Nematullah Waseem’s & Dakota Turk’s Proposals

We have 2 projects that we are proposing to work on:

1. Particle Hydrodynamics Simulation Using CNDO/2

Would be largely based on work produced by Sebastian Lague’s work in Unity using C#: https://www.youtube.com/watch?v=rSKMYc1CQHE&t=2089s. We both have watched the linked video a couple of times and have gained an appreciation for the simulation and how Sebastian implemented the simulation. In the video he also shares all of his resources and we would leverage them as well to do the work. We would likely utilize docker for the implementation as well as C++. During this we would have to do some parallelization work as well as conversions from mathematical equations to implemented C++ code.

We would aim to get to around maybe 25% of the same level of efficiency or runtime as Sebastian does in the video and utilize an API system and some basic html endpoints for the host machine to have a small server be accessible from the docker container so that the user could render the simulation in their web browser. This would be more involved, but also would assist in real world applications of point-based simulation work, which is very similar to molecular dynamics simulations.

We would likely spend the first week pulling in all resources in the video or other relevant C++ related work and begin the implementation with Matt focusing on converting the show available C# code to C++ and Dakota would focus on getting the docker-html visualization working. Then we would come together in the last two weeks to combine our code and to write up our report.

Once again, for this project we would likely use a much smaller number of particles to simulate, in order to account for the time constraint of the deadline for submission. Further work we do to it we would be able to make it more efficient to handle a much larger number of particles.

Objectives:

1. Convert Sebastian’s code into C++ [joint focus]
2. Basic docker to host machine html visualization [Dakota’s focus]
3. Basic freezing expansion property of water using dipole calculations [Matt’s focus]
4. Switch out his density force calculations with one learned in class [using force field calculations] [joint focus]

Notes

- Basic fluid SIM in 3 dimensions with boundary conditions and gravity using classical physics and CNDO calculations.
- Out of the sim we want to be able to visualize density and temperature of the water
- Visualize the expansion/crystallization of water as well as simple movements when fluid

2. CNDO Parallelization (Backup)

We would aim to agree on an object format and collapse our functions into them then utilize OpenMP and/or MPI to parallelize our code. The above would allow us to extend our current CNDO/2 for models with a larger atomic count. In addition, it would be interesting to account for electron shielding that occurs for larger atoms.

To begin the first week, we would combine the existing CNDO code to align with one class that combines the various aspects of retrieving the atom, building the matrices. This would allow for easier parallelization of the code. The following week, the focus will be to ensure the various parts of our code block are parallelized and compatible with OpenMP. The last two weeks, the focus will be to optimize the parallelization and do a comprehensive performance analysis for a final report.