

Assignment #4: 2D Steady Vector Field Visualization
COSC 6344 Visualization
Fall 2021
University of Houston

Raunak Sarbajna
1956665

October 21, 2021

Exercise 1

1.1 Visualizing vector fields is more challenging than visualizing scalar fields

There are multiple reasons:

- Vector fields have additions directional information, which need to be depicted without making it cluttered, while also depicting magnitude
- Scientific datasets are often very large and hard to parse.
- The visualization can be heavily time-dependent. The flows may be steady while static but heavily chaotic while in motion.

1.2 Pseudo-code for the LIC algorithm

Algorithm 1 LIC Pseudocode

Require: Vector Field V over a mesh

Require: Noise texture N

Ensure: Streamline $S \rightarrow [V]$ ▷ Streamlines in both forward and backward direction

for all $s \in S$ **do**

$Point(p) \leftarrow Center of Grid$

$p \leftarrow Colours_N(PixelSpace)$

$p \leftarrow Convolution(p)$

▷ get correlated pixel from the noise texture

▷ Could be Simple Average, Gaussian, etc.

end for

1.3 Features that people care about in vector fields

To do feature-based classification, several types of features are considered interesting: Features:

- Topology based features:
 - Relying on fixed points:
 - * Sink
 - * Source
 - * Saddle
 - Relying on periodic orbits:
 - * Attracting
 - * Repelling
- Physics based (non-topological) features:
 - Vortices
 - Separation Flows
 - Coherent Structures

Exercise 2

The assignment was done using

1. Python 3.8.3
2. vtk 9.0.3
3. PyQt5 5.15.4
4. VS Code 1.60.2

2.1 Arrow Plots

The Density filter was used to set the maximum number of arrows based on user input. The scale factor is also adjustable.

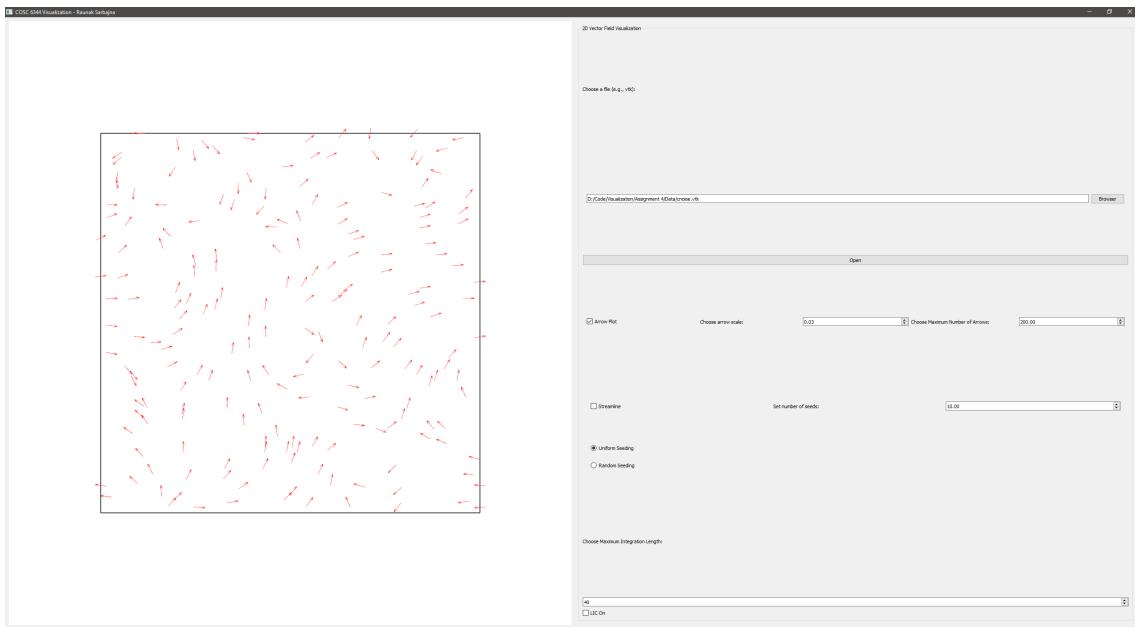


Figure 1: Arrow plot of cnoise data, with $ScaleFactor = 0.03, MaxPoints = 200$

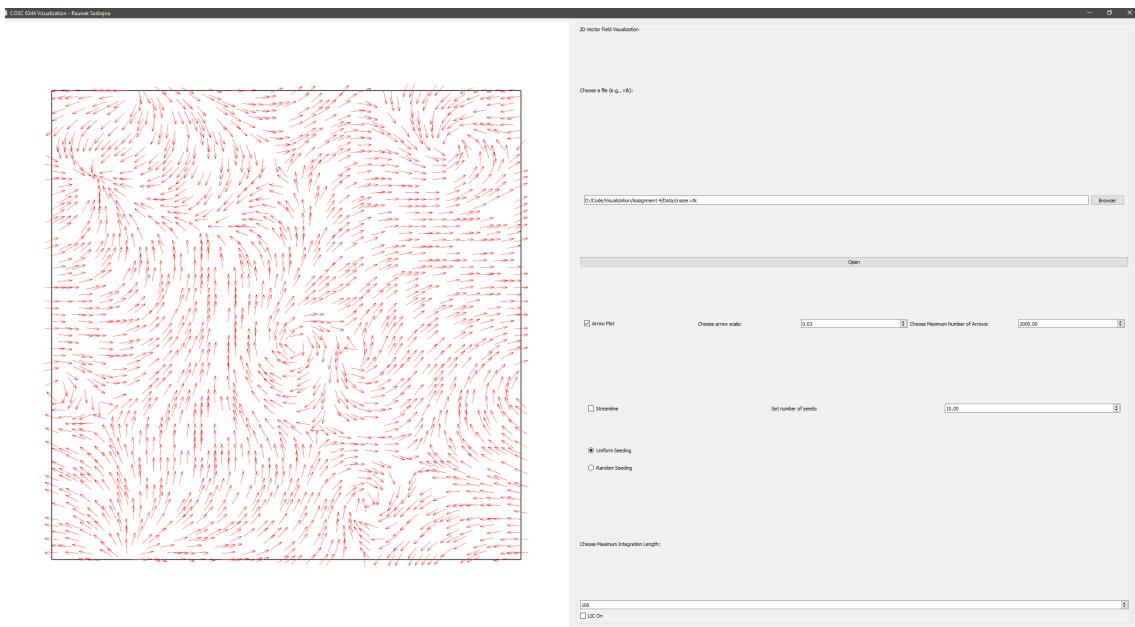


Figure 2: Arrow plot of cnoise data, with $ScaleFactor = 0.03, MaxPoints = 2000$

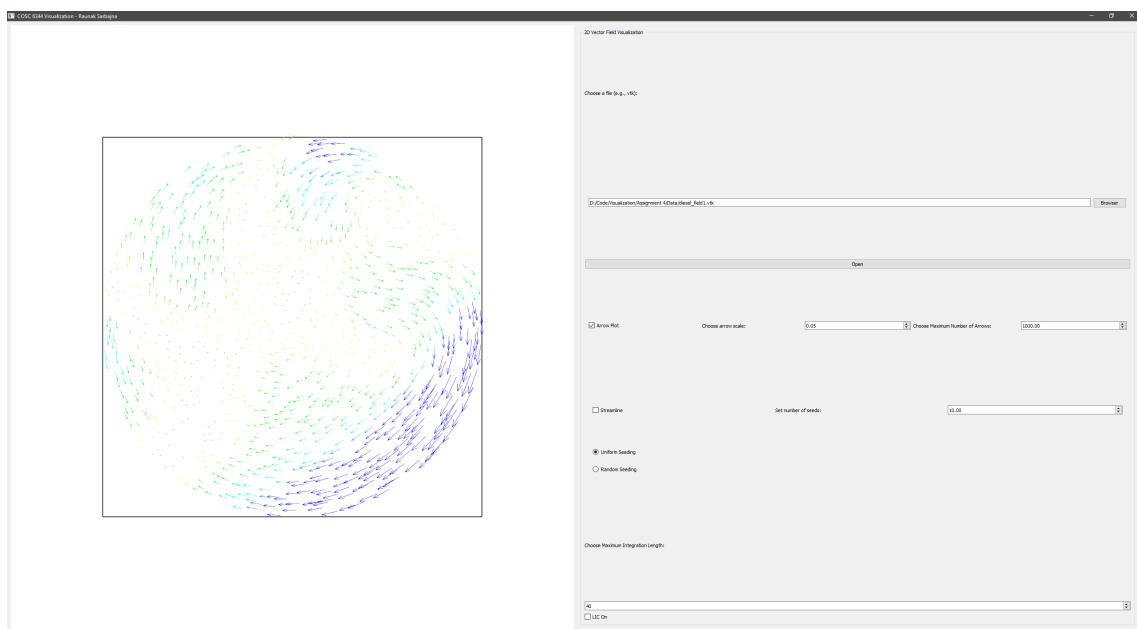


Figure 3: **Arrow plot of diesel field data, with $ScaleFactor = 0.05$, $MaxPoints = 1000$**

Exercise 3

3.1 Compute and visualize streamlines

3.1.1 Uniform seed generation

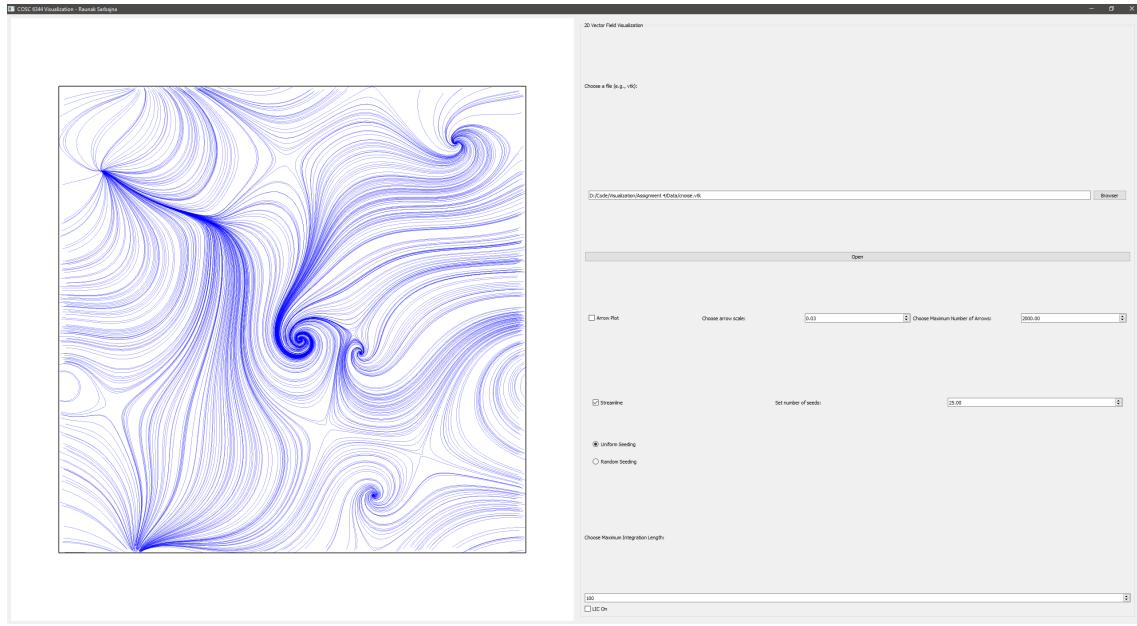


Figure 4: Uniform Streamlines of the cnoise data, with $N = 25$

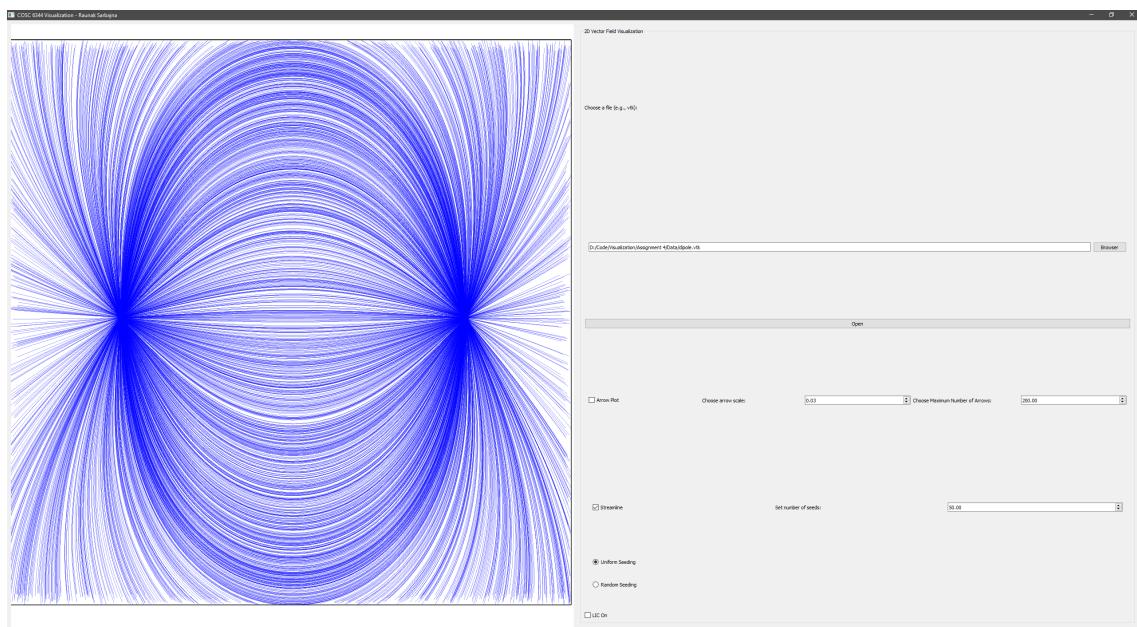


Figure 5: Uniform Streamlines of the dipole data, with $N = 50$

3.1.2 Random seed generation

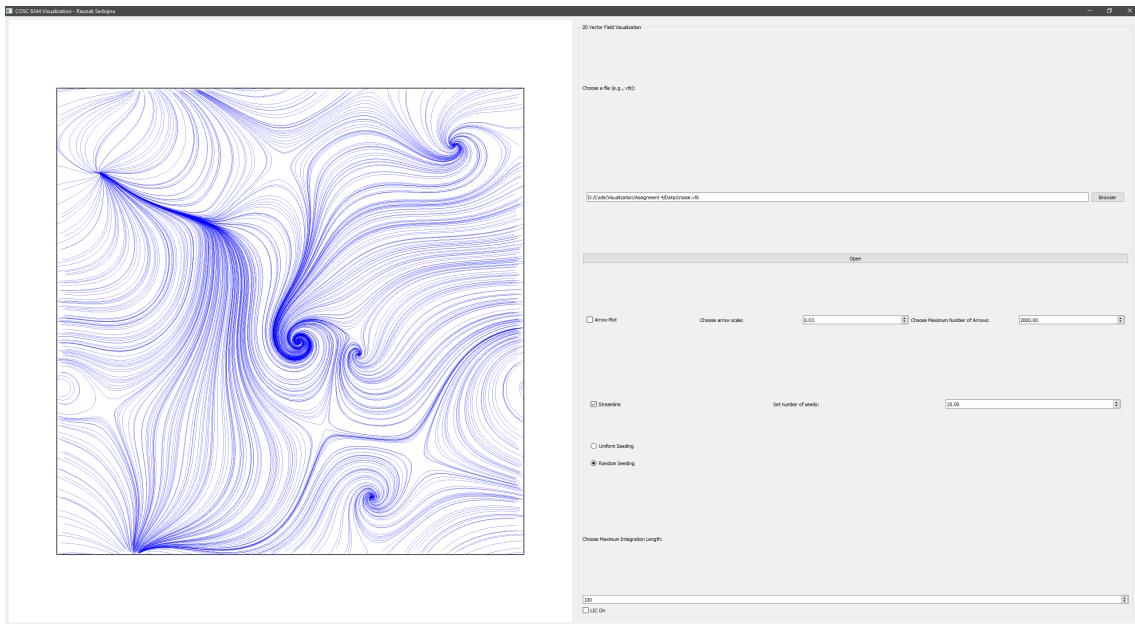


Figure 6: **Random Streamlines of the cnoise data, with $N = 25$**

Exercise 4

4.1 Compute and visualize LIC texture

The LIC colour plot was not performed due to unusually long compile times and image artifacts.

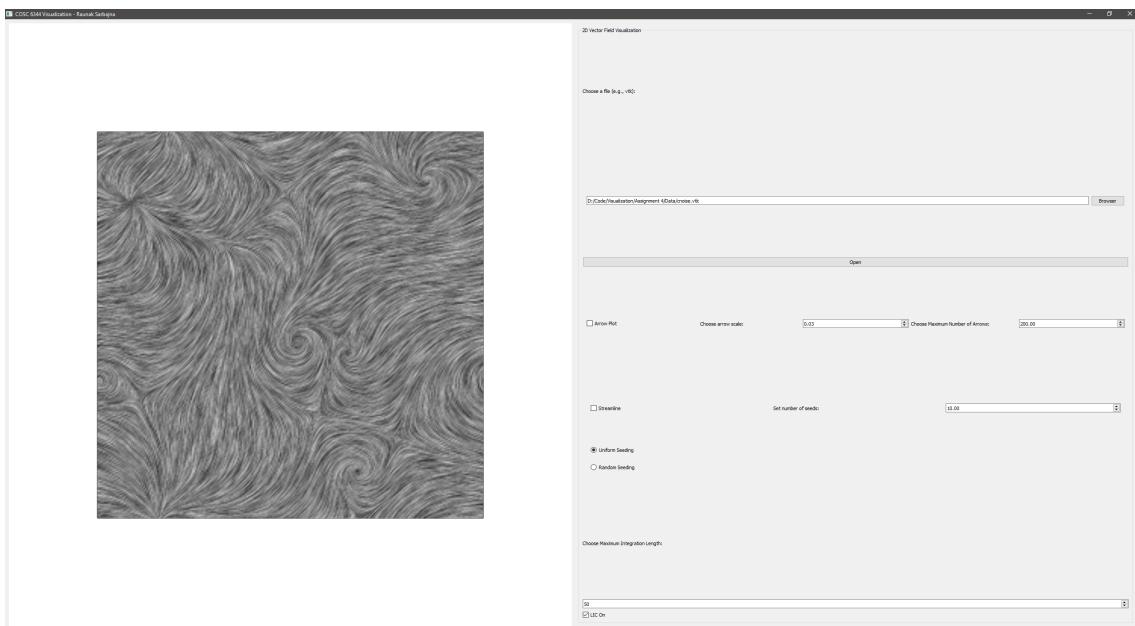


Figure 7: **LIC Textures of the cnoise data with $IntegrationLength = 50\%$**

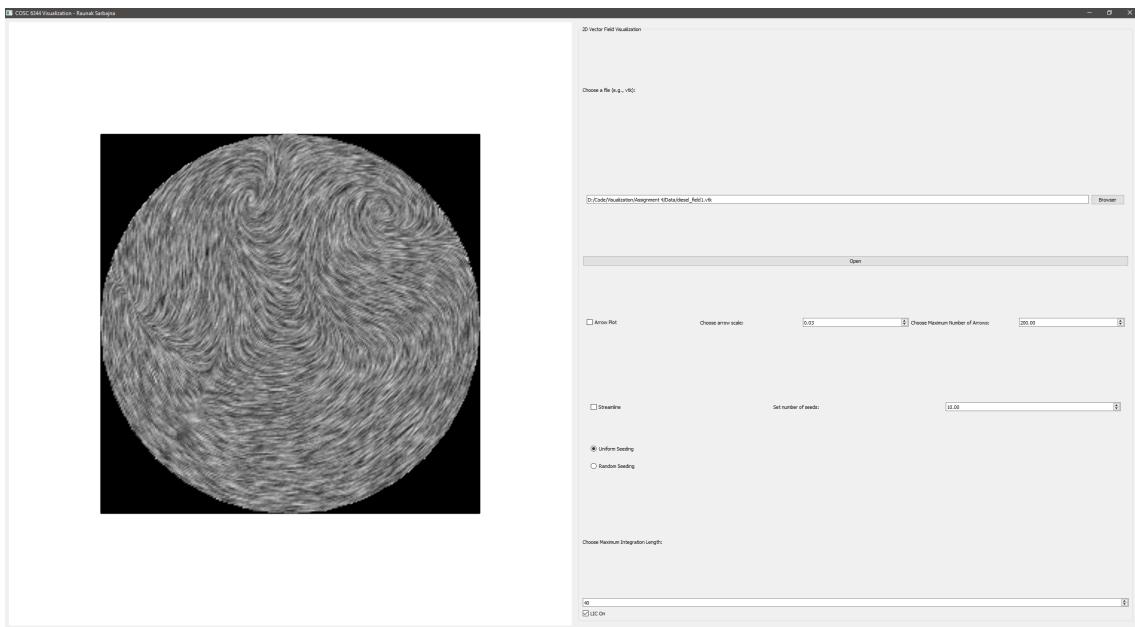


Figure 8: **LIC Textures of the diesel field data with $IntegrationLength = 40\%$**

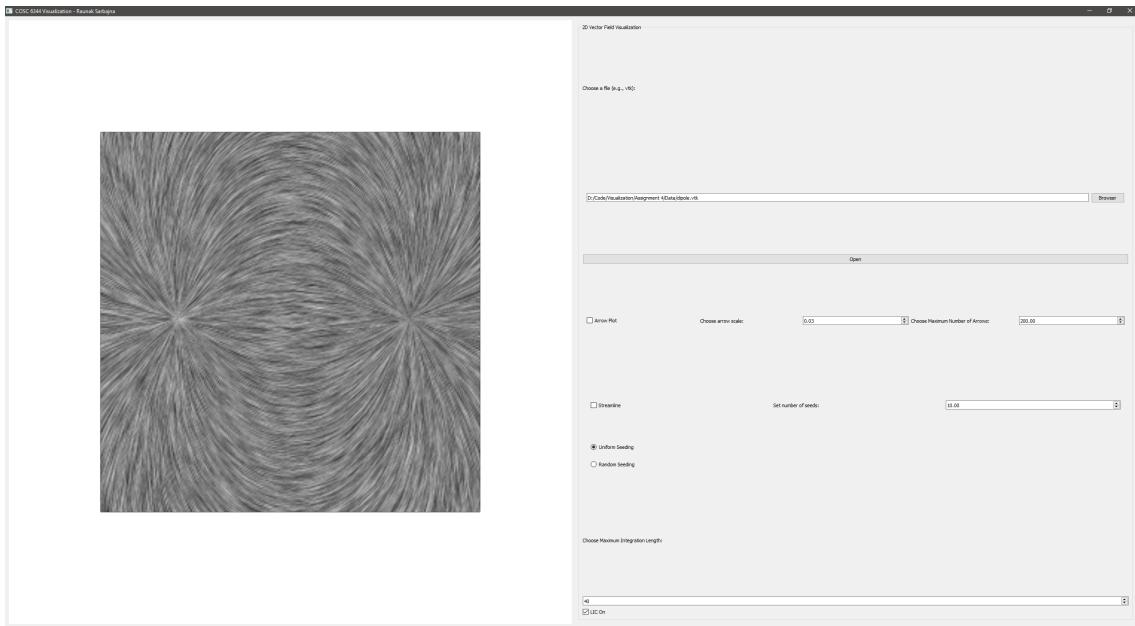


Figure 9: **LIC Textures of the dipole data with $IntegrationLength = 35\%$**