Thermocline Detection and Water Density Analysis in Great Pond, Maine

Abstract:
This project explores the concept of thermoclines, a phenomenon in which temperature gradients in a body of water cause variations in density with depth. The goal is to analyze temperature data from Great Pond and calculate the thermocline depth for specific days in July 2019. Using computational methods, I created Python functions to process CSV data, compute water densities, and identify the largest changes in density, which define the thermocline. Core computer science concepts such as modular design, file processing, and function implementation were essential in solving this problem. My key findings revealed significant temperature and density changes in July, with thermocline depths varying based on air temperature conditions.

Methods:
To solve the problem of identifying thermocline depths, I used modular design principles. This involved breaking down the task into smaller, manageable functions, such as one to calculate water density from temperature and another to compute the derivative of density with respect to depth. A key computational strategy was iterating through the temperature data, calculating densities at each depth, and then using the `max` function to find the largest density change. This approach allowed me to automate the identification of the thermocline depth for any given day. Additionally, I had to handle noisy data by using smoothing techniques to ensure meaningful results.

Results:
I ran several experiments based on different days in July 2019, comparing the thermocline depths on the warmest and coolest days. The results are presented below, showing the temperature, density changes, and calculated thermocline depths for three columns.

Analysis 1 from T8
Min value for column 4: 20.58 on day 1
Max value for column 4: 27.17 on day 31
Thermocline depth for day 1 in July: 8.00 meters
Thermocline depth for day 31 in July: 8.00 meters

Column 4 is likely Air temperature. This could impact thermocline depth as warmer air heats surface waters, possibly causing a deeper thermocline. However, no significant change was observed between the thermocline depths on these two days, suggesting air temperature alone might not explain the variations in thermocline depth.

Analysis 2 from T9
Min value for column 24: 4.89 on day 29

Max value for column 24: 7.90 on day 6

Computing thermocline for min day:
Thermocline depth for day 29: 8.00 meters

Computing thermocline for max day:
Thermocline depth for day 6: 4.00 meters

Column 24 is likely Solar radiation. This might directly influence thermocline depth due to increased surface heating. In this case, higher solar radiation on day 6 was correlated with a shallower thermocline, supporting this hypothesis.

Analysis 3 from T9
Min value for column 17: 17.58 on day 12
Max value for column 17: 21.33 on day 31

Computing thermocline for min day:
Thermocline depth for day 12: 6.00 meters

Computing thermocline for max day:
Thermocline depth for day 31: 8.00 meters

Column 17 is likely wind speed. I chose wind speed because it can mix water layers and potentially alter thermocline depth. The results showed that higher wind speed correlated with a deeper thermocline, possibly due to increased water mixing.

Based on our results, wind speed and solar radiation seemed to have stronger correlations with thermocline depth than air temperature.

Reflection:
This project helped me connect the concept of modular programming with real-world environmental data analysis. I also learned how to apply derivative calculations in a practical context to determine changes in water properties. These concepts could be useful in analyzing other natural phenomena, such as climate data or pollution levels in bodies of water, where small changes over time have significant impacts.

Extensions:
For the extension, I worked on the first question. I combined the `analyze_and_thermocline.py` script to calculate thermocline depths for any selected day from the `GoldieJulyNoon2019.csv` file. By allowing command-line input to specify the day, the script now computes thermocline depth based on the maximum change in water density with respect to depth. I also incorporated data smoothing to improve the accuracy of density derivative calculations.

The outcome was a more flexible script that dynamically calculates and visualizes thermocline depths, showing temperature gradients and density changes across selected days. Users can now easily analyze different days and generate visual outputs of temperature versus depth.

To replicate this, run the script with a specified day as a command-line argument: python3 analyze_and_thermocline_extension.py <day>

Acknowledgements:
I would like to thank Professor Isaac Lage for providing guidance on the T7 section of the project, and Catherine Ama for discussing CSV data processing strategies with me. Online resources such as the Python documentation for file handling, and W3 schools were also instrumental in completing the project.