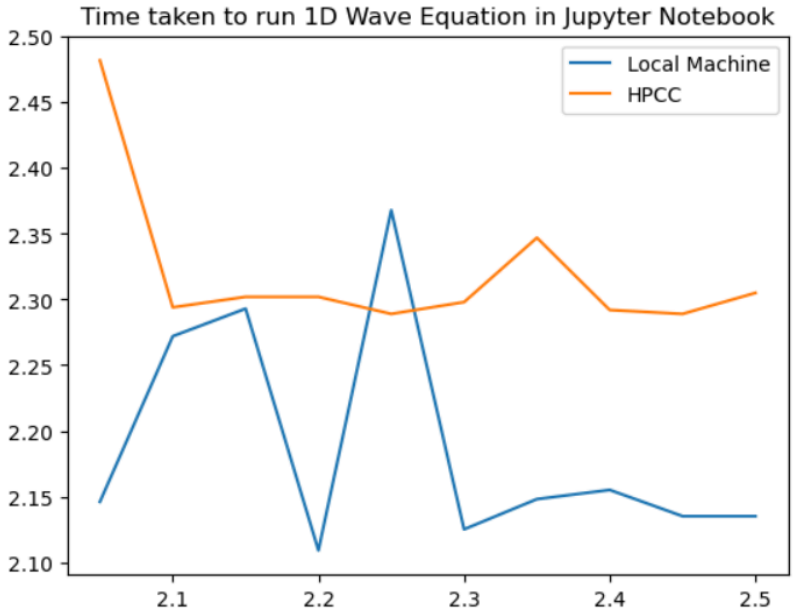
CMSE 401 Homework 1 Write-up
Annie Wilcox

Instructions:

This homework assignment was completed in Jupyter Notebook using a Python kernel. To reproduce the results of the assignment on your local machine, you will need to have Python downloaded (https://www.python.org/downloads/). After downloading Python you will be able to open an Anaconda Prompt on your computer. Type "conda install jupyter" into the Anaconda Prompt in order to download Jupyter Notebook. Once downloaded, open the homework file. All of the necessary libraries are imported for the user in the file already. If you run into error running the imports cell, you may need to download a library. In order to download a library, type "pip install <package name>" into an empty cell and run it. Once all the libraries are downloaded, you can select "Run" and then "Run all cells" at the top of the notebook to see the results of the assignment. Within the folder, there is also a file named "wave.gif". This is the full visualization of the wave equation, as the visualization in the notebook does not show the animation, but just a screenshot of it.

Write-up:

For the timing study, I chose to run my Jupyter Notebook file on both my local computer and the HPCC Jupyter server. My local computer has an AMD CPU with a speed of approximately 3.3 GHz. It has 15.7 GB of RAM, and uses Windows 11 as its operating system. I only used 1 core when running this code, but my computer has 6-8 cores. The HPCC computer also has an AMD CPU, but it's much larger and more efficient than my laptops. It has 128 cores, though I only used one, and the HPCC server I was using had 503 GB of RAM available. Linux is the operating system that the HPCC uses.



Using the time library that is a part of Python, I ran my function 10 times on both the HPCC Jupyter server and my local machine. On average across both machines, my function took 2.2543 seconds to run. I was surprised to see that the HPCC (average of 2.32 seconds) almost always took longer than my computer to run (average of 2.1886 seconds). If I had to guess why this is, I would say it is because the HPCC is optimized to only alot the user the resources they absolutely need. Maybe my small, non-parallelized project doesn't require the fastest possible resources to run. I also find it interesting that the HPCC runs more consistently than my computer. It makes sense, because the HPCC is not allotting the resources its using on my code to anything else, while my computer has to use the same resources to run my code and all other background processes.

To speed up my implementation of the wave equation, I could have used NumPy's vectorized operations (np.linspace, as used in the pseudocode) to do operations on whole arrays at once, thus eliminating some usage of loops which are not very efficient. I could have also removed the helper function and replaced it with a single, defined computation to once again avoid the slowness that loops are known to cause. Finally, there are some better libraries for animations that run more efficiently than ArtistAnimation.