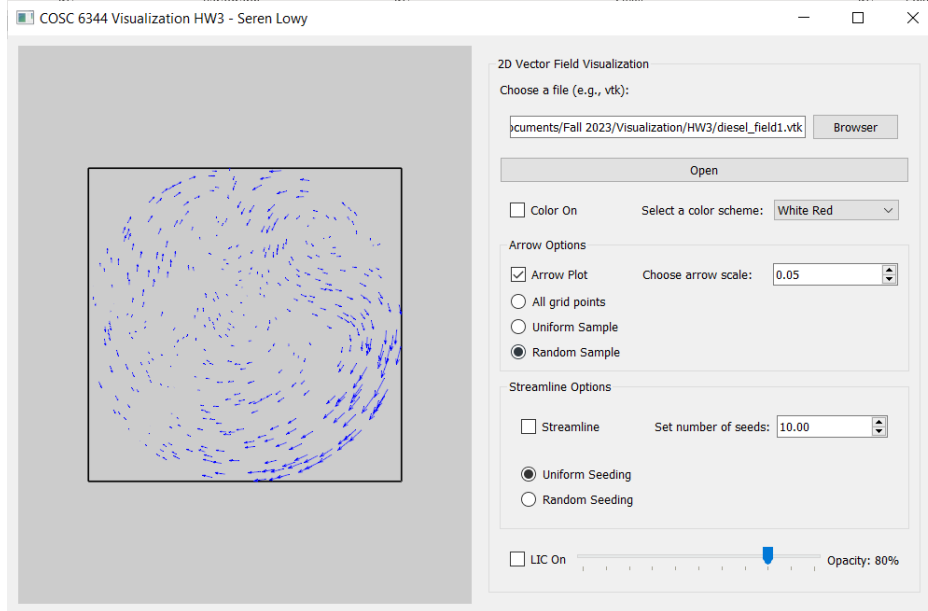
diesel\_field1.vtk



All arrows. The diesel\_field1 dataset appears to have vectors with different magnitudes.

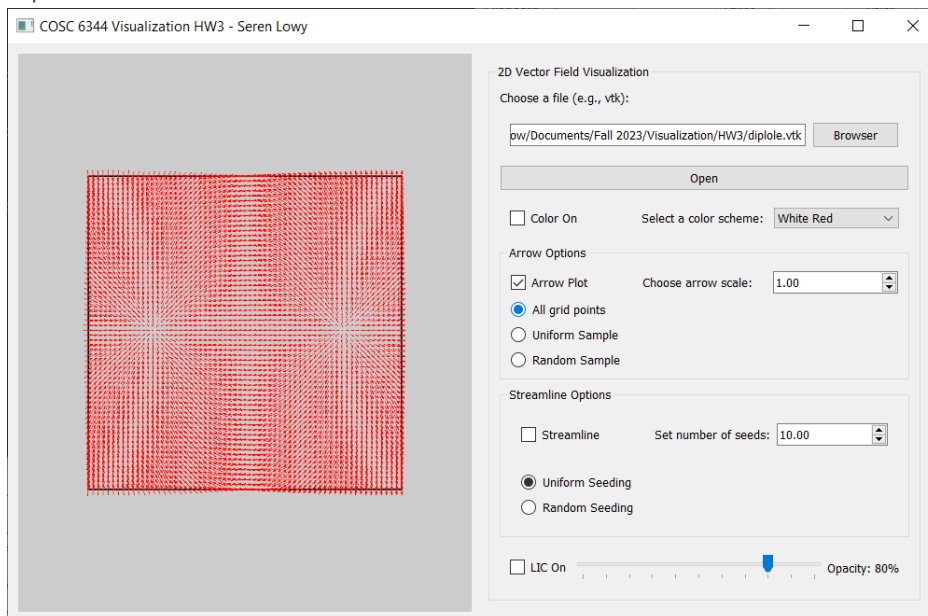
## Visualization Homework 3

Seren Lowy



Random down sample (500 arrows)

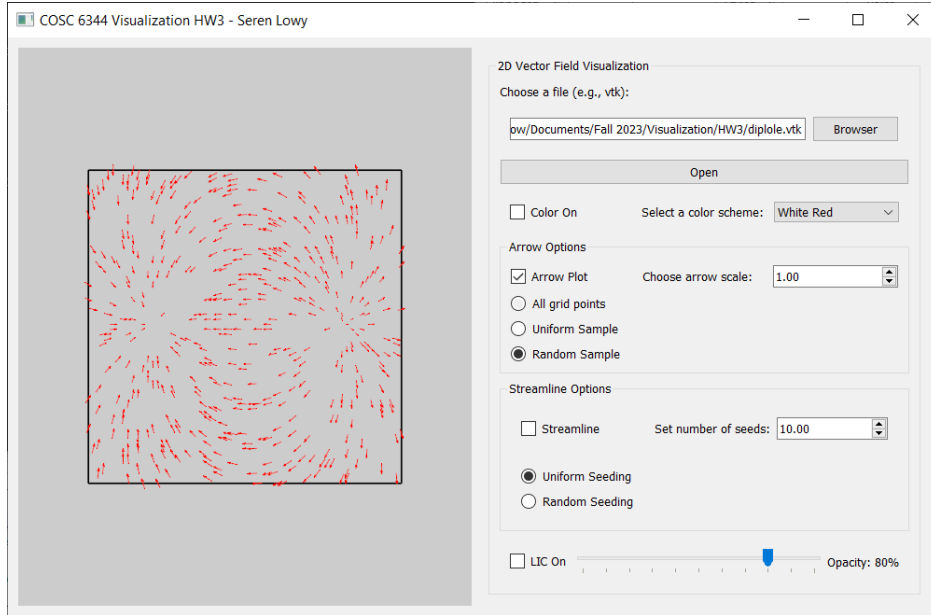
dipole.vtk



All arrows. The “dipole.vtk” dataset appears to have vectors with very low magnitudes. The arrow scale factor was set to 1.00 (much higher than other datasets) to make the arrows visible.

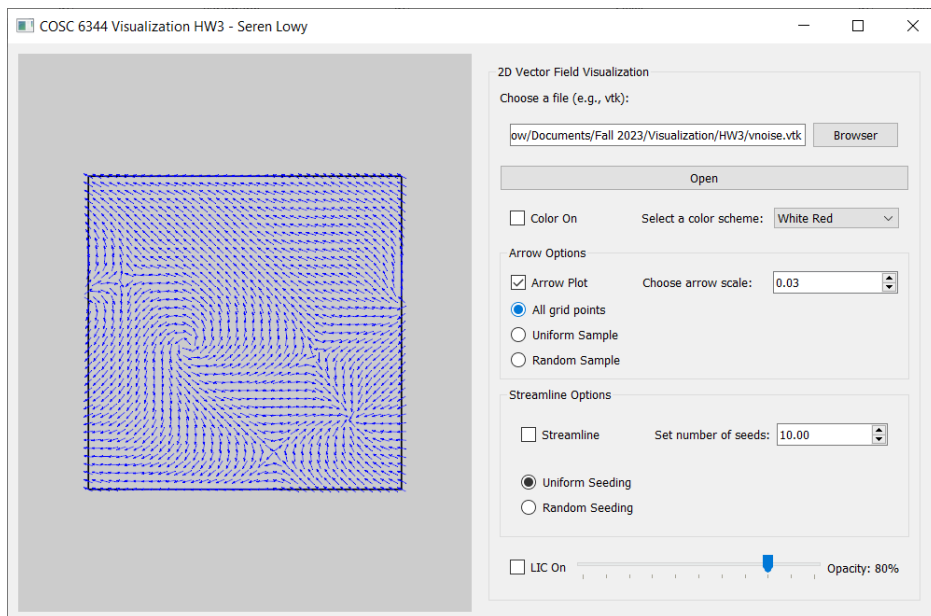
## Visualization Homework 3

Seren Lowy



Random down sample (500 arrows)

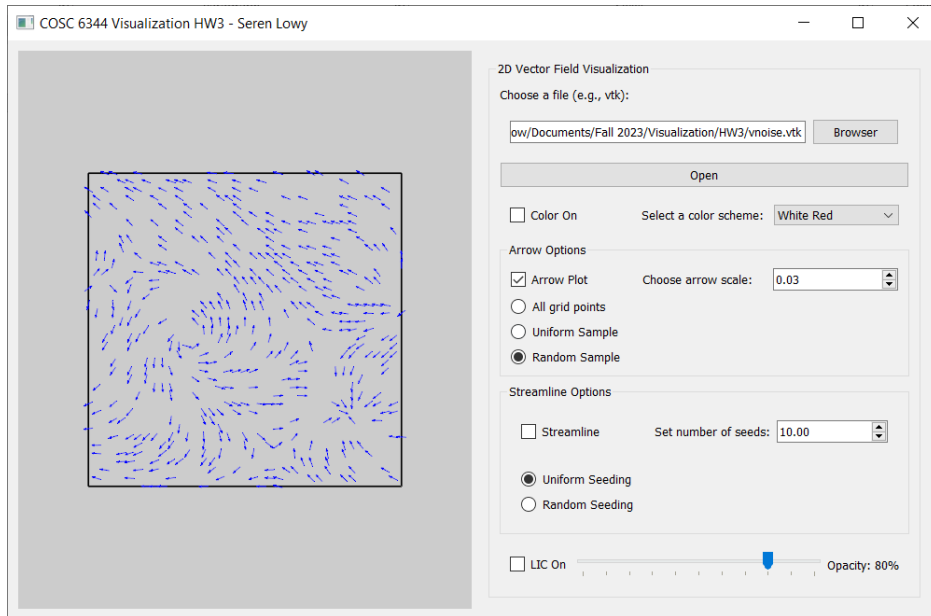
vnoise.vtk



All arrows

## Visualization Homework 3

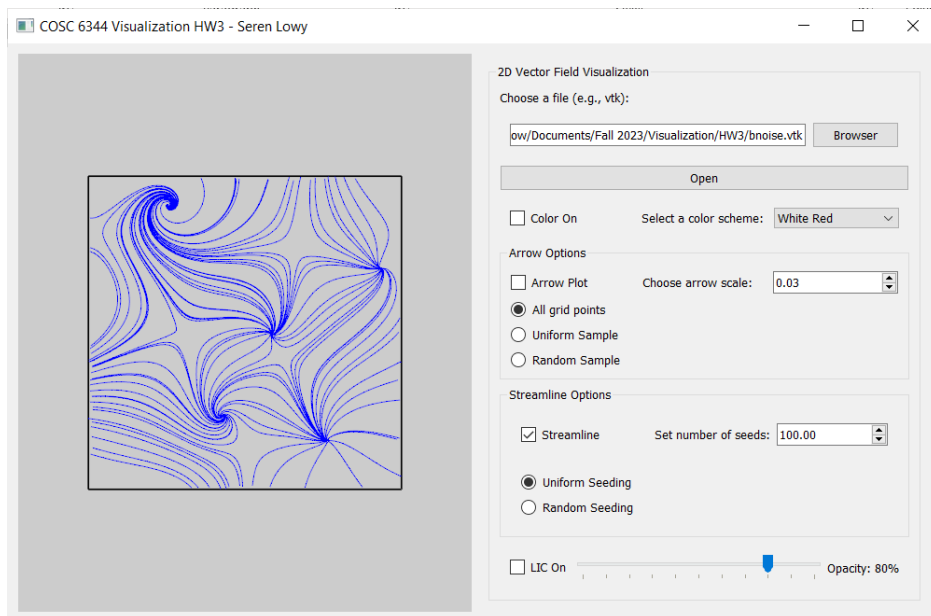
Seren Lowy



Random down sample (500 arrows)

### 3. Streamline plots

bnoise.vtk

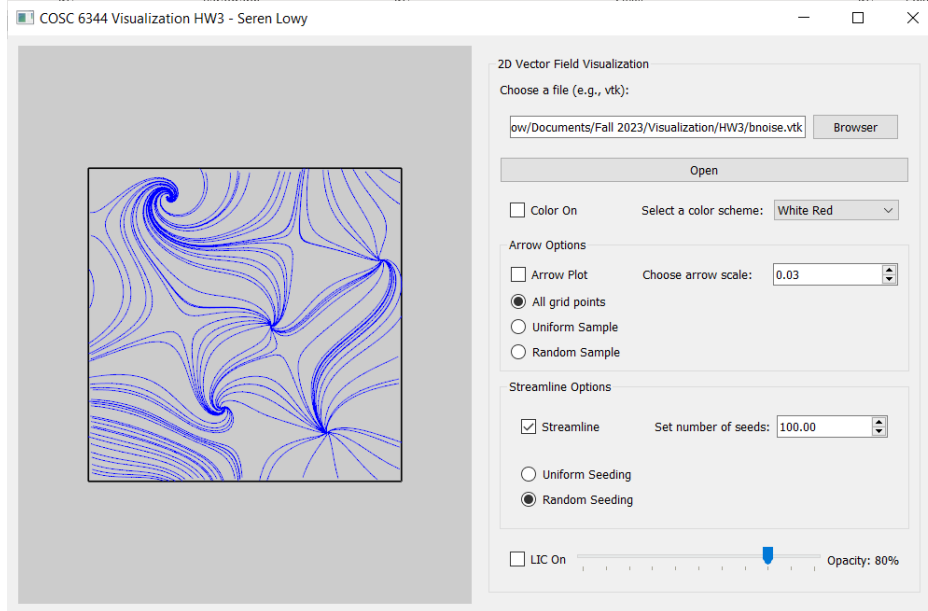


Uniform seed placement (100 seeds)



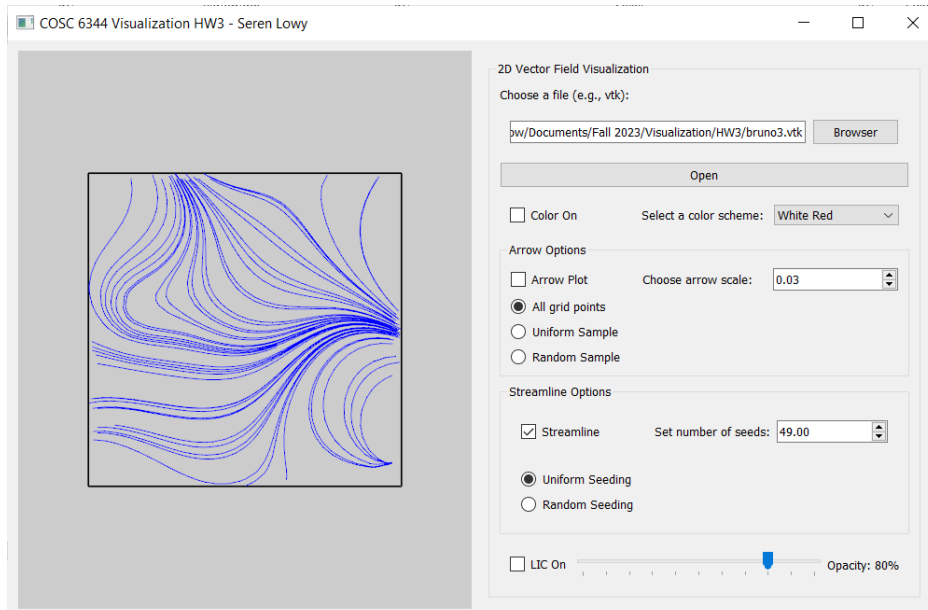
## Visualization Homework 3

Seren Lowy



Random seed placement (100 seeds)

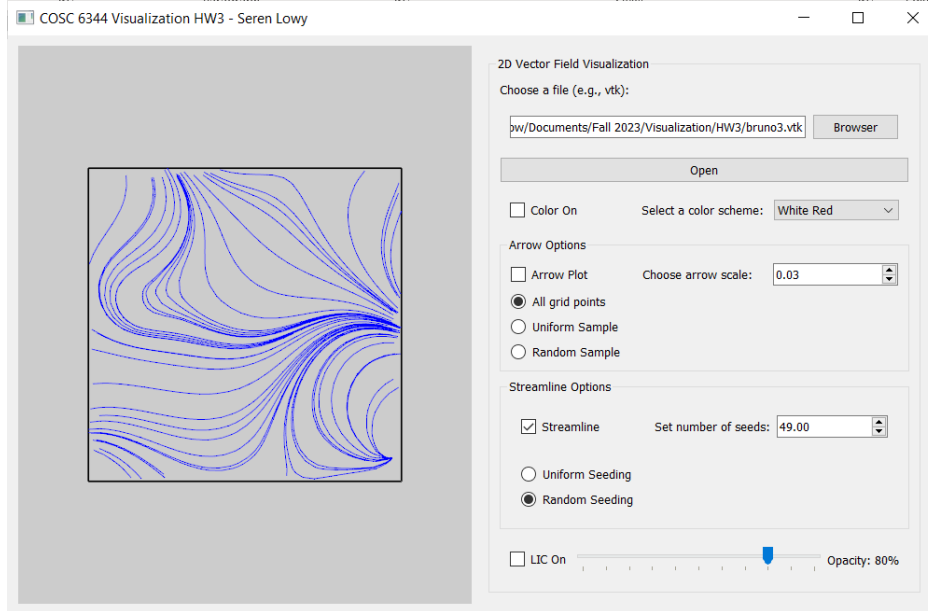
bruno3.vtk



Uniform seed placement (49 seeds)

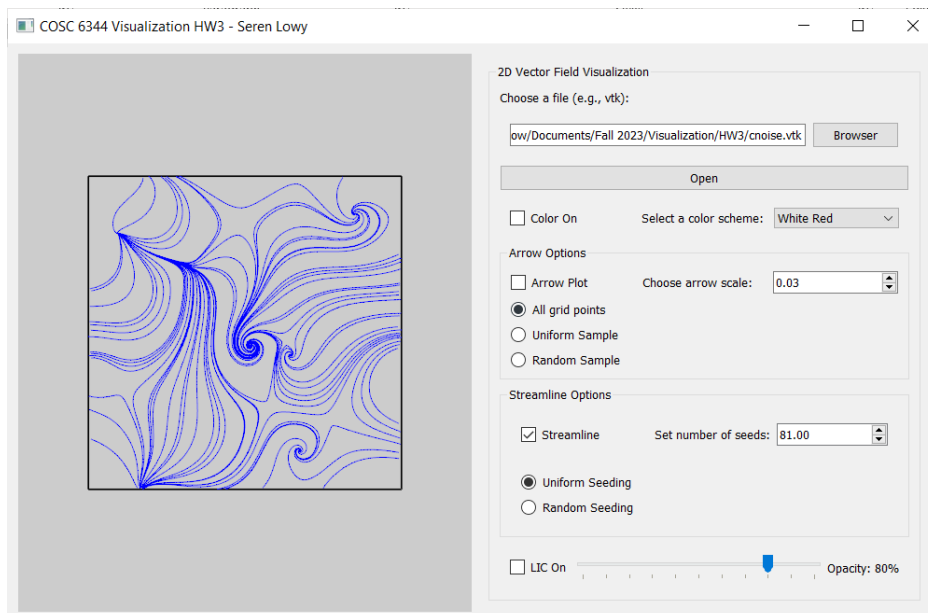
## Visualization Homework 3

Seren Lowy



Random seed placement (49 seeds)

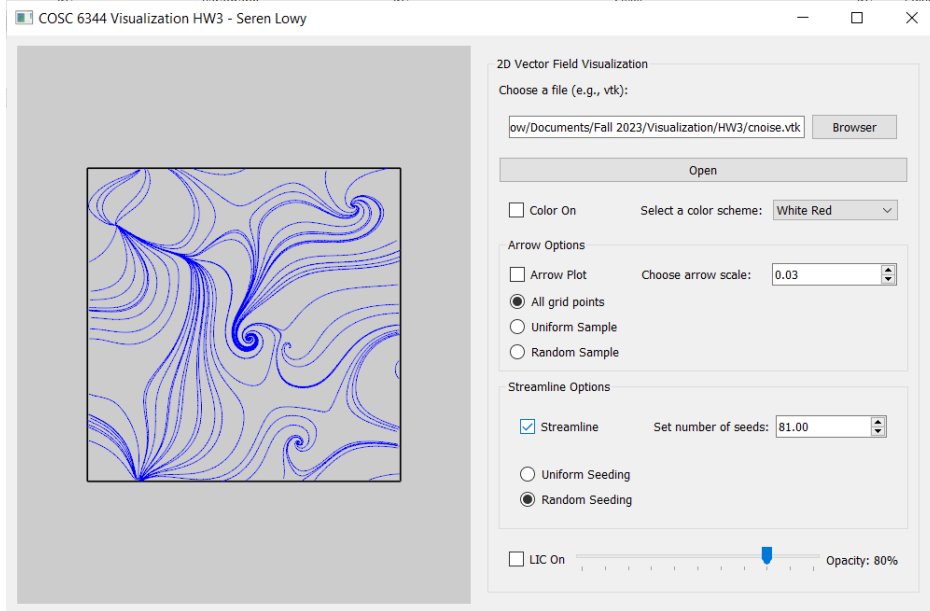
cnoise.vtk



Uniform seed placement (81 seeds)

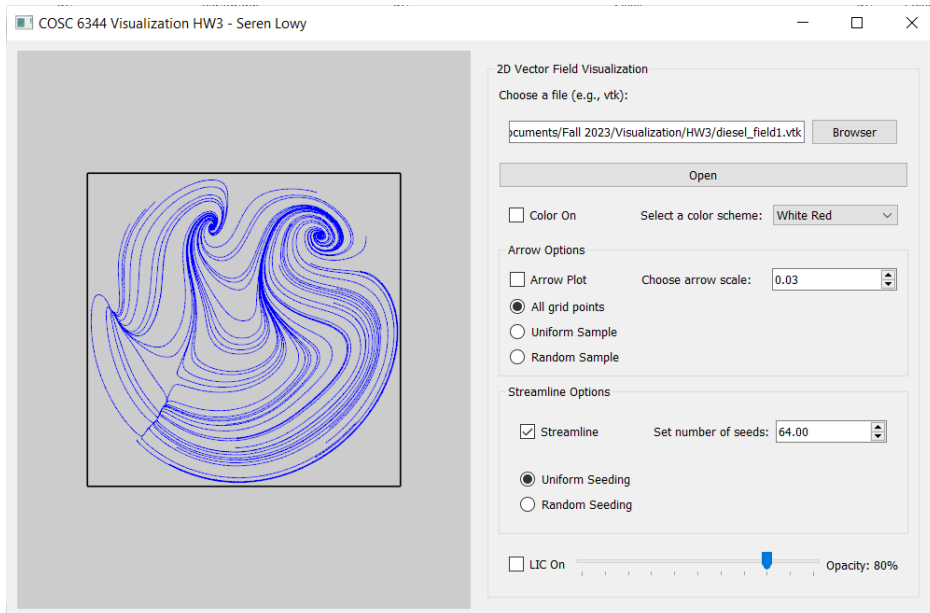
## Visualization Homework 3

Seren Lowy



Random seed placement (81 seeds)

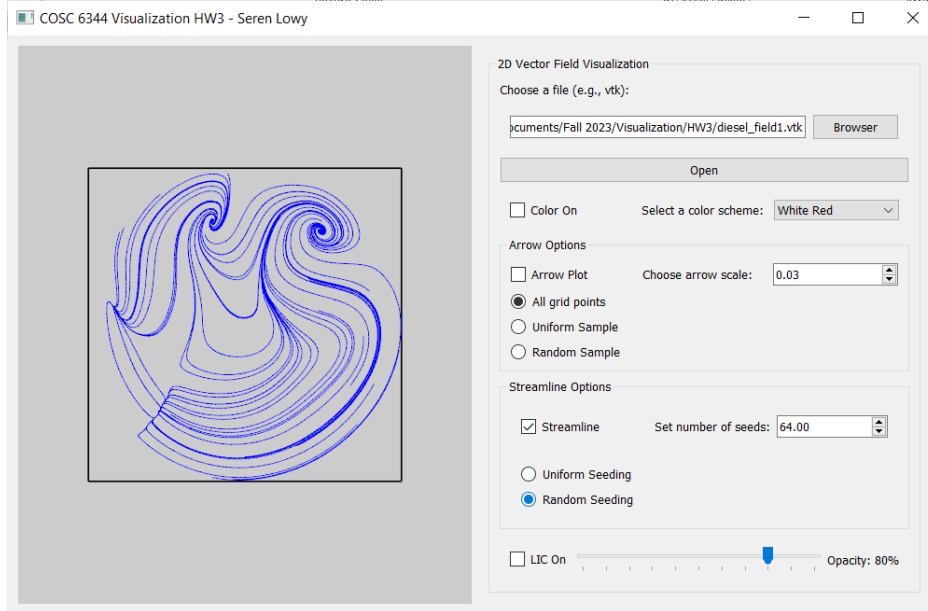
diesel\_field1.vtk



Uniform seed placement (64 seeds)

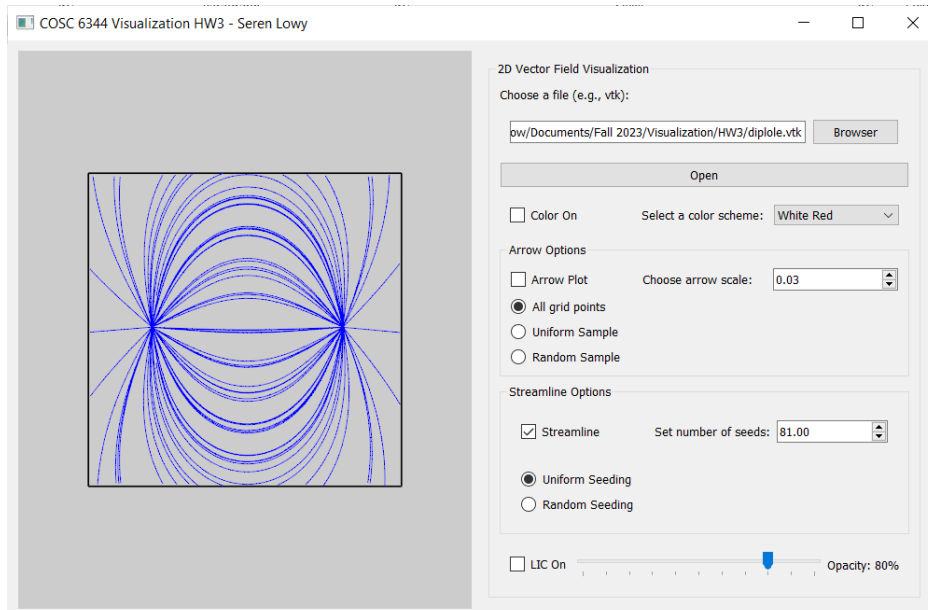
## Visualization Homework 3

Seren Lowy



Random seed placement (64 seeds)

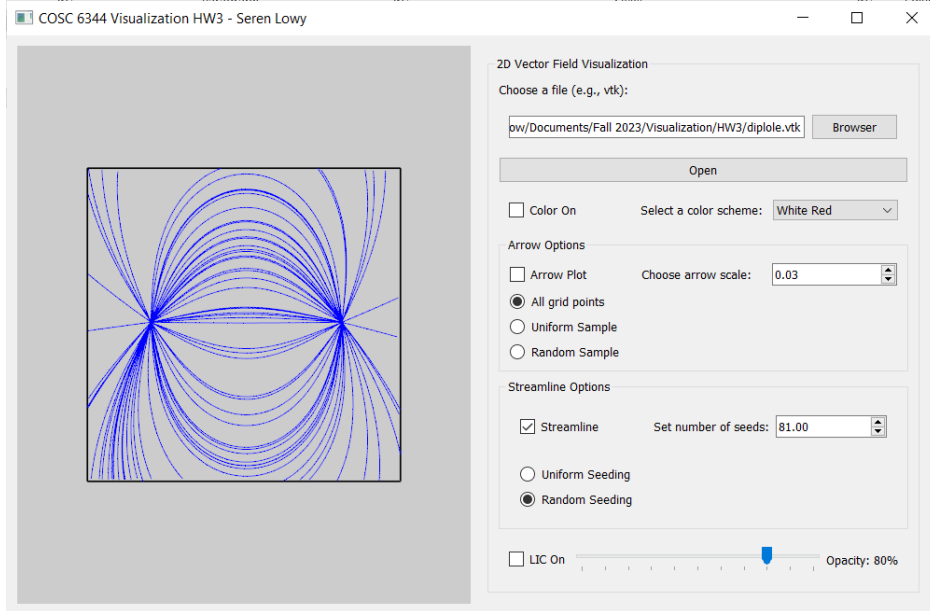
dipole.vtk



Uniform seed placement (81 seeds)

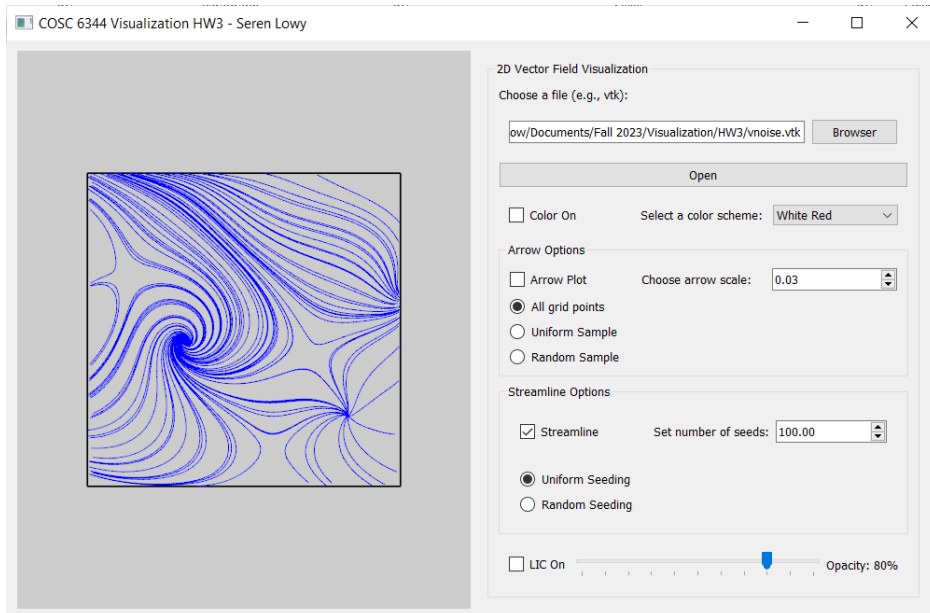
## Visualization Homework 3

Seren Lowy



Random seed placement (81 seeds)

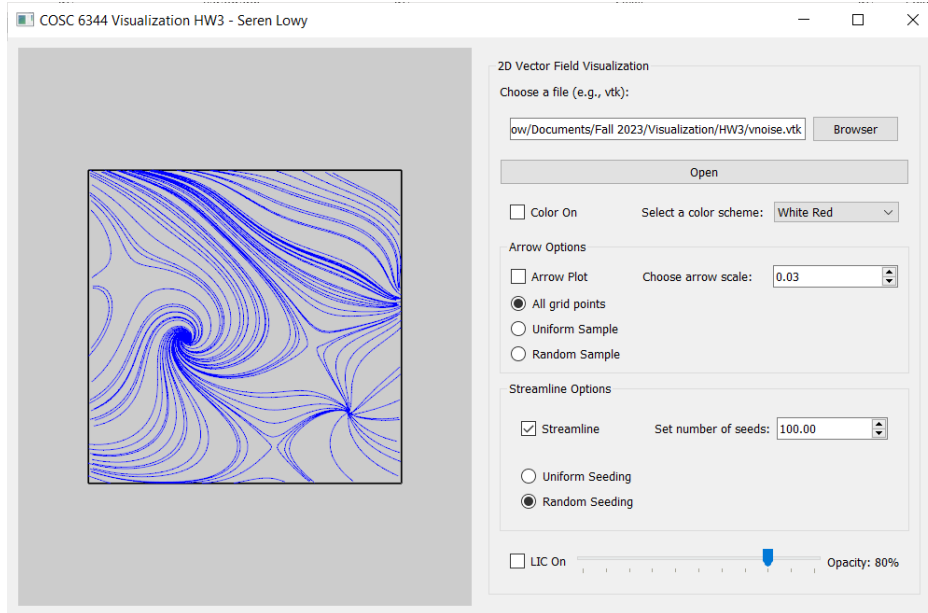
vnoise.vtk



Uniform seed placement (100 seeds)

## Visualization Homework 3

Seren Lowy



Random seed placement (100 seeds)

Comment: In all datasets, both types of seed placement (uniform and random) produce plots with disruptive bunching of streamlines. The random seed placement is usually not much better than the uniform seed placement, because there is still no criteria that respects previously drawn lines and moves or stops lines when they get close to each other.

### 4. LIC textures

Parameter settings for all:

Step size = 0.5 units (\* in units of cell diagonal)

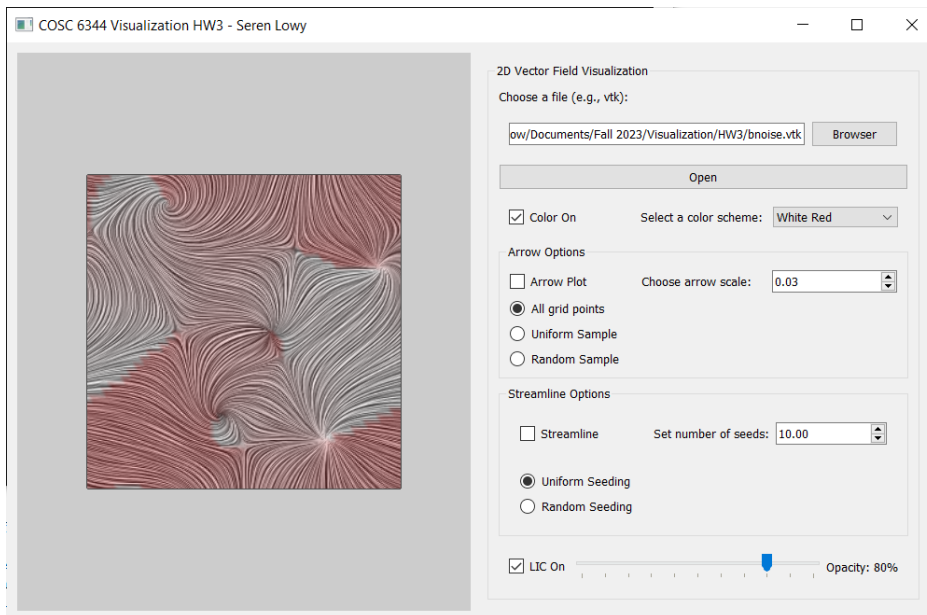
Step count = 30 many steps

Opacity = 80%

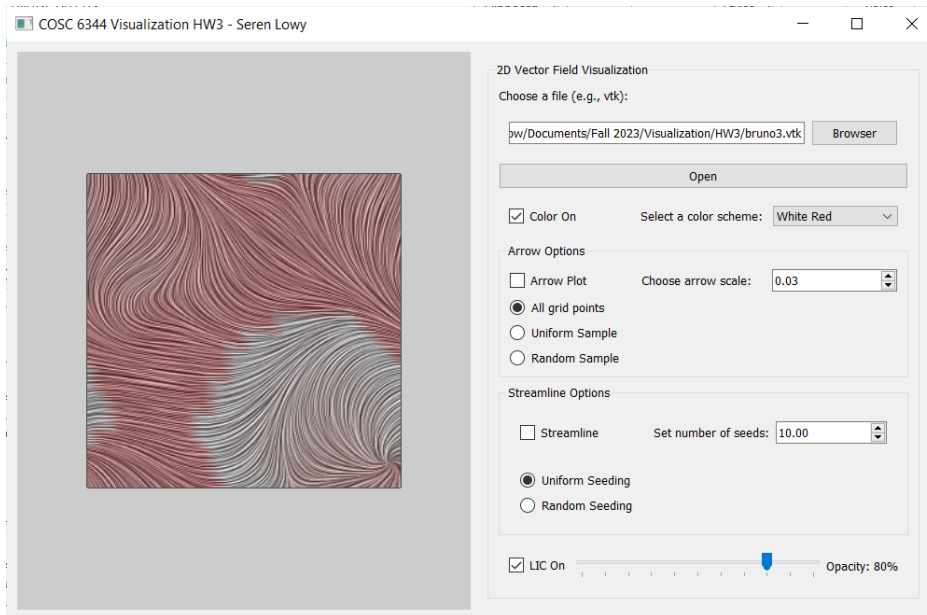
# Visualization Homework 3

## Seren Lowy

bnnoise.vtk



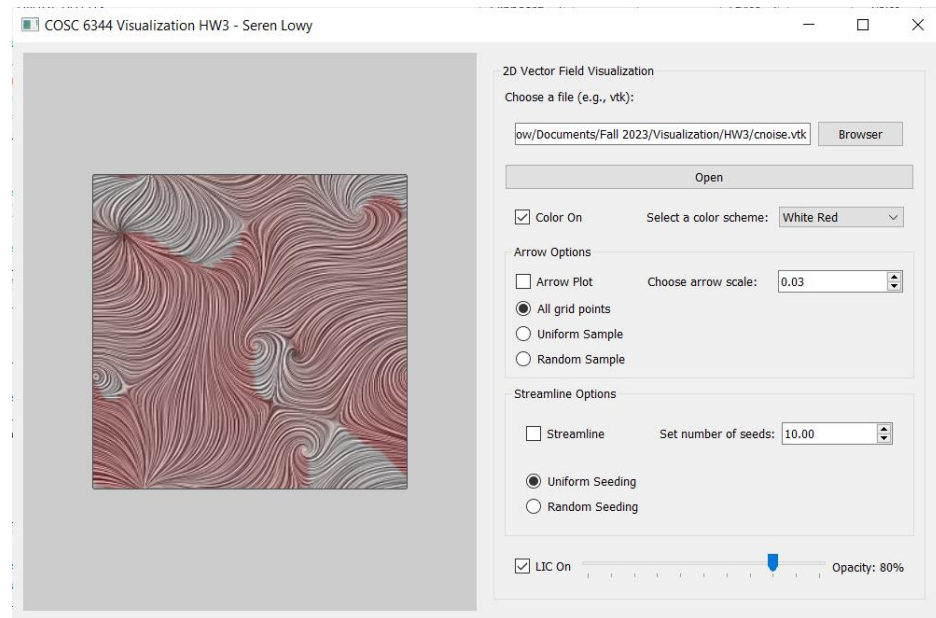
bruno3.vtk



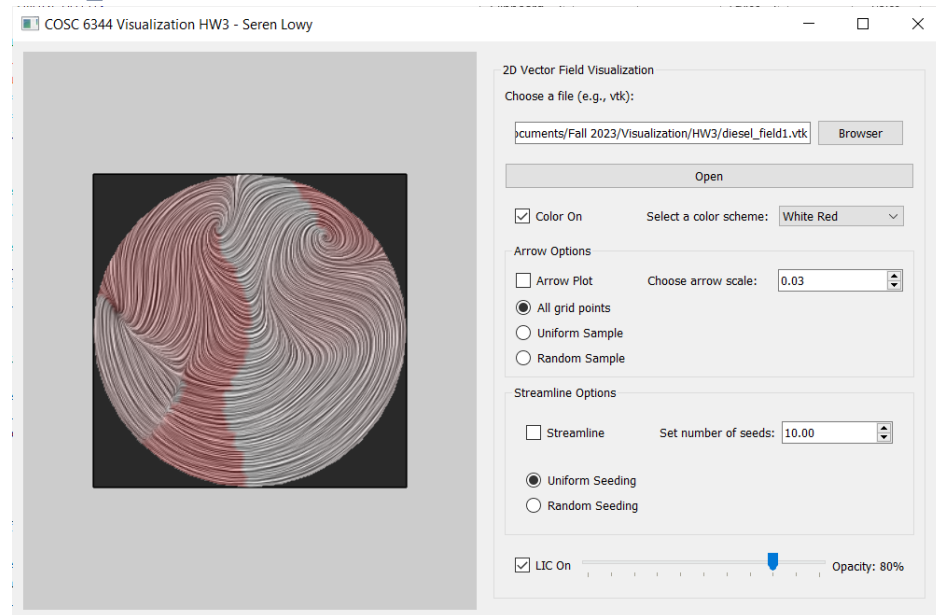
## Visualization Homework 3

Seren Lowy

cnoise.vtk



diesel\_field1.vtk

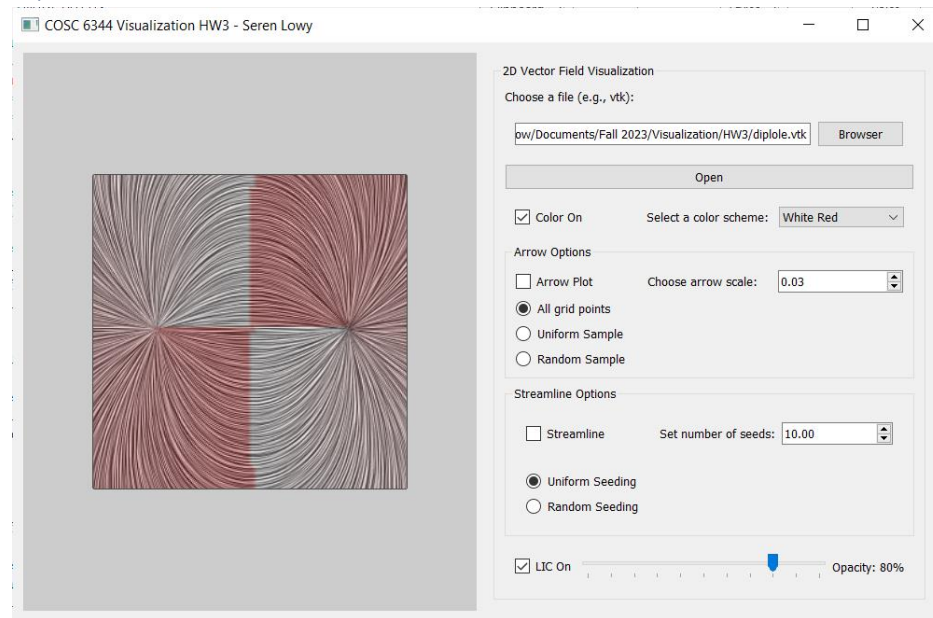




# Visualization Homework 3

## Seren Lowy

dipole.vtk



vnoise.vtk

