

Report For Assignment 3

by Group E

by Joshua Voigt, Luca Tänzler, Georgios Chatzikyriakou

June 6, 2024

Link to the pull request
clang version: 10.0.0

1 Introduction

In this assignment, we had three tasks to complete: The XML parser, which will be used to parse XML input files for ease of use on the side of the user, the Linked Cell Algorithm, which optimizes the simulation, and last but not least, the boundary conditions, which handle the particles that go out of bounds. All of these tasks help make the simulation faster to run and more user-friendly.

2 Task 1: XML Input

The first thing we were tasked to do is to implement support for XML files, which will be used for input. We decided to use XSD for our project and implemented an XML parser, that parses the arguments for use in the program. This eliminates the need to manually set up all the arguments through the command line each time the simulation is run. We also implemented a driver, which needs to be compiled before it is used. The purpose of the driver is to act as an intermediary between the XML input file and the XML parser, translating the XML input arguments to a more readable map of key-value pairs for easier parsing. [[[Sadly this feature has not been fully implemented yet, so for now the user needs to manually input the arguments through the command line each time (REMOVE IF DONE)]]]

3 Task 2: Linked-Cell Algorithm

In this task, we had to implement a version of the container that implements and uses the linked-cell algorithm. This algorithm divides the container into cubes of side length r_c , where r_c is the cut-off point for the Lennard-Jones potential. This

allows for every particle to seek other particles that reside in its neighboring cells and only do calculations with them. This algorithm greatly decreases the complexity of the program from $O(n^2)$ to $O(n)$. The new container also includes functionality for retrieving particles that are on the border cells, as well as the halo cells. We also created a particle container interface for seamless changing between the basic and the linked-cell container. The interface contains 3 functions: One for retrieving all particles, one for only retrieving neighboring particles and one for updating the simulation, in case one of the particles moves from one cell to another in a single frame. [[[INCLUDE DATA ABOUT RUNTIME AND RESULTS]]]

4 Task 3: Boundary Conditions

Last but not least, we had to implement boundary conditions, which denote how the container handles particles, that are on the border of the simulation. We had to choose between outflow, which means that the particles can freely exit the simulation and get deleted, and reflecting, which means that the particles that are about to exit, instead get repelled by the wall through a ghost particle with the same attributes, but residing in the corresponding halo cell. In the end, we decided to go with simple outflow, since it is much simpler than reflecting and also works better for our program structure and code. We sadly did not have time to implement tests for outflow or the linked-cell container.

5 Task 4: Simulation of a Falling Drop

In preparation for the experiment in this task, we implemented a disc generator, that inherits from the Initializer class and generates a disc with the given radius, distance between the molecules, mass of each molecule, velocity and center coordinates. We also added these attributes to the XML input format, so that the user can generate the discs himself.

As always, the diagram is included in the zip folder. Only the new classes and functions have been added to make it more readable, as well as the old classes, if they're used in some way by the new ones