Scientific report for the duration of the project

First stage (15.09.2020 – 06.12.2020)

During the first stage of the project, the following activities have been carried out:

1. Expansion of computing capabilities of the group

We originally planned to acquire graphics cards to expand the GPU computing capabilities of our compute nodes. However, the appropriate graphics cards are in short supply and we could not buy standalone GPUs that would fit our need. We therefore decided to get two high-end desktops which contain nVidia RTX 2060 GPUs, and add them to our computing nodes in a mesh configuration, with computing tasks distributed by a central node. We also acquired DDR4 memory modules to increase the available memory of the computing nodes so that larger systems can be simulated.

2. Setting up the software framework for efficient exploration of potential energy surfaces in model systems

The scripts needed throughout the course of this research project will mainly be written in the Python programming language. Python is a high level programming language, for which multiple different packages are available. The main packages which we will use include:

- **HOOMD Blue**, a general-purpose particle simulation toolkit optimized for execution on both GPUs and CPUs. Website: http://glotzerlab.engin.umich.edu/hoomd-blue/
- **PELE** (Python Energy Landscape Explorer), a package that contains tools for global optimization and energy landscape exploration. Github: https://github.com/pele-python/pele
- **TensorFlow**, a platform for developing and testing machine learning models. Website: https://www.tensorflow.org/

The environment was created and deployed with the Docker platform (https://www.docker.com/). Docker provides the ability to package and run an application in a loosely isolated environment called a container and thus eases the process of deploying and maintaining computational environments. Once the images of these containers are created or downloaded, they can be moved to any other machine, which has docker installed, without worrying about the applications dependencies.

Docker was installed on a local machine to develop and test any images used and on the server to carry out the calculations. Alongside docker the Nvidia run time was installed, so that we can utilize the GPUs of each system. The three main Docker containers, which we will use:

1. The molecular dynamics environment:

This mainly consists of the HOOMD Blue library mentioned earlier, and some custom scripts which will be used to automatize the workflow of guessing and perfecting the simulation parameters, also a basic database which will record all the data, written in Pandas. This container can be accessed through the built in Jupyter notebook, which is an interactive Python scripting environment.

2. The structure optimization environment:

This Docker image is built on the Python 2.7 image, with PELE and some other Python packages installed, like NumPy and SciPy for performing the required calculations and Pandas to manage data. The main role of this container is to optimize simple molecular structures, to produce input for the machine learning environment and to study the energy landscapes of molecules.

3. The machine learning environment:

This container is based on the official Tensorflow-Jupyter image, with some extra packages to support the creation of 3D images about the molecular structures. The purpose of this container is to create a neural network, which takes as input a randomly generated molecular structure and predicts a structure, which on the energy landscape is located closer to the global minima of the respective structure.

3. Testing of the software framework

MD simulations with different LJ potentials

A system of 125 particles at equal distances in a simple cubic lattice has been set up for molecular dynamics (MD) simulations (Figure 1). Three MD simulations were run using different LJ potentials: the value of the potential well depth (epsilon) was increased from 0.1 through 1.0 to 10.0 while keeping the other components of the potential constant.

The differences in the LJ potentials clearly affect the way how the particles interact with each other (Figure 2). The physical parameters of the systems are also clearly different (Figure 3).

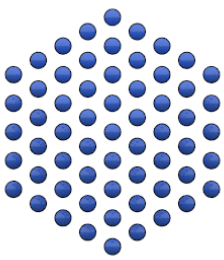

Figure 1. Starting position of the MD simulations.

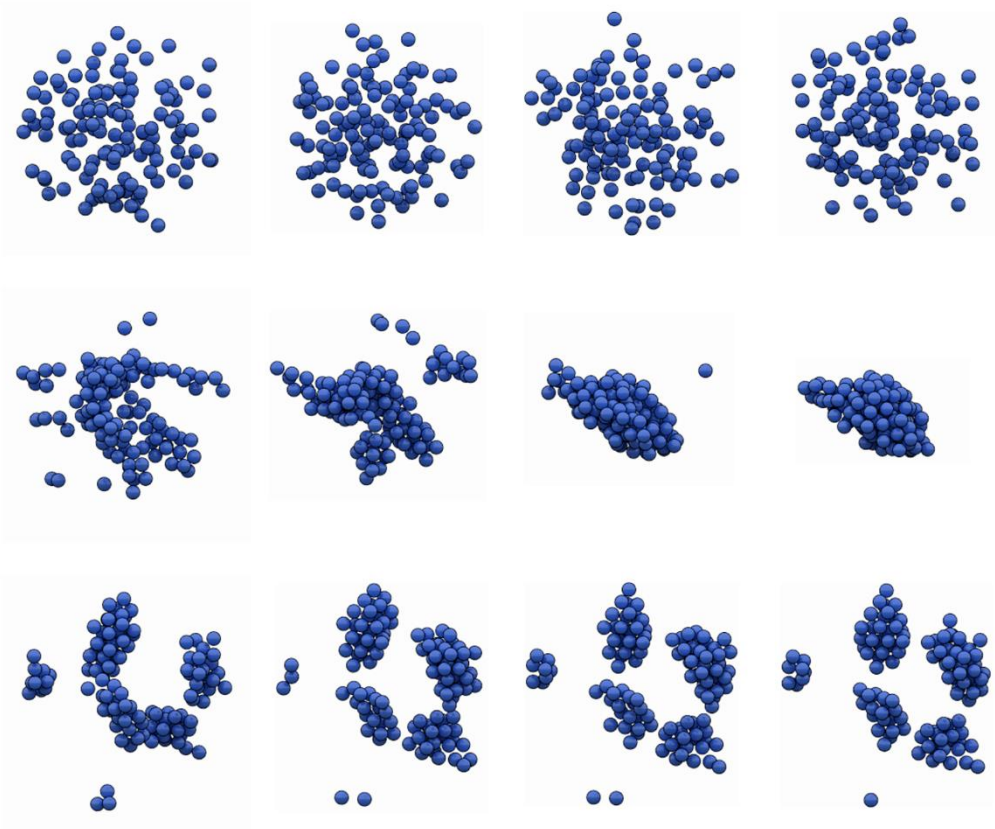

Figure 2. MD snapshots of the three different simulations taken at equal time intervals.

Top: epsilon = 0.1. *Center*: epsilon = 1.0. *Bottom*: epsilon = 10.0.

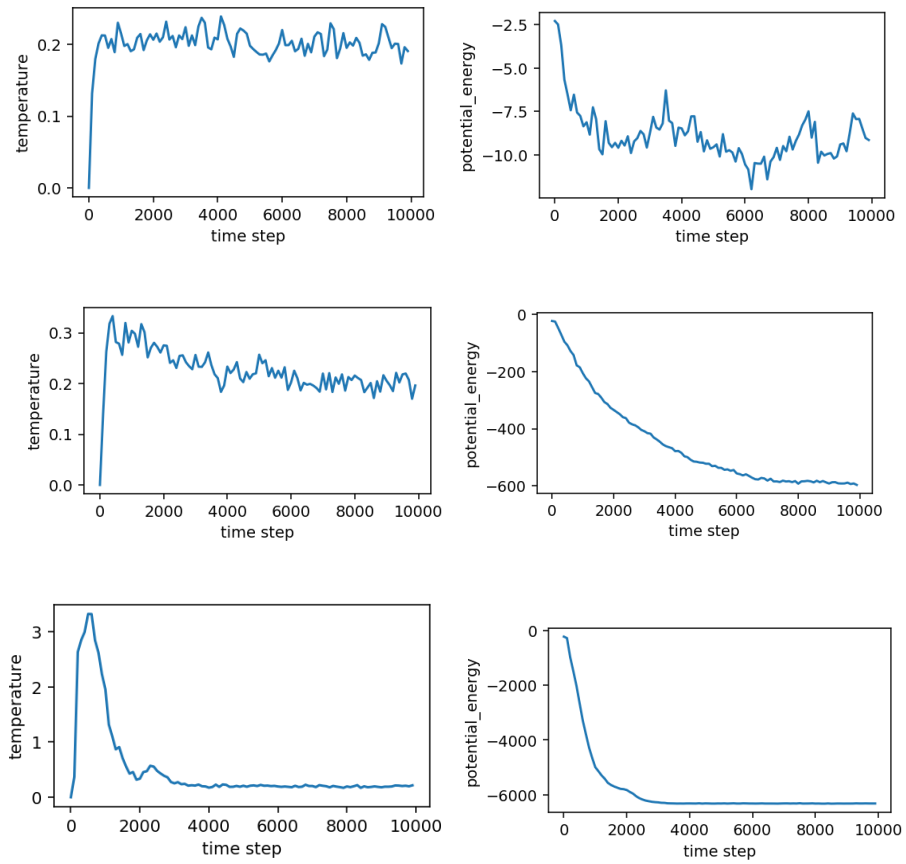

Figure 3. Evolution of the temperature and potential energy during the three MD simulations.

Top: epsilon = 0.1. *Center*: epsilon = 1.0. *Bottom*: epsilon = 10.0.

MD simulations with two types of particles

64 particles of two different type have been set up in a simple cubic lattice for MD simulations. The interaction strength between the different particle types can be controlled by changing the pair potential well depth (epsilon) for each particle type pair. In the first simulation, the interaction between different types of particles was forced (Figure 4, left). In the second simulation, the interaction between similar types of particles was preferred (Figure 4, right).

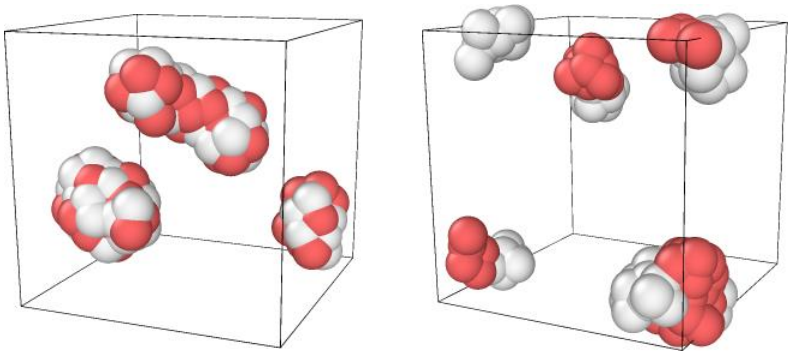

Figure 4. Final snapshots of the MD simulations using two types of particles (shown in different colours). *Left*: interaction between different particles was forced. *Right*: interaction between similar particles was favoured.

Figure 5 shows example results of a simulation on rigid body particles that has been set up with the developed Docker images. The script takes as input only a few parameters (definition of rigid body types and numbers, cell size) and then the system setup, interaction matrix and starting configuration are determined automatically. After that step, input files are generated to be run on GPUs on our computing cluster. Trajectory files can be collected from the compute nodes by a script and visualized by the user.

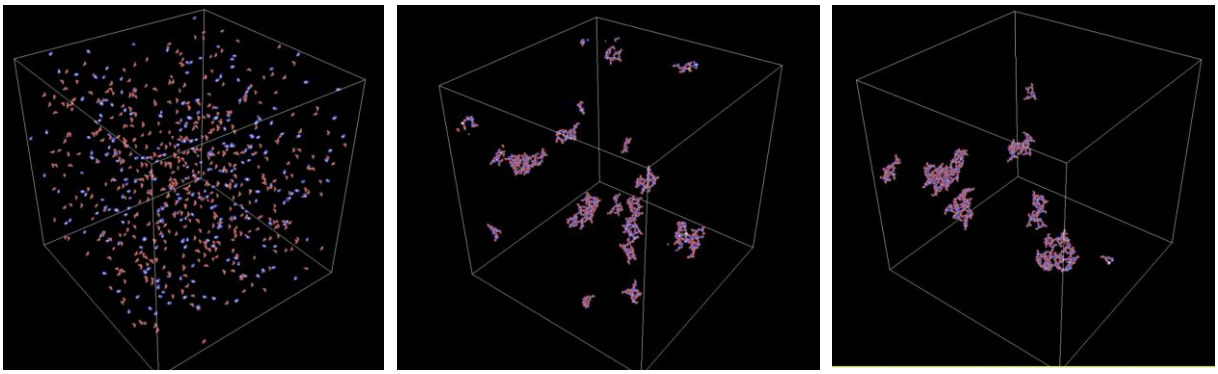

Figure 5. Snapshots along the trajectory of a two-component rigid body system using the automated script generation platform.

Sfantu Gheorghe, on 07.12.2020

Dr Szilard Fejer