Assignment 06:
Uncertainty Visualization and Perception / Color

Scott Merkley

April/29/2023

Department of Computer Science, University of Utah, Salt Lake City, 84112, UT, USA

Introduction

This report covers the data and results for visualization of uncertainty and the importance it plays in modern day scientific visualization as well as preattentive processing, spaghetti vs contour box plots, and creating custom colormaps. This report is for Assignment 06 in CS 5635 at the University of Utah. Python, Paraview, and SciVisColor were used to create the visualizations in this report.

Part 1: Uncertainty Visualization of Isocontours

For the first part of this assignment, we were asked to generate some noisy data and visualize the uncertainty. We generated 100 different plots of $y(x) = sin(10\pi x) + sin(20\pi x)$, then added some noise into each using Gaussian noise with a mean of 0 and a standard deviation of $\frac{abs(y)}{2}$ to each array. 100 plots were generated using a for loop, noise was added, and then the function was plotted. The code for doing this can be seen here:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy as sp
import scipy.fftpack
import scipy.signal
import pandas as pd

def f(x):
    return np.sin(10 * np.pi * x) + np.sin(20 * np.pi * x)

# Make an array to plot everything on
n = 100
x = np.linspace(0, 1, n)

# Plot a bunch of random functions
numberOfArrays = 100
allPlots = []
for i in range(0, numberOfArrays):
    y = f(x) + np.random.normal(0, abs(f(x)) / 2, None)
    allPlots.append(y)
    plt.plot(x, y)
plt.title(f'{numberOfArrays} Random Arrays')
plt.xlabel('x')
plt.ylabel('y(x)')
plt.grid()
plt.show()
```

The plot generated from this data shows the variability in noise and can be seen here.

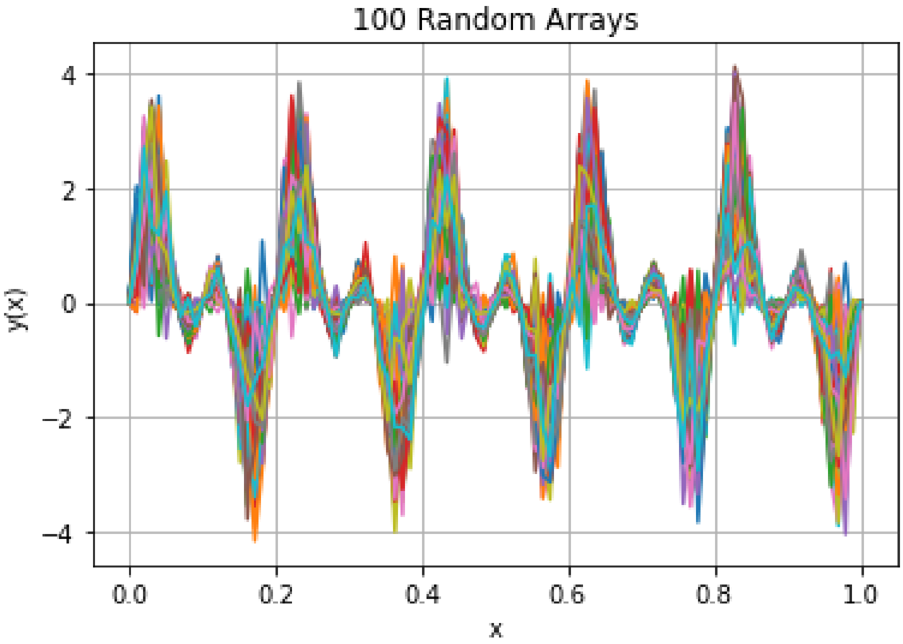


Figure 1: Visualization of 100 Randomly Generated Arrays with Gaussian Noise

Next, we were asked to find the mean of the data as well as one standard deviation from that mean and plot them. Using the Pandas data frame library, this was done using the pandas.Series.mean() and pandas.Series.std(). Doing this got the mean of every column in the data frame which contained the 100 random arrays, then the standard deviation of each column.

```
# Now plot the mean of all the different plots
allPlots = pd.DataFrame(allPlots)
meanPlots = pd.Series.mean(allPlots)
stdPlots = pd.Series.std(allPlots)

plt.title(f'Mean of Random Arrays')
plt.plot(x, meanPlots, 'tomato', label = 'Mean')
plt.fill_between(x, meanPlots - stdPlots, meanPlots + stdPlots, facecolor = '
                                                    lightskyblue', label = 'Standard Deviation
                                                    ')
plt.xlabel('x')
plt.ylabel('y(x)')
plt.legend()
plt.grid()
plt.show()
```

The plot showing the mean and standard deviation of the data can be seen here:

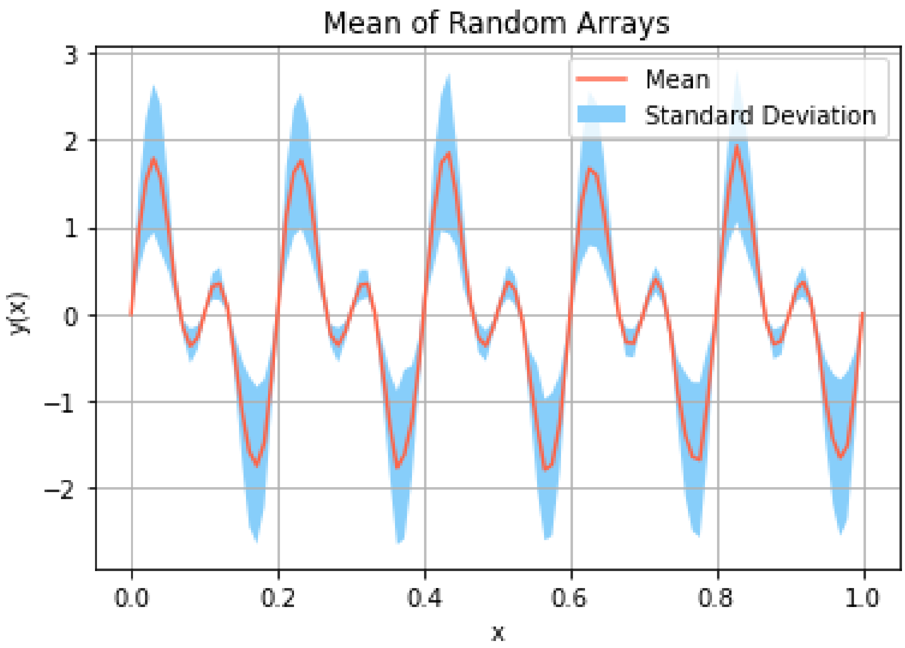
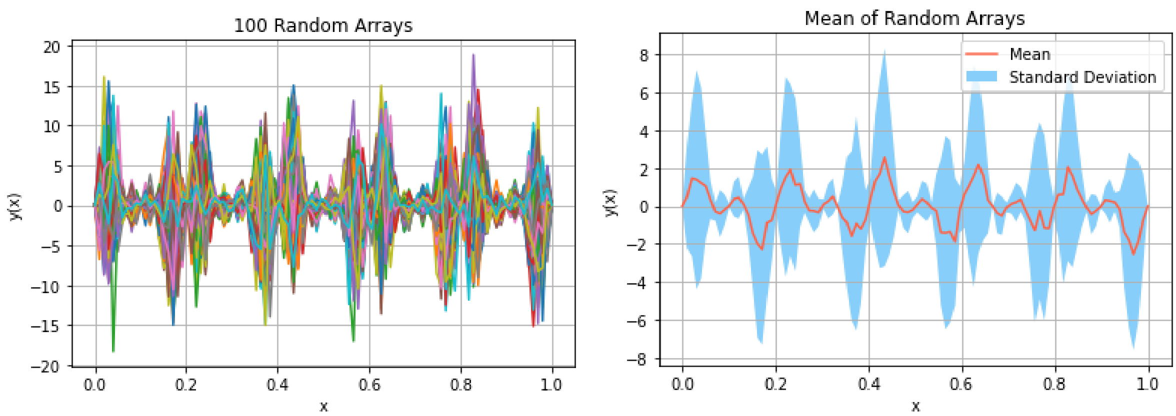


Figure 2: Visualization of the Mean and Standard Deviation of 100 Randomly Generated Arrays with Gaussian Noise

This plot represents the uncertainty in the data because the difference in standard deviation around the mean shows changes in values. Some peaks are higher, while others are wider. The variation in the data occurs from the Gaussian noise added to the values. This noise changes each value but the overall trend can still be seen quite clearly since the added noise is only values between [0, 1). If the values were to be much higher, the trends would be less easy to see.



(a) (b)

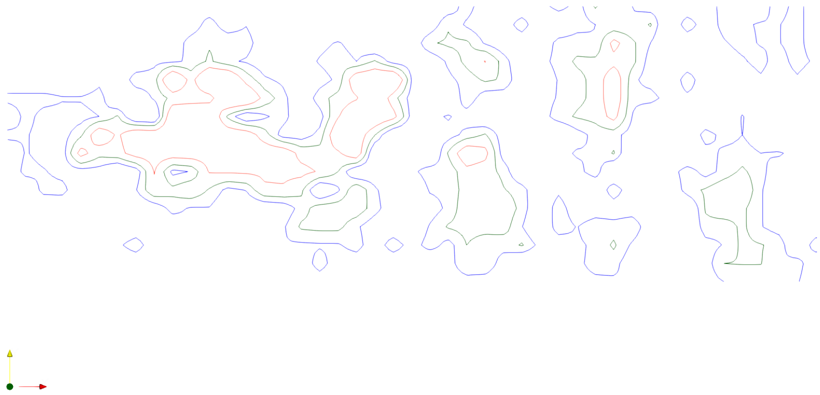
Figure 3: Plotting 100 Random Arrays with Noise from [0, 100)

In this visualization, the mean of the data can be seen to stray further from the underlying function used to generate the data. This makes it more difficult, if not impossible, to filter the data and extract the original function.

0.1 Generating a Spaghetti Plot

For this section of the assignment, we were asked to generate a spaghetti plot of isocontours for flow simulation data using Paraview. We loaded the provided flow data into Paraview, changed the coloring array to the different flow array and then was able to visualize the different plots.

(a)

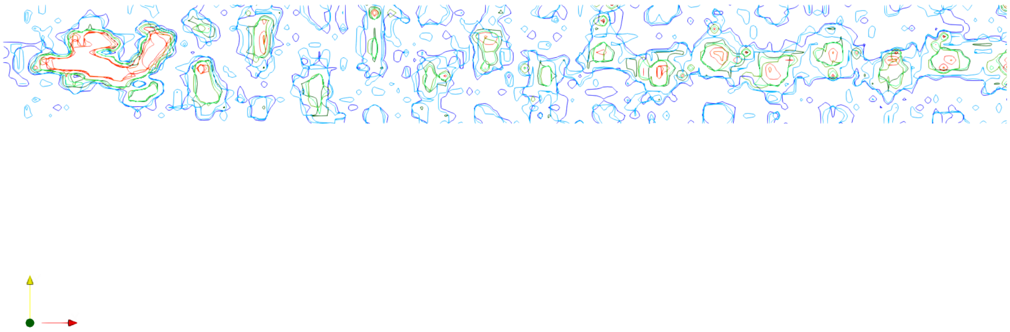


(b)

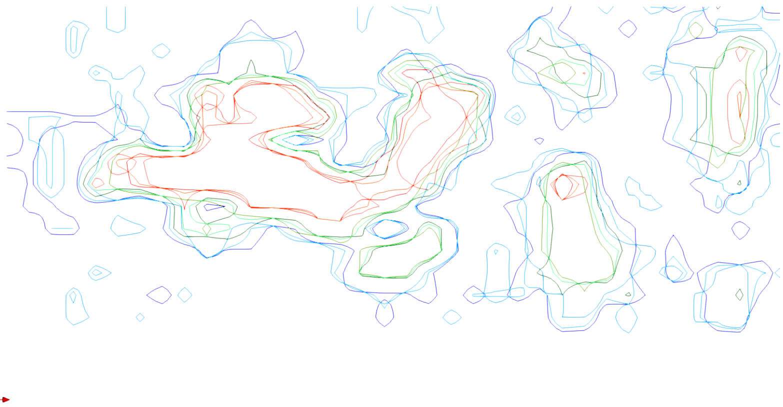
Figure 4: Visualizing 3 Isocontour Plots as a Spaghetti Plot

It was found that not all the isocontours coincide. Some of the isocontours are slightly off from each other and don't have the same lines, while others are completely off and overlap each other. A spaghetti plot makes this easy to see as the areas where the lines overlap and look tangled. In these tangles, there is seen to be more variation in the data and more interesting trends.

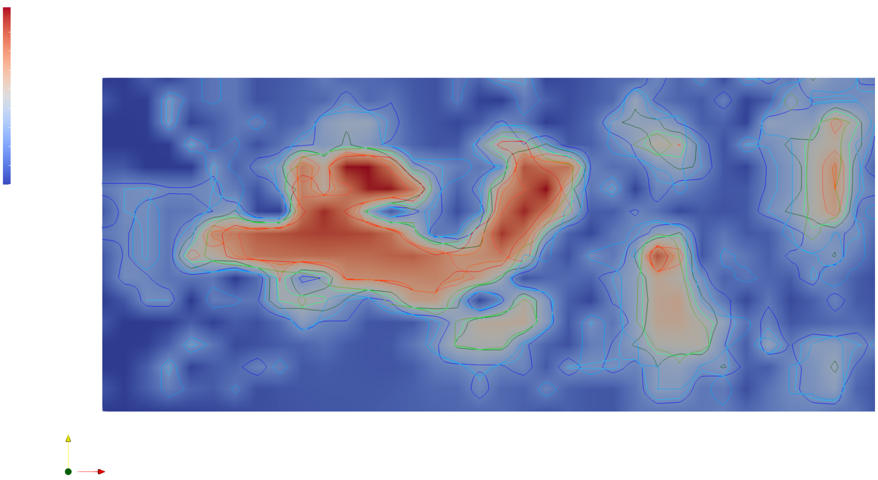
When stacking all isocontours on top of each other, the spaghetti plot starts to show a difference in overall trending as you can start to see all the many lines start to overlap each other. Areas of high activity can be seen as well as low activity.



(a)



(b)



(c)

Figure 5: Visualizing 9 Isocontour Plots as a Spaghetti Plot

The difficulty with spaghetti plots is that it is hard to see any general trends from them. It is easy to see where the interesting data lies but when trying to get a better understanding for the big picture, you need a different form of visualization.

When looking at contour box plots, you get a much better understanding for the general trends in data. Instead of seeing the all the little details, you get an understanding of where your data is heading in the future or where it has been in the past. Both types of plots are desirable as sometimes you want a better understanding of the little details.

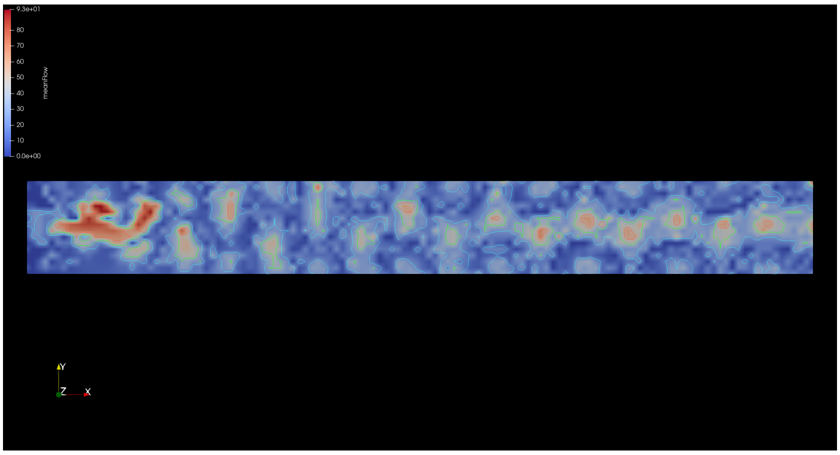
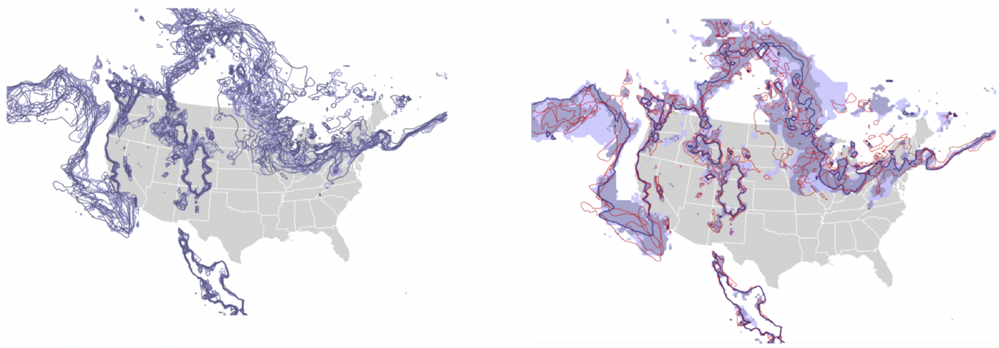


Figure 6: Visualization of the Mean Wind Data

Lastly, when looking at the differences between these two figures:



(a)Spaghetti plots

(b) Contour box plots

Fig: Uncertainty visualization of isocontours

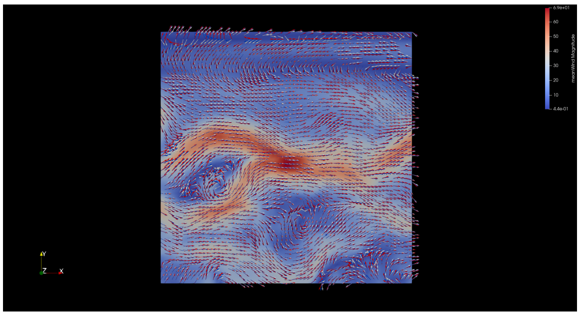
Median Isocontour

Outlier Isocontour

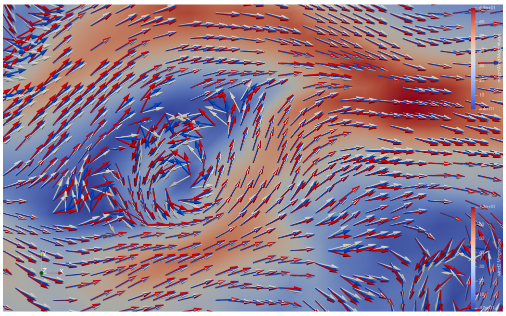
The contour box plot makes it a lot easier to see where that the areas of lower temperature are where the areas are colored in with purple. The spaghetti plot looks like a bunch of flow streams, which while nice as a way to plot, doesn't give the best understanding given that the plot is representing temperature. I personally find them both a little difficult to read since there are many areas on the plot that are not taken into account. There should be more values covering the whole plot and not just a small area of it.

Part 2: Uncertainty Visualization of a Vector Field

For part 2 of the homework we were asked to visualize the uncertainty in a vector field. The vector field we were given is a wind field and has 3 different data. We were asked to visualize each of the different data sets glyphs on top of each other. In doing this, the different trends can be seen in the field for the different data sets.



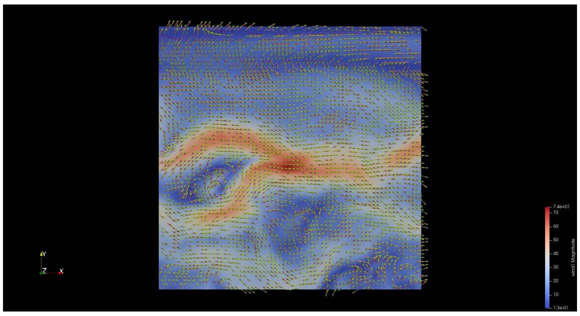
(a)



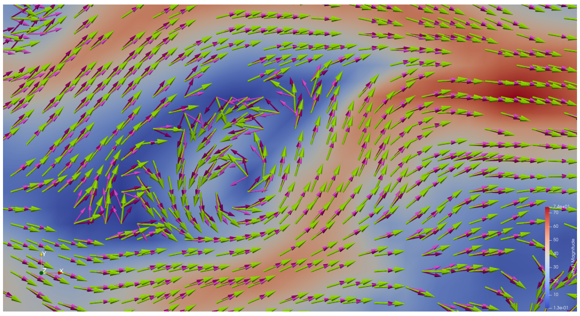
(b)

Figure 7: Glyph Visualization of Wind Flow Vector Field

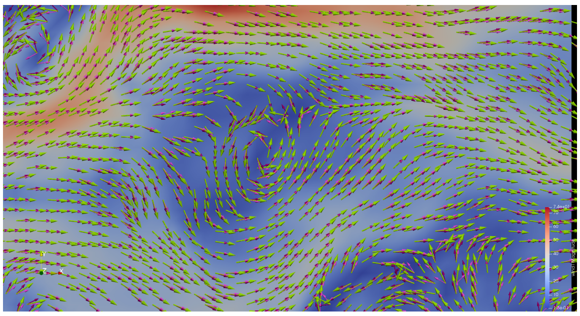
After visualizing all the glyphs of the different vector fields together, we were asked to generate glyph data for the data between the two wind fields. This was to find the angle between each of the data vectors [4] and plot is as new data to see any underlying structures. In order to find the angle between the two vectors, the identity $\cos(\theta) = \frac{a \cdot b}{||a||||b||}$ was used.



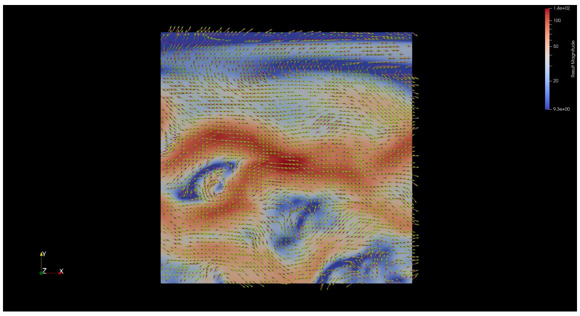
(a)



(b)



(c)

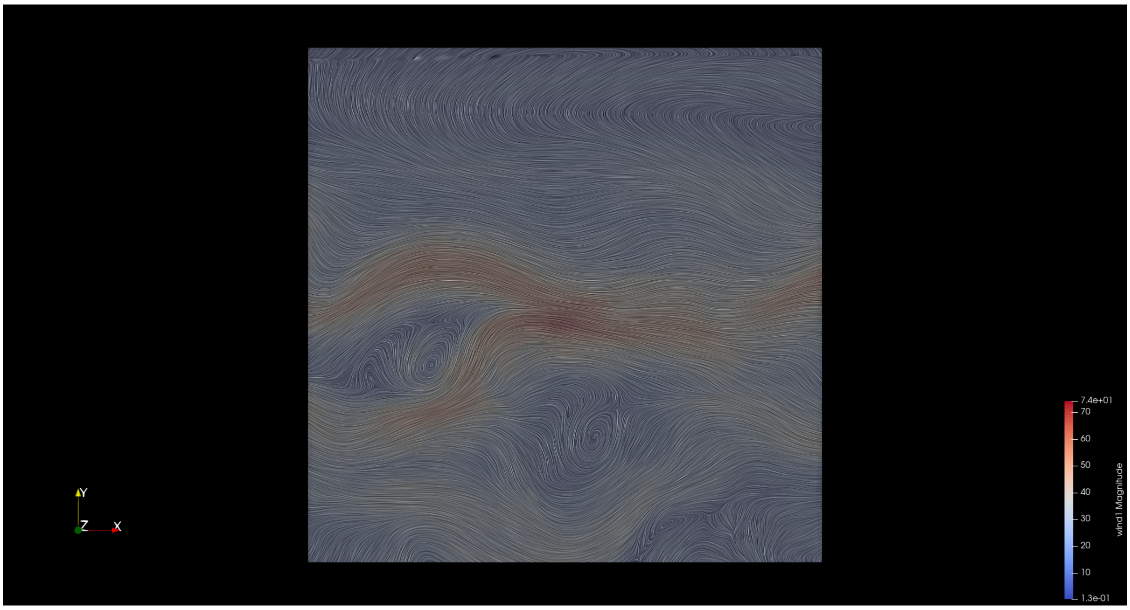


(d)

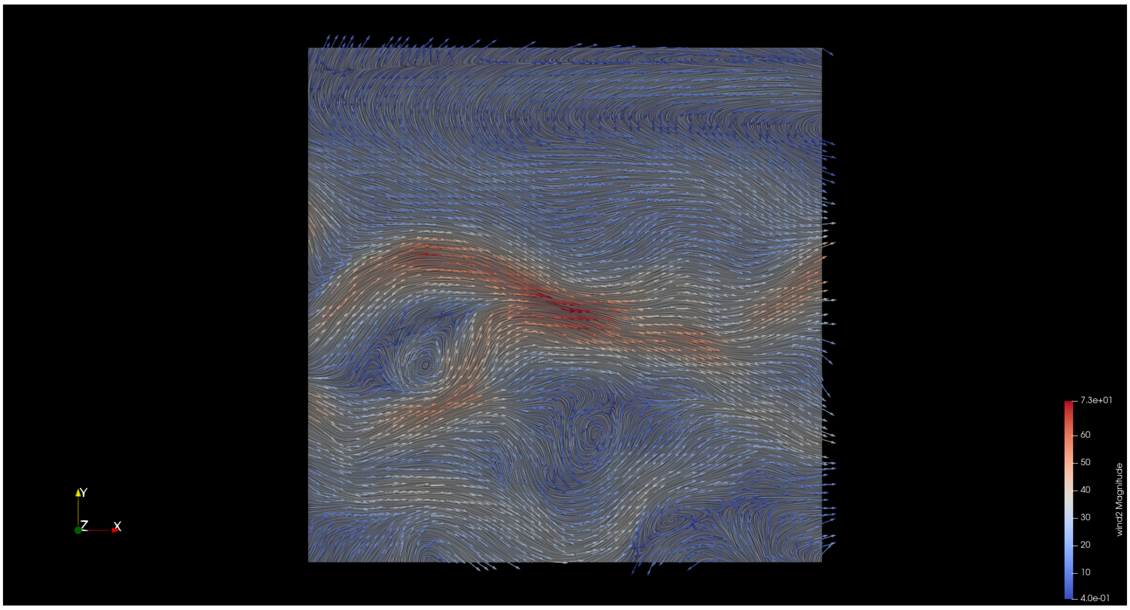
Figure 8: Glyph Visualization of Wind Flow Vector Field with Angle Between

Visualizing the flow between two angles helps see the overall flow between the two fields and what the difference between them looks like. Uncertainty here is difficult to gauge. I am unsure where the uncertainty lies as we are just finding the difference between the two fields. I do see how changing the size of the glyphs is helpful to see the overall trends in data and any underlying artifacts in the areas of higher concentration and activity.

I accidentally pushed surface with LIC for visualizing the wind data and found some great ways of seeing the flow throughout the datasets.



(a)



(b)

Figure 9: Surface LIC Visualization of Wind Flow Vector Field

Part 3: Reading Questions

For part three of this assignment, we were asked what preattentive processing is, and why is it important? Preattentive processing is the little things that let you know that a grouping of data should be considered a grouping of data, such as font, color, spacing, indentation, shape, placement, etc. When people see a scatter plot with different icons and shapes, if there are like shapes, we tend to think that those shapes should go together. Same thing with color, if the color of a grouping is the same, then we start to bring different data together.

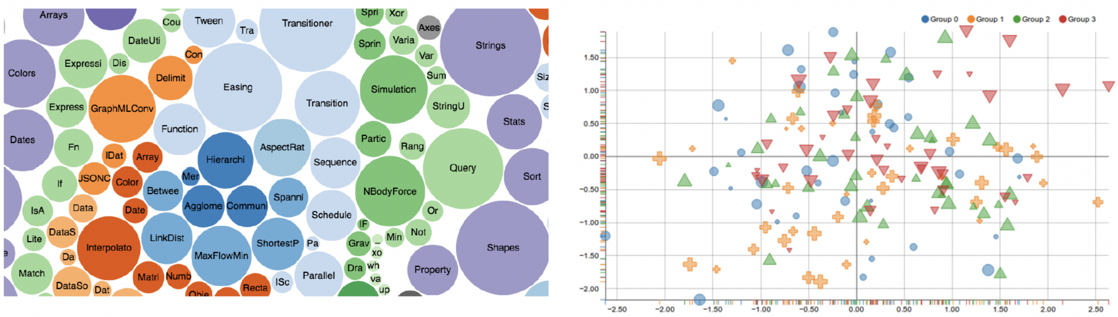
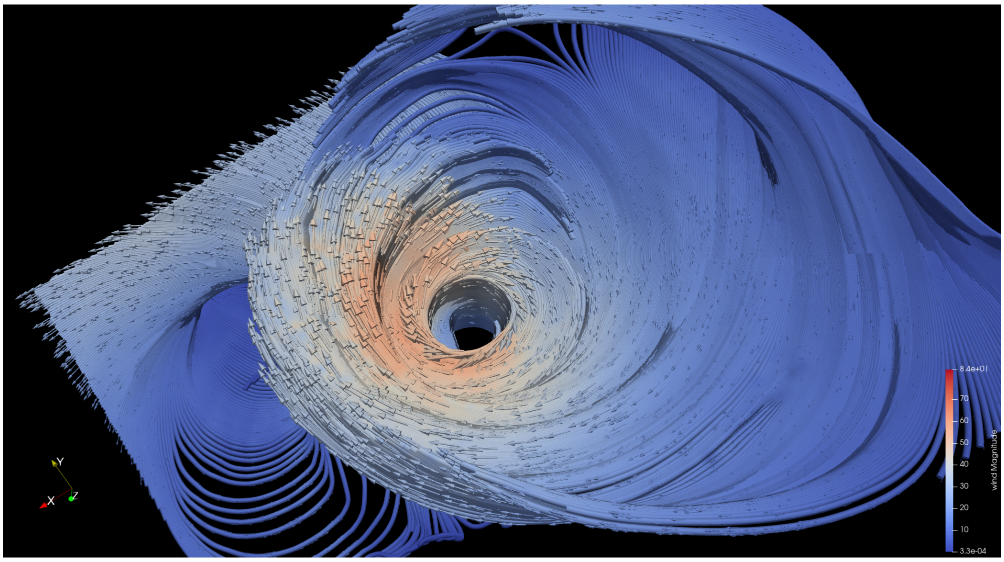


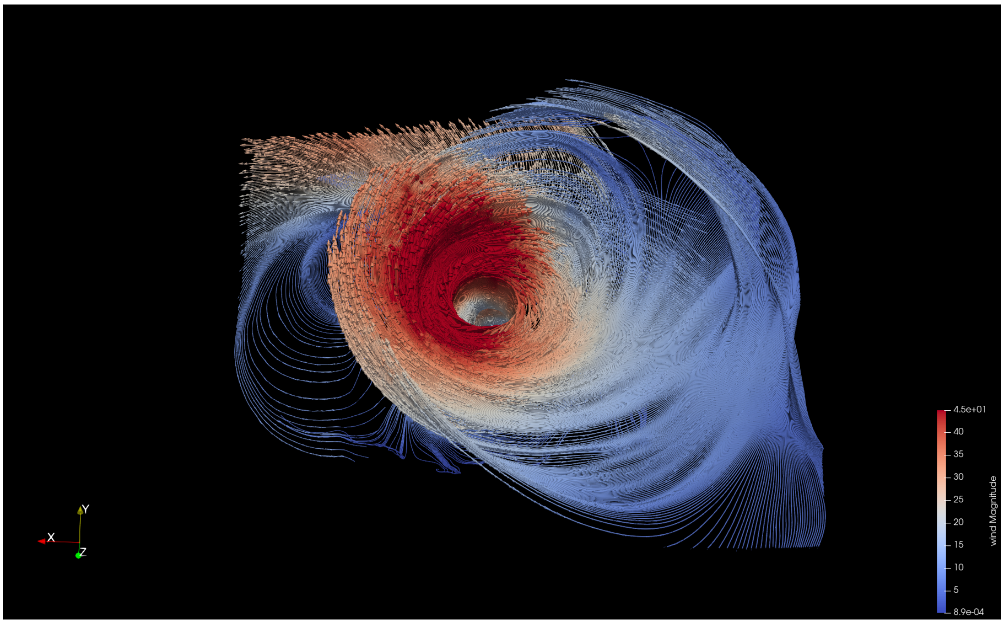
Figure 10: Different Coloring and Shapes Leads to Different Preattentive Grouping of Data

The types of preattentively processed features in these two visualizations are shape, size, color, and proximity. The first visualization looks less cluttered and gives you a sense of understanding what is in the figure, the second visualization looks like a regular scatter plot with well thought out shapes for different groupings of data, though the data is very convoluted and mixed so it is difficult to get a real sense of any underlying trends in the data.

Lastly, we were asked to find a visualization we created from a previous assignment and try to improve the perceptualization using Paraview. The figure chosen for this part of the assignment was one from assignment 04 which was of hurricane Katrina using stream tracers. The coloring of the overall diagram is a bit off, more deep reds should be used towards the eye of the storm signifying that is the area with the highest velocity of wind. The stream tracers should also be slightly thinner to get a better view of the glyphs magnitude and direction in the hurricane.



(a)



(b)

Figure 11: Revisualization of Hurricane Katrina a) Old, b) New

The new visualization shows the values more prominently. The reds signify the higher velocity of the wind in the eye of the storm and show how quickly the wind begins to slow down as it reaches the outskirts of the hurricane. The thinner streamlines give a better look at the glyphs, showing their direction and magnitude. This visualization gives a better representation of hurricane Katrina.

Part 4: Color Maps

For the last part of the assignment, we were asked to create a custom color map using *https : //www.sciviscolor.org/colormovesapp/*. To use this tool, we created a reference image in Paraview, then dragged and dropped it into the online tool, chose which colors we wanted for our custom colormap, downloaded the colormap as a .xml file, and then imported it into Paraview where we were able to use it to create a visualization. First, the reference image was created using a .xml colormap file that was provided by SciVisColor.

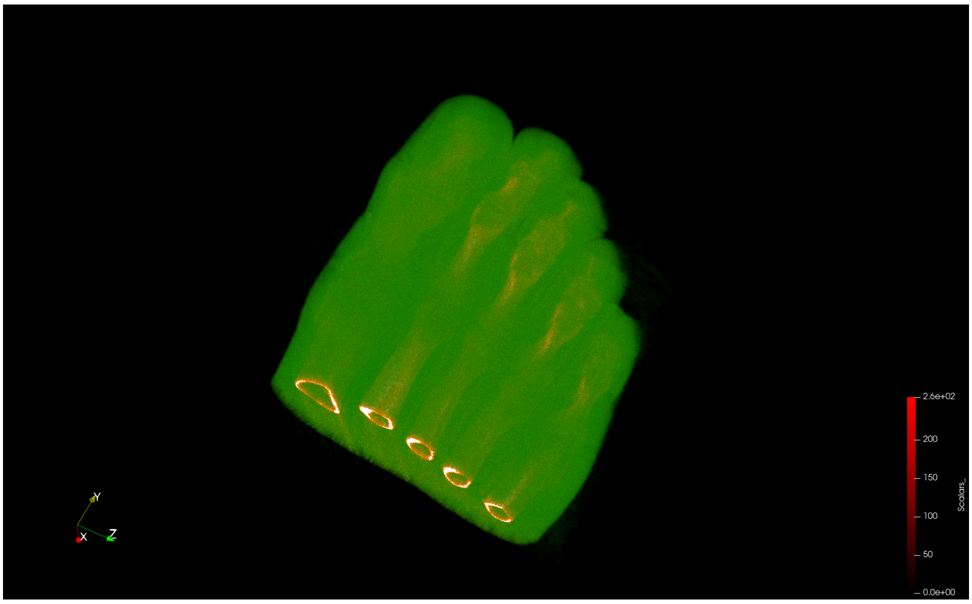


Figure 12: Reference Image for SciVisColor

Then the reference image was dragged and dropped into the websites online tool, allowing to create new custom colormaps.

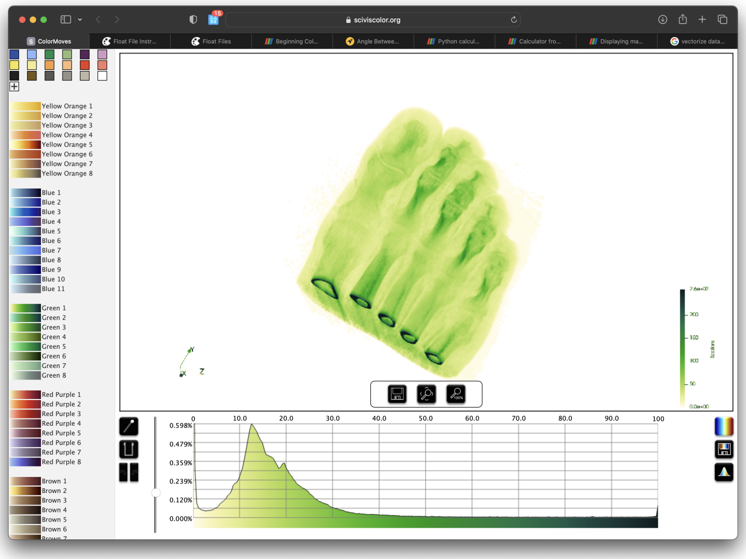


Figure 13: Generating Custom Colormap in SciVisColor

After choosing a custom colormap and exporting the colormap as an .xml file, it could be imported into Paraview and applied to the visualization.

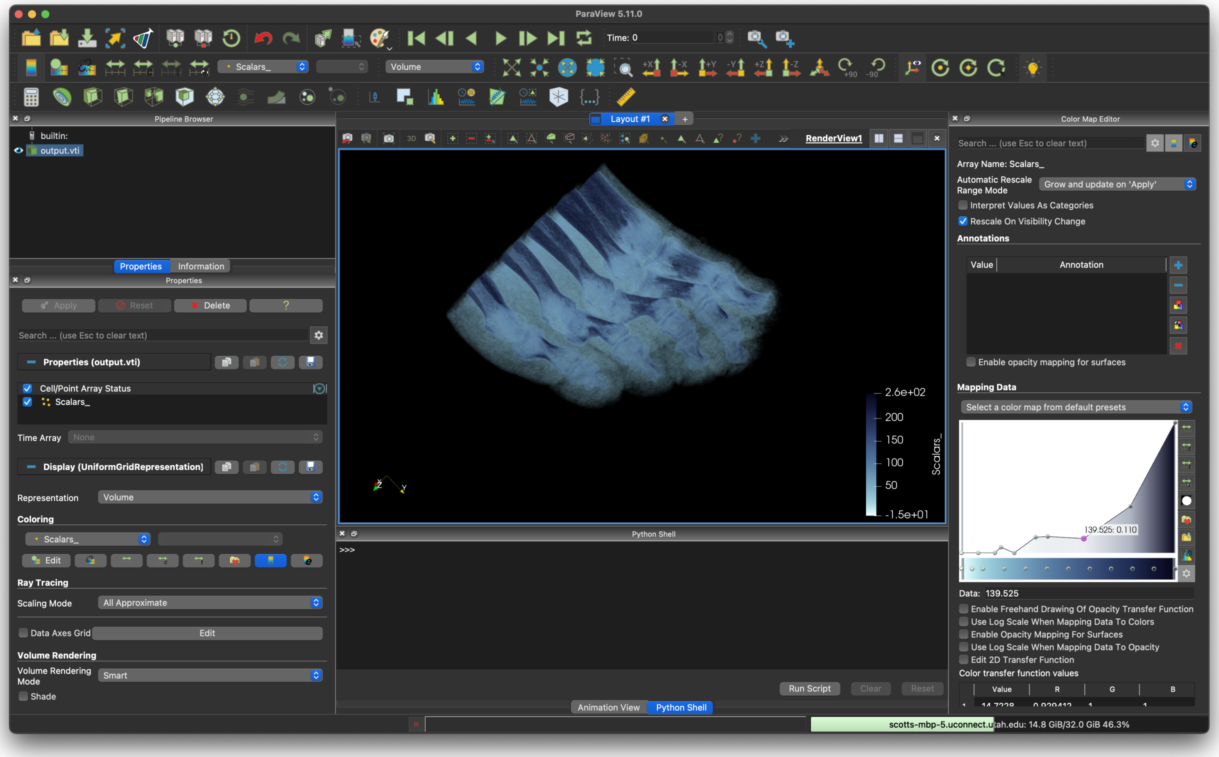
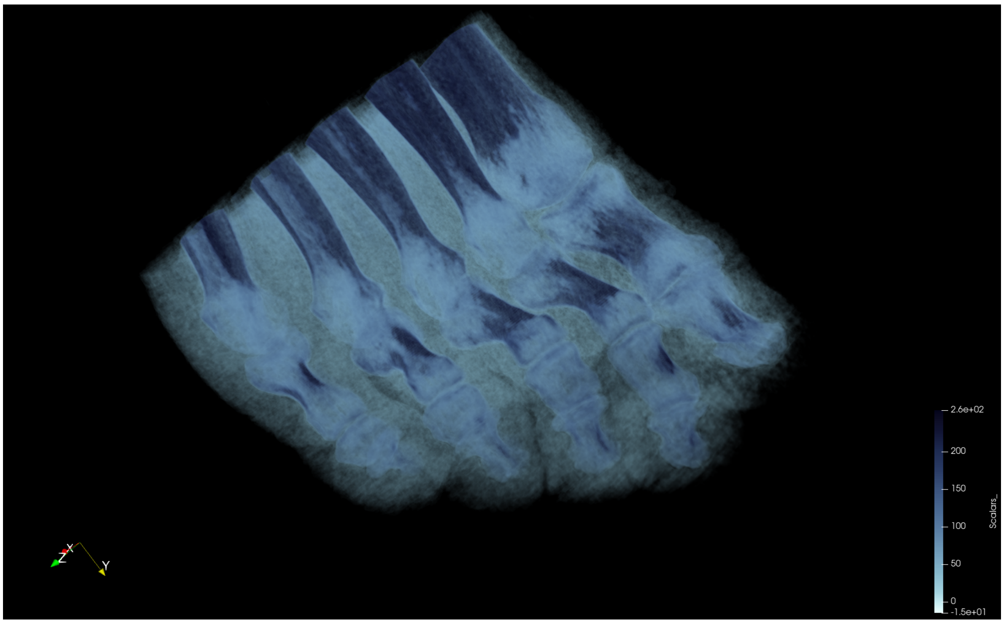
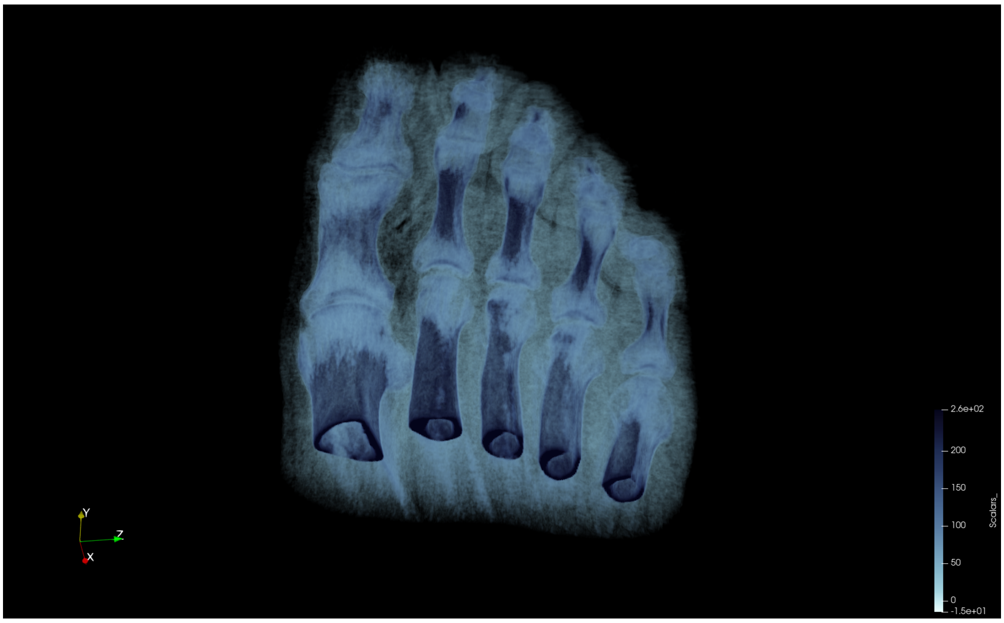


Figure 14: Visualizing Custom Colormap in Paraview

Then the custom colormap could be used in the visualization process to try and recreate the visualization provided on the homework.



(a)



(b)

Figure 15: Visualizing Custom Colormap in Paraview

Conclusion:

This assignment showed that uncertainty visualization is key to truly understanding scientific data. Without uncertainty being shown, the whole picture of the data is not able to be correctly portrayed. Also shown was spaghetti plots vs contour box plots and their abilities to show off different data. As stated above, contour box plots help show the overall big picture of the data and the general trends while spaghetti plotting can help see the minute details as well as areas of high importance in a dataset. Next, uncertainty in vector fields was shown using the difference between angles of two vectors. Preattentive processing was then covered and the importance of grouping data appropriately so that it can be easily viewed and understood by the user and what the different forms of effective preattentive processing are. Lastly, we saw how to create a custom colormap and use it to create better visualizations for our data, showing off different details that we want to focus on and giving you a more full understanding of the data.

References

[1] "Custom Colormap" : https://sciviscolor.org/colormovesapp/

[2] "Custom Colormap Instructions" : https://sciviscolor.org/colormoves/float-files/instructions/

[3] "Paraview Custom Colormap" : https://www.paraview.org/Wiki/Beginning_Color_Maps_and_Palettes: :text=Open%20

[4] "Angle Between Two Vectors" : https://www.cuemath.com/geometry/angle-between-vectors/

[5] "Using Calculator Functions" : https://www.paraview.org/Wiki/Python_calculator_and_programmable_filter