CMSE401 HW01 - WaveEquation

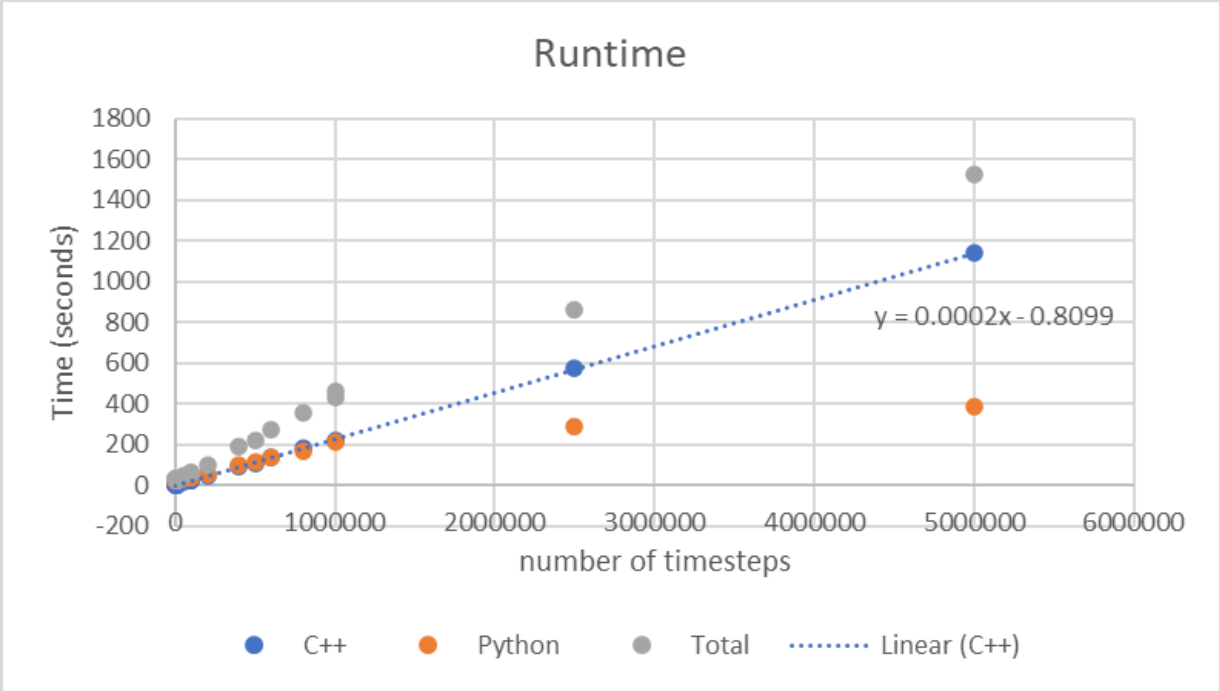
Kyla Hotton

In this homework assignment I simulated a 1D wave equation using the instructions provided to me. I did this using C++ version 20 with the compiler g++ version 9.4.0. With the simulation I then made a visualization with python version 3.9.12. I ran my time evaluations of the codes using my laptop. My laptop has a 11th Gen Intel(R) Core(TM) i7-1165G7 processor with a CPU speed of 2.8GHz, a RAM of 16.0 GB (15.6 GB usable), a wsl linux operating system with software version Ubuntu 20.04, and 1 core used.

To reproduce my results someone would need g++ version 9.4.0 with libraries iostream, vector, fstream, algorithm, and math.h. From there they will need to run the command “g++ simulation.cpp” in their terminal while the folder hw01. The folder hw01 is inside of the folder CMSE401. That compiles the C++ code. To run the compiled code, they then need to run the command “./a.out” in their terminal. This will create a simulation of a 1D wave equation and put the data in a csv file in the folder Data. The first row of the csv is the x positions and the rest of the rows are the y positions of each additional time step.

To reproduce the visualization, someone would need to have python version 3.9.12 in the same system as g++ version 9.4.0. They will need the libraries matplotlib and numpy for python. They will also need the application ffmpeg for python. To get a visualization they need to run the command “bash make_visualization.sh” in their terminal while in the folder hw01. The visualization will appear in the folder hw01 with the title “wave!.mp4”

To do a timing study of the code I used the command “time bash make_visualization.sh” which returns the time it took the simulation to run, the time it took to create the visualization, and the total time of both. I changed the value for n_t in simulation.cpp for each trial. The results of the trials are shown in the graph below.



On average the simulation ran following the function Time(n)=0.0002n-0.8099 where n is the value of n_t and time is in seconds. This means that the simulation code is O(n) time complexity. This is expected based on the code. The visualization code started to trend slightly towards being constant. Once the csv file from the simulation loads in python the rest is run in constant time. Having very large csv files from the simulation means that the visualization time can become very large. For 1,000,000 the simulation took 217 seconds.

One way to make the program run faster is to reduce the size of the csv file in the simulation code. This would allow the visualization portion of the code to run much faster. To run the simulation portion of the code faster I could try using On Demand. I don't know if this would work but I could try if I had more time. I am assuming there is some kind of parallel computing trick I could use that I don't know yet. I will learn more soon though :)