Examining and Presenting Cycles in Temperature Logs from the Vulcan Diffractometer

Regan Moreno, Chad Steed,

and Erik Schmidt

[1] University of Tennessee, Knoxville, TN, 37996, USA
[2 & 3] Oak Ridge National Laboratory, Oak Ridge, TN, 37830
reganmoriah@gmail.com

Abstract. As scientific facilities produce more and more data, it is the job of scientists to parse through and present the data in clear, well-formatted, and accurate mediums in order to apply this information to future projects. This was the primary goal when processing the Spallation Neutron Source's challenge data for the VULCAN diffractometer. The diffractometer generates high-intensity neutron pulses towards a material and the changes in temperature over time are recorded. This information can reveal the molecular structure of the material as well as their behavior under certain conditions and duress. As the material was rapidly heated and cooled, we examine the change in temperature, eliminate stagnant data, and divide information into cycles. Doing so provides insight into the composition and behavior of the material, as well as the diffractometer's process.

Keywords: Time Series, Data Visualization, Neutron Science, Data Wrangling

1 Introduction

When confronted with the challenge data, our team decided it was a perfect opportunity to experiment with data visualization using the virtual reality headset Magic Leap. Remotely accessible, yet interactive visualization was the primary goal when processing the Spallation Neutron Source's challenge data for the VULCAN diffractometer. The diffractometer generates high-intensity neutron pulses towards a material and the changes in temperature over time are recorded. This information can reveal the molecular structure of the material as well as its behavior under certain conditions and duress. The temperatures and time were recorded as the material was rapidly heated and cooled. The goal of this project was to wrangle this data into appropriate formats and visualize it in a clear, accessible, and informative way, thereby providing further information and concepts to future studies.

2 Tools

2.1 Software

Pandas provided an opensource library which we took full advantage of. Pandas provided the data frame structure we use to load in and wrangle the provided information.
Matplotlib/NumPy [4] allowed for simple and internal examination of the data.
Altair is a declarative statistical visualization library for Python based with **Vega:** a high-level grammar of interactive graphics.
Seaborn is a Python based visualization library based on matplotlib and provided a high-level interface for the statistical graphics.

2.2 Hardware

As the team was heavily involved with the Visual Informatics for Science and Technology Advances (VISTA) Lab at Oak Ridge National Laboratory, visualization in a 3D space was a focus and learning opportunity of this challenge. We used the **Magic Leap** virtual reality headset and its connection to Unity and Helio, its built-in browser, to visualize within our homes.

3 Data

The data contained values of the tested materials temperature over time as the diffractometer conducted its tests. It was provided in a HDF5 file [1]. A Python script broke this information down into two .csv files (t1 and t2.) All examples listed throughout the paper have been taken from the second set (t2) on suggestion of the challenge authors. Each file contained about sixty-four thousand values of time and temperature as the diffractometer measured the cycles of heating and cooling in the material. All information was then loaded into a Pandas [2] data frame to wrangle with greater efficiency.

4 Technical Approach

Pre- and post-processing

Approaching time series data is an arduous process, there is a lot of wrangling to be done. One of the first issues we came across were stagnant values at the beginning and the end of the files. These values represented a. The material's resting temperature and b. Its gradual return to resting temperature post heating. The first of these issues could be eliminated so as not to throw off calculations at a later point. The second had to be carefully examined to not disregard the cooling process of the material inside of the cycles, yet not provide false positives during the final cooling. We eliminated these first values by parsing through the values and determining a cutoff temperature of 29 degrees Celsius. Filtering through the values above that cutoff provided the data we could begin to section off into cycles, remaining aware of the small spices in temperature during cooling.

Another problem we had to be aware of when examining data were small spikes and valleys in the data we needed to disregard when determining the pattern. This was accomplished by identifying cycles with a rolling average window. This is establishing a mean while traversing the data while being conscientious of the weight of certain values. A rolling average with a window of 20 values allowed our team to identify cycles with precision and append the cycle start and end values to the data frame. A rolling average is represented by:

$$SMA=nA_1+A_2+\ldots+A_n$$

Pandas allowed this to be easily accomplished as it provided built in functions to take the average between values, calculating the moving range with a window of 20 values to smooth small jumps in data, and then to calculate the differences between those ranges.

```
#Calculate the differences between values
newData['dtemp'] = newData['temperature'].diff()

#Calculate the moving range
newData['matemp'] = newData['temperature'].rolling(window=20).mean()

#Calculate the change  in the moving range values
newData['dmatemp'] = newData['matemp'].diff()
```

Code Listing 1 - Brief Calculations

Then, by setting a Boolean variable, we were able to observe whether we consider the values to be in a cycle by whether they have stopped cooling. We monitor cycle start and stop values by iterating through the information to look for when the rolling average window has begun the next heating process, if it is true, we update the cycle count, mark the starting value of the next rotation, and continue.

```python
#Create column to track cycles
cooling = False
start = 0
end = 0
newData['cycles'] = np.nan
cycle_count = 1
#Append to the cycles column if value is starting or ending a cycle

for i in range(len(newData)):
    if not np.isnan(newData.iloc[i]['dmatemp']):
        if not cooling:
            if newData.iloc[i]['dmatemp'] < 0:
                cooling = True
        else:
            if i + 1 == len(newData):
                end = i
                newData['cycles'][start:end+1] = cycle_count
            else:
                if (newData.iloc[i]['dmatemp'] > 0) and (newData.iloc[i + 1]['dmatemp'] > 0):
                    if (i + 2) < (len(newData) - 1) and (newData.iloc[i + 2]['dmatemp'] > 0):
                        end = i - 1
                        # cycles.append([start, end])
                        newData['cycles'][start:end] = cycle_count
                        cycle_count += 1
                        start = end
                        end = start
                        cooling = False

newCycle = newData.groupby('cycles').head(1)
```

Code Listing 2 - Cycle Detection Code

All code was contained within the public GitHub repo of the author [6].

Problem 1

Identify heat cycles pertaining to equivalent temperatures during the heating and cooling phase.

Graphing the data provided a clear view of decreasing plateaus of the high temperatures over time. The base graph [Figure 1] demonstrates the values of the time series as a blue line, while light orange vertical lines denote the beginning/stopping points of the marked cycles. Doing so allowed the team to then isolate specific cycles to then group together by similarity of temperature height. The Magic Leap then allowed us to overlay individual cycles for comparison. With a focus on this project being on data visualization in a 3D space, this allows the representation of chosen separate cycles to be compared visually.

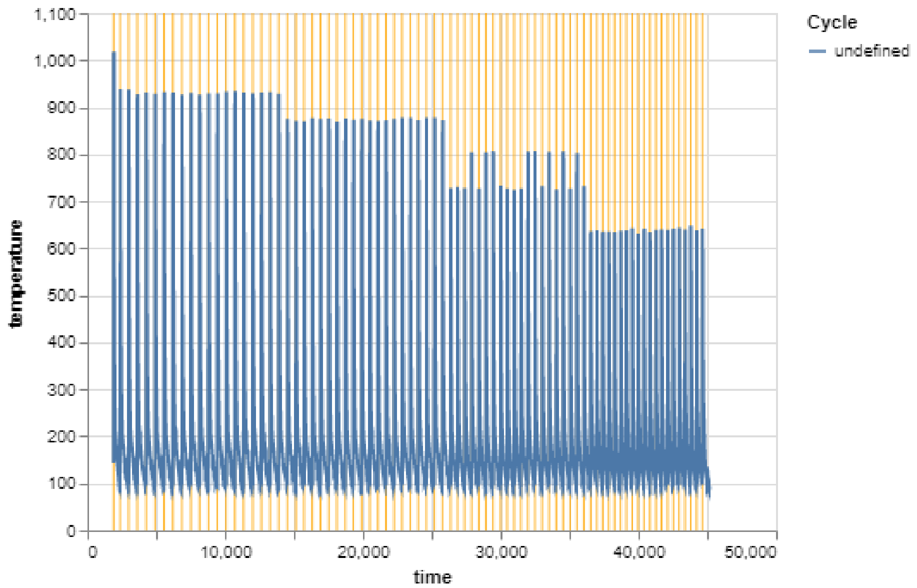

Figure 1 - Cycle Detection Marked

A suggestion inside the challenge was to compare the cycles by equivalent temperatures during heating and cooling cycles as that may provide insight on the material's response to the rapid temperature changes over time. Choosing a cap of 800 degrees Celsius will show that specific temperature being reached 304 times within the second dataset. As shown there were plateaus above 800 throughout the first half of the experiment but from a certain point that height was only reached through intermittent spikes and then never reached again as the cycles top height maxed out at around 650 degrees Celsius before the final cooling. The below graph [Figure 2] was a color histogram using the library Seaborn [5] to pinpoint recurring temperatures up to five decimal places, the range adjusted as needed.

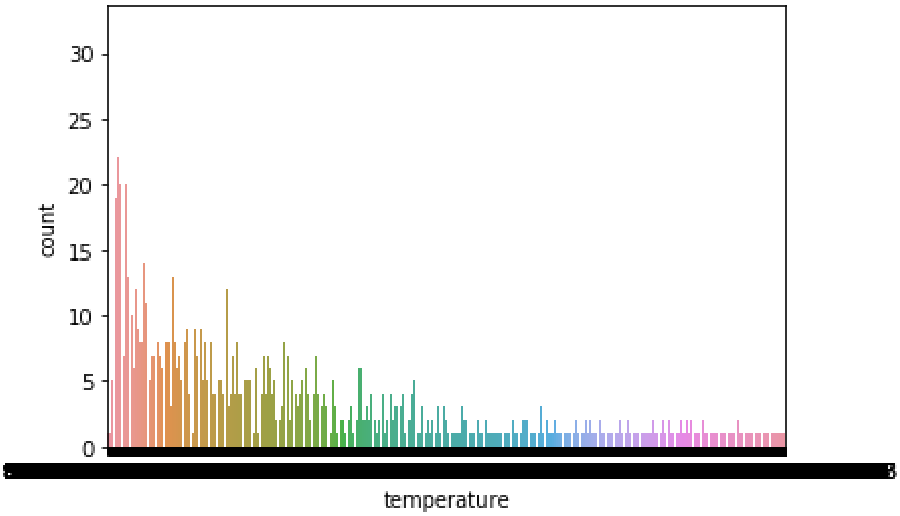

Figure 2 - Frequency of Temperature

Problem 2

How many events are there in each group and how similar are the identified events?

There was a total of 80 cycles within the second data set (t2) as identified by our program. Cycles span between 400 and 700 seconds in length with an average of 541.1457244551898 seconds in a cycle. Isolating cycles for comparison yields a varying representation of sharp inclines sliding down to a slow return to resting temperature. They vary significantly over time as the peak temperature reached lessens consistently. The below graph is the difference between the first and the second cycle.

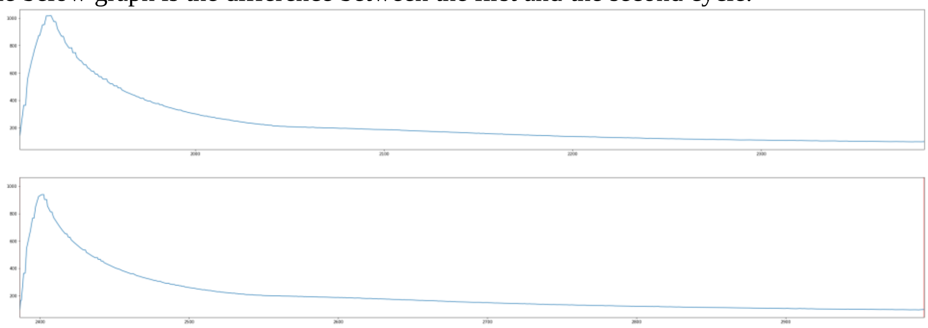

Figure 3 - First and Second Cycles

The graph below represents the difference in presentation between the first and final cycle.

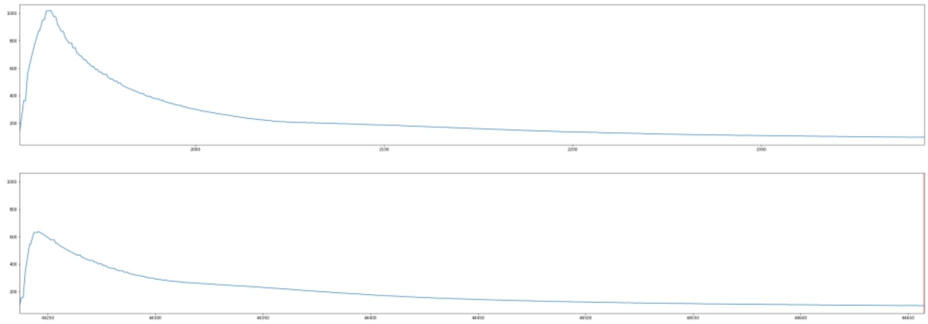

Figure 4 - First and Last Cycle

As we observe, the peak of the cycle lowers significantly after numerous tests. This may be due to the material decomposition due to frequent exposure to the beam-line. By comparing the cycles and their differences, we can observe a steady decline in the heat cap of the material as the plateau of absorption decreases.

5 Results

Using Helio, the Magic Leap built in browser, we pulled up the Altair [3] provided visualization through GitHub. This allows users to access interactive graphs within their homes [Figure 5]. The graph was interactive to the points of manipulation along the scale and zooming capability. As such, users can isolate a specific cycle for examination, zoom in or out, highlight, or travel along the axis. The Magic Leap headset is also capable of photo and video capture as well as streaming on the live streaming platform Twitch. Live streaming capabilities does increase accessibility, especially during the time of COVID enforced remote work.

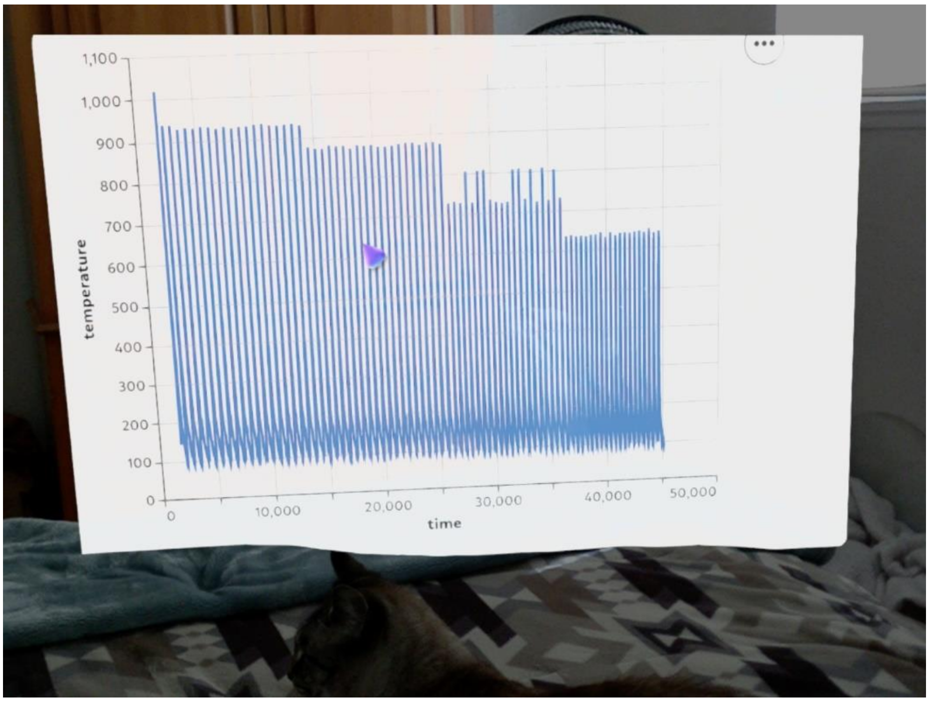

Figure 5 - Observation of the Graph in Magic Leap

Assisted by Dr. Finnegan (cat) 1

6 Improvements

Improvements would be focused on data visualization in a 3D space. We ran into some implementation issues with the Magic Leap when it came to maintaining compatibility with game engines like Unity and Unreal. Using Unity to create an interactive 3D visualization that could be openly manipulated was a goal that was unfortunately cut short by the incompatibility. Branching out to possibly use the Microsoft HoloLens 2, as a virtual reality substitute is in the works. The clear goal would be an interactive heat map that could be streamed to audiences of scientists. The presence of the COVID-19 pandemic was an inspiration for ensuring data visualization remained accessible while remote without disregarding the importance of interaction. Using Vega and the built-in browser was an acceptable substitution but branching out into a clearer execution would be an improvement to strive for in the future.

7 Conclusions

Our team took the data, wrangled it by eliminating unnecessary values, calculated, and stored important information within the data frame, and graphed information using the Altair library. Accessed the interactive graphs through the Magic Leap virtual reality headset allowed 2D holographic interaction with the data and shared the interactions through a live stream. This can allow future projects to be visualized remotely yet effectively using a single virtual reality headset and a streaming service.

7 References

1. HDF5 Homepage, https://www.hdfgroup.org/solutions/hdf5/, last accessed 2020/7/27
2. Pandas Homepage, https://pandas.pydata.org/, last accessed 2020/7/27
3. Altair Homepage, https://altair-viz.github.io/, last accessed 2020/7/27
4. Matplotlib Homepage, https://matplotlib.org/, last accessed 2020/7/27
5. Seaborn Homepage, https://seaborn.pydata.org/, last accessed 2020/7/27
6. Personal Github Repo, https://github.com/raygunzapzap/intern, last accessed 2020/7/27