Data Visualization - Assignment 2

Vignesh Bondugula
IMT2019092
vignesh.bondugula@iiitb.ac.in

Abstract

This report contains the 3D scalar and 2D vector field visualizations of INCOIS dataset of the Indian Ocean. It explains the use of isosurfaces and slicing for scalar fields and streamline plots for vector fields. We try to study the spatial and temporal patterns related to ocean phenomena using the plots and animations generated.

1 Introduction and Data Overview

The visualizations in this report are done on the INCOIS dataset of the Indian Ocean. The dataset consists of 4 folders, each for different variables, namely,

- Salinity
- Potential Temperature
- Meridional Current
- Zonal Current

The data is during the period November 2004 – January 2005. The datasets are at 5-day interval, hence overall there are 18 timesteps. The 3D datasets for Salinity and Potential Temperature have data for 180 longitudes and 188 latitudes for heights varying from 5m to 255m, with 10m intervals. Thus, each data file for a time-stamp has 180*188*23=778,320 scalar values.

The zonal and meriodional current values can be treated as a single 2-dimensional vector field, where, zonal current speed (east-to-west speed, along latitude) can be assumed to be u- (or x-) component, and meridional current speed (north-to-south speed, along longitude), as v- (or y-) component. The datasets for current values have data for 181 longitudes, and 189 latitudes for heights varying from 5m to 255m, with 10m intervals. Thus, each data file for a time-stamp has 181*189*23=786,807 scalar values in each file.
The packages used to visualize the plots are,

- Numpy
- Pandas
- Plotly, Matplotlib

Various visualization techniques were used such as Isosurface plots, Slicing in various planes, Streamline plots, Streamline animation along depth profile as well as with time.

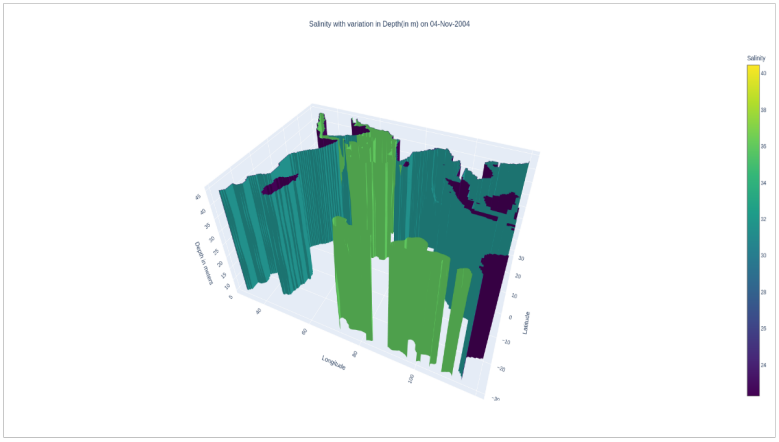
2 Visualizations

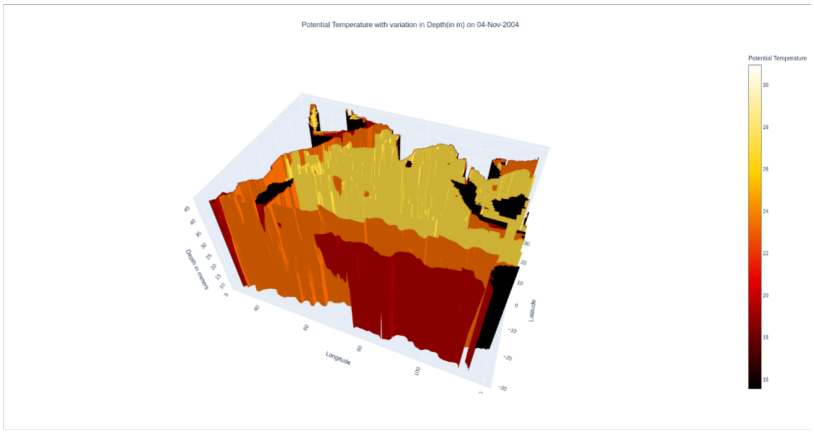
The data is extracted from the files using the numpy and pandas packages. The class DataGenerator reads the data into dataframes and stores it in a pickle file, which can be read easily whenever necessary. The Scalar field data i.e Potential Temperature and Salinity along with their latitudes, longitudes and depth for each timestamp are extracted and put in a dataframe. All the dataframes for each timestep are stored in the pickle. Similarly, the Meridonal and Zonal currents are extracted and stores in dataframe for each timestep. All the vector fields' dataframes are stored in another pickle.

2.1 Scalar Field Visualization

2.1.1 Isosurface

The isosurface module from graph objects in plotly package was used to generate the plots. The 5 different isosurface values for each field are found out by studying the data of each field. The minimum and maximum values are found for each field and then tried to study the values that are close by and repeated more often. Depth has been chosen only from 5-45 to observe the data closer to the surface. I have chosen Sequential colormaps for both the fields. 'Hot' colormap is chosen for the temperature as it represents temperature difference in a better way. The details regarding 5 diffferent isosurface values are explained in further sections. We can observe the isosurface farther from equator has a lower temperature compared to the ones closer to the equator. We plotted 5 isosurfaces in one plot using plotly as shown below,

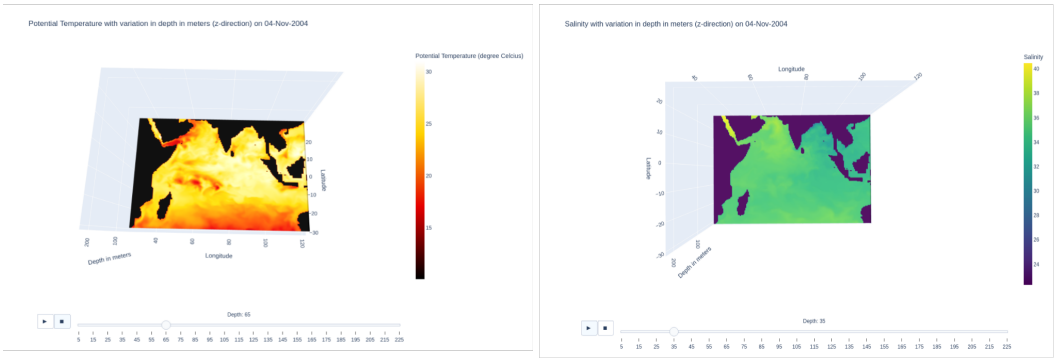




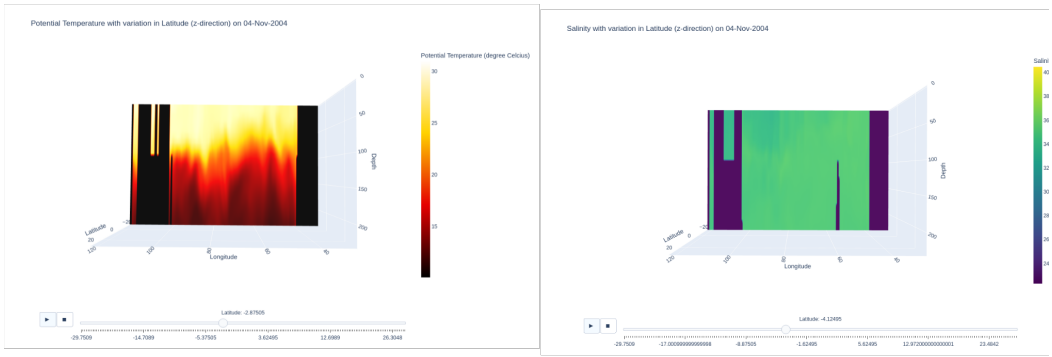
2.1.2 Slicing

6 different slicing has been done for each field. Namely,

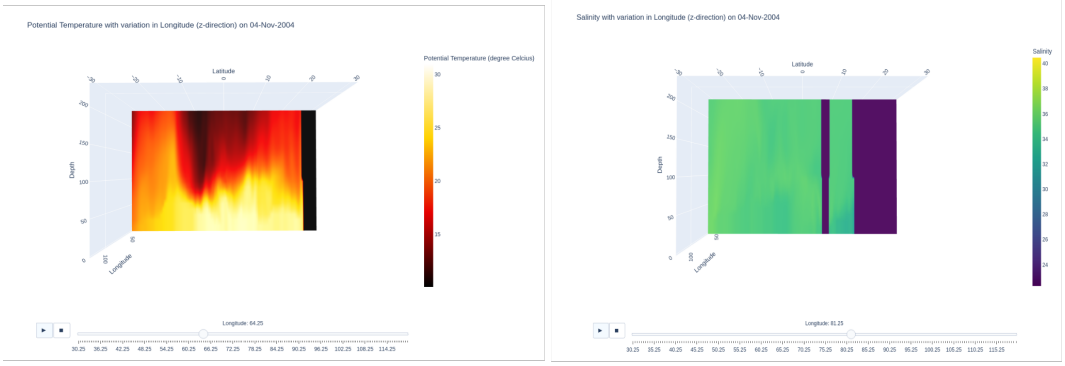
- XY Slicing - Slicing is done with respect to the depth. We can observe how the temperature and salinity changes with depth. The temperature decrease with increase in the depth can be clearly observed by this visualization.

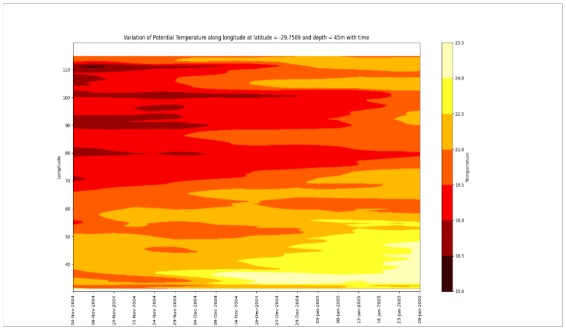


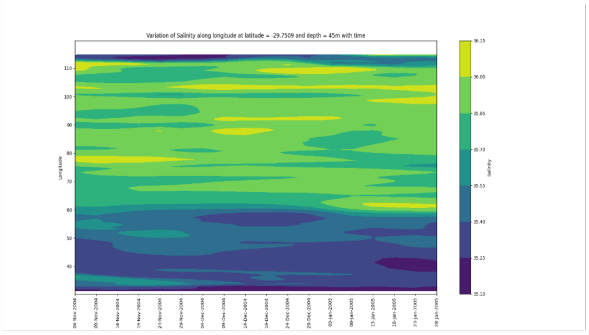
- XZ slicing - Slicing is done with respect to the latitude. The depth profiling along the longitudes can help us to observe changes in temperature/salinity with depth along a particular longitude.



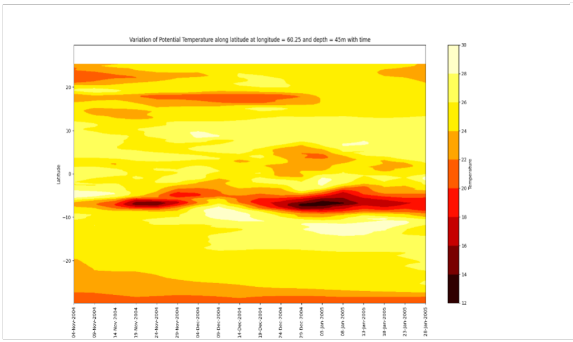
- YZ slicing - Slicing is done with respect to the longitude. The depth profiling along the latitudes can help us to observe changes in temperature/salinity with depth along a particular latitude.

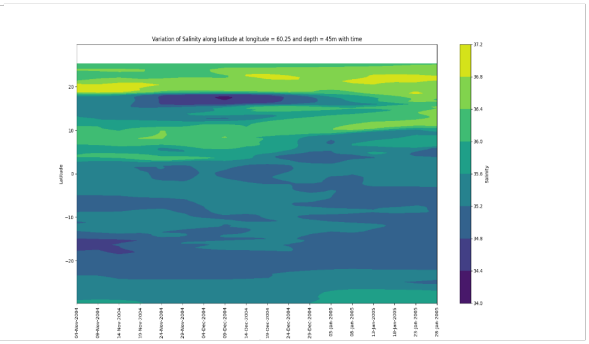




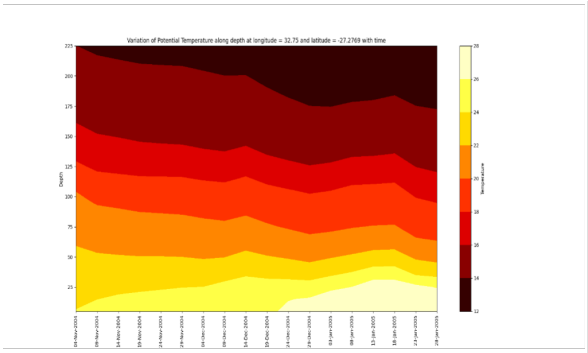


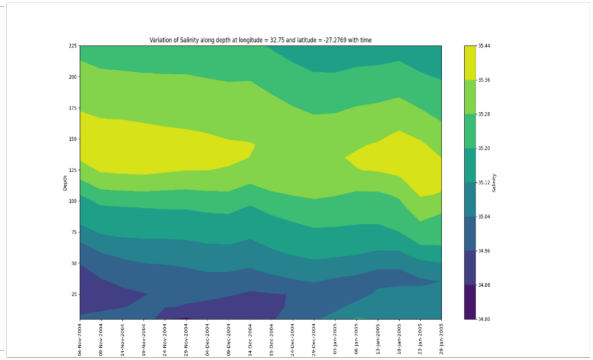
- XT slicing - Slicing is done with respect to the latitude and depth. We observe time varying aspects of the fields along a particular longitude.
- YT slicing - Slicing is done with respect to the longitude and depth. We observe time varying aspects of the fields along a particular latitude. varying aspects of the fields along a particular longitude.





- ZT slicing - Slicing is done with respect to the latitude and longitude. We observe time varying aspects of the fields along a particular depth.



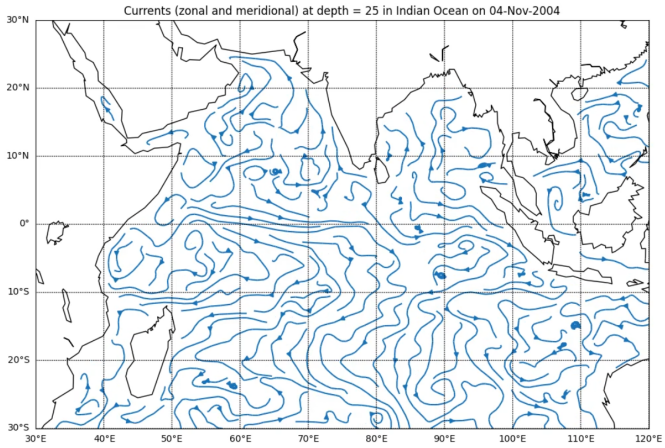


These are plotted using Surface from graph objects in plotly library. The Spatio-Temporal planes are plotted using contourf function from matplotlib. XY, XZ and YZ slicing are also animated with help of plotly.

2.2 Vector Field Visualization

2.2.1 Streamlines

Streamplot function from matplotlib is used to draw the streamlines for the given vector field. Since, the given data is unstructured i.e not uniform, griddata function from scipy library is used to interpolate the data and form a structured data. Using this the streamline plots are formed. The streamline plot is given below,



Two animations of change in streamlines in the vector field are created,

- Along the depth profile - To observe the change in streamline as the depth increases.
- Same depth value progressing in time - To observe the streamline flow with time at a particular depth.

3 Tasks

Did you interpolate intermediate slices for Task1? Why or why not?

No, I haven't interpolated the data for intermediate slices in Task1. The data given is scientific and interpolation might lead to inaccuracies. The results achieved after interpolation might not be proper and might be misleading.

How did you arrive at the 5 isosurface values?

I have calculated the minimum and maximum values for each field. I have also calculated the modes for each field. Accordingly, I have chosen the 5 isosurface values for each field. The minimum and maximum values in case of potential temperature were 15.5 and 30.9. The isosurface values chosen for potential temperature field are 15, 20, 23, 25 and 27. The minimum and maximum values in case of salinity were 22.3 and 40.4. The isosurface values chosen for salinity field are 23, 27, 33, 36, 38.

How did you visualize the 5 different isosurfaces? If the visualizations for 5 values are done or have to be done together in a single view, how will you combine them together?

The isosurfaces for 5 different values were initially visualized individually. Later on these isosurfaces are combined into one plot with help of colormap. The isosurface values are mapped to different colors of a chosen colormap in the combined view. This way we can visualize multiple isosurfaces in a single view.

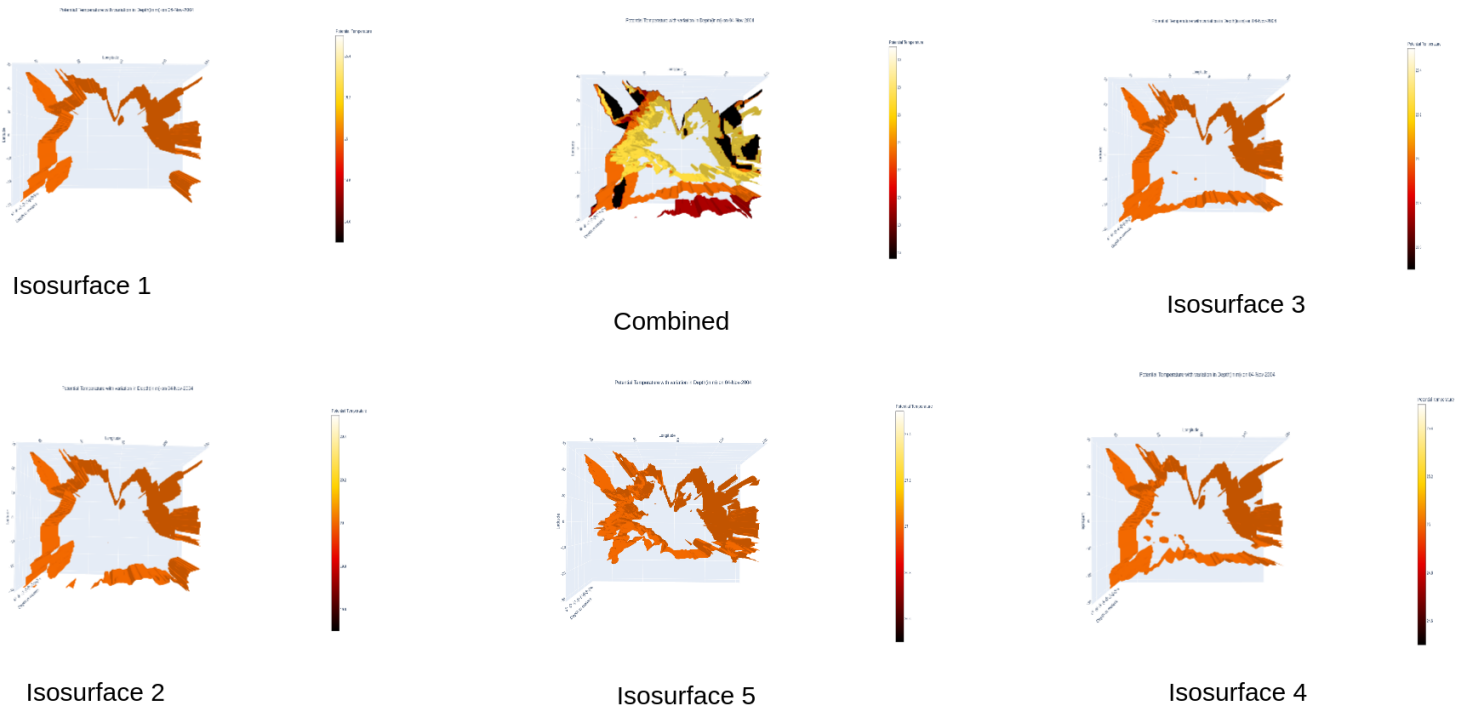


Figure 1: Isosurfaces for Potential Temperature

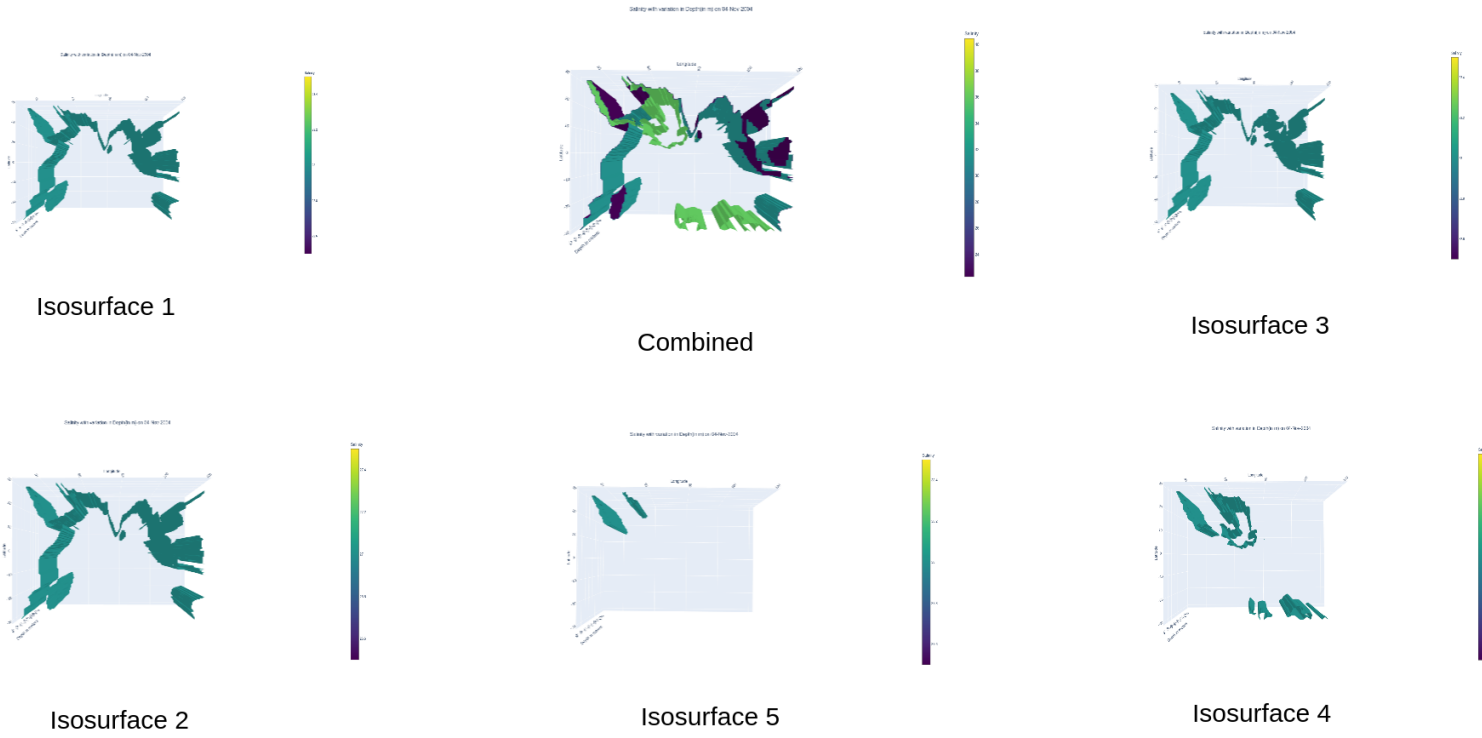


Figure 2: Isosurfaces for Salinity

How did you determine the seed points for the streamline computation?

The seed points for the streamline are calculated by the function used in the matplotlib. I also tried to experiment by choosing seed points near the locations of tsunami to observe the streamline flow.

How would the streamlined computation differ from the current implementation, if you were to negate the vector field, and use the endpoints of the streamlines as the seed points?

When we negate the vector field the streamline computed will be same but it occurs in the opposite direction of the streamline in the current implementation.

4 Conclusion

The visualizations gave us lots of insights. Since the data was really big it would not be possible to understand the information and gain knowledge without using different types of visualisations. Many more insights could have been made with much more field knowledge related to oceanic currents.