University of São Paulo
Institute of Physics
SAMPA research group of Material Physics and Mechanics Department

Lauro Barreto Braz

HOW TO USE VIRTUAL REALITY TO INTERACT IN REAL TIME WITH YOUR LAMMPS SIMULATION

University of São Paulo
São Paulo, 2019

Summary

Introduction

Virtual Reality (VR) visualizations are a big advantage in physics studies, since it can help the researcher or the student to create a certain intuition about his study system. This work try to help researchers and students to have contact with unusual molecular systems and allows them to experiment different interactions creating arbitrary situations in molecular systems.

This specific method allows the simulations be governed by any force field and any atom present in standart periodic tables. It is possible because of interactive molecular dynamics (iMD) functionality, present in *LAMMPS [1]* standard library. This functionality sends positions information to the LAN server in real time, what permit the researcher use one computer to run the simulation and another one to render it at the same time. Also, iMD let we apply forces to atoms during the simulation calculations, what is the most interesting part of this work.

Through the lines below you will learn how exactly prepare your systems and files to experiment real time simulations interactions in your study systems. In chapter 1 we will learn how to prepare the initial positions file (1.2), which will be opened in the *UnityMol [2]* program in the VR accessible setup (1.1) and in chapter 2 you will learn how to prepare LAMMPS input (2.1) and *UnityMol* connection with your simulation computer (2.2).

I hope you enjoy the iMD experience.

1. Starting to work with Virtual Reality

1.1 Virtual Reality setup and UnityMol

To use this interactive simulation method you need in your setup a HTC Vive VR oculus or an Oculus Rift VR oculus and a computer that supports your oculus.

The program used is *UnityMol* a molecular scale visualization program that supports our interest functionality, iMD.

1.2 Preparing the initial positions file

The initial positions file must be a *.pdb (Protein Database Bank)* file, for *UnityMol* only read this file format. But it's important to say that the iMD info sent to the program is independent of file format, so we only need an *initial positions file* in that format. Therefore, I wrote a code that turn normal *dump* positions file in a .pdb file. Note that you need to run the simulation once according to your needs. For example, if you need an equilibration level, you run it and run only 1 time step in the production level (you only need one, but if you have more the code will work too). For convenience I wrote it in bash, because it's easy to execute in any linux system, the main LAMMPS operational system.

As a prerequisite to this code work with your LAMMPS dump output file, you need to write the following line in your input file in production stage after your dump defining line:

```
        dump_modify group-id sort id
```

After prepare your simulation, run it once with normal equilibration time and just 1 step in production stage. Then run the bash code in the link present in *Links* section in the end of this file.

2. Working with iMD

2.1 Preparing LAMMPS input file

This section will just show you what you need to append in your LAMMPS input file (*in.something* file type) to use the iMD functionality and how to use its options.

Basically you just need to append one line in your LAMMPS input file. Is recommended to add this line above the *compute* and *fix* lines. The code to be written has the following example form:

```
        fix unitymol all imd 5678 trate 10 fscale 1
```

In this example, we are saying to the program: “Save information of the *unitymol* group-id of *all* the atoms and groups of the simulated system using the method *imd* and access the port *5678* to send one time step information each *10* time steps (*trate 10*) and apply the user done force in a normal scale (*fscale 1*).” Therefore, if you prefer you can just copy and always use this exactly line excluding *trate 10*.

Knowing that interpretation language, let’s see with more detail each option.

imd number (port number): Port number is a property of the setup that is doing the simulation. It’s recommended use a number bigger than 1000, for the smaller numbers are usually used in intern function of the operational system and other programs.

trate number: This function is a refresh rate to the positions information sent to the VR setup, like *dump* function does when you choose to save positions info each *n* time steps. It is important that this rate (time step rate) must be nor too long neither too short. For if the calculations are slow, then too short *trate* result in a long time to the the forces application response and if calculations are fast, then too long *trate* result in a big jump in time and too short *trate* result in a rederization problem, because a big refresh rate requires a big processing capacity. Use this option carefully, for this is the reason of most of the problems of rendering in the VR setup. Common values to *trate* are between *1* and *25*. The best experience is using *trate 1*, but it is not always possible.

fscale number: This function sets the scale of the applied force in a atom by the user in VR setup. The applied forces are calculated in unities of *kcal/mol/angstrom*. If LAMMPS is running with different unities so *fscale* may help you to do adjustments.

It’s important to say that by default LAMMPS only starts the production stage when you set a connection with the simulation setup. Also, you only can set a connection with the simulation setup if both computers, the one responsible by simulation and the one responsible by VR setup, are connected in the same local area network (LAN).

In the *Links* section below there is a link to the LAMMPS official documentation page about iMD function.

2.2 UnityMol connection with the simulation setup

After prepare your initial positions file, start LAMMPS and see the initialization message on your terminal, you are able to connect the simulation setup with the VR setup, then this section tutorial need to be executed in the VR setup. To do that you need to install *UnityMolX 1.0.18* version or higher, present in section *Links*, and follow the steps below.

First, you double click the *UnityMolX* execute file, it will open the app. So you dress the oculus (HTC Vive or Oculus Rift) and use the controllers to access the menu inside the application. A mouse click is the same that a touchpad click in HTC Vive controller inside the app. In version *1.0.18* you can drag the menu to wherever you want clicking on the blue circle above the menu. Knowing that, you will click in “Load” button and you may see a new window opened. This new window allow you to navigate through your directories. Then, go to the directory where you stored the initial positions *.pdb* file. After select the file and click “Load”, you will see your system in its initial state.

Now you can finally connect to the simulation setup. Inside the application but without the oculus you press the keyboard key “ ‘ “ what will open the Python menu. Then you type in this menu a command in the following format:

```
connectIMD(“fileName”,”IPAddress”,portNumber)
```

Let’s understand better what is each variable here:

fileName: That’s the name you chose to the *.pdb* file without the file format information. For example: you named a file as *water.pdb*, so the *fileName* is *water*.

IPAddress: The IP address is an intrinsic information of the computer that is running the simulation. In the section *Links* you will find a website page that explains how you can see your IP address in a Linux computer through the terminal.

portNumber: That’s the same port number you set in the iMD code line in your LAMMPS input file. For example: The same in the example of the section 2.1, *5678*.

After doing those steps you are able to dress your oculus, press “Enter” in your keyboard and star the experience of Interactive Molecular Dynamics.

To apply a force in an atom, point the “laserline” that starts in your hand or controller (touching the touchpad in HTC Vive) to the atom you want to move and press the touchpad button. To continue applying the force it’s just keep your finger touching the touchpad, you don’t need to press it anymore.

Links

1. Format converter code - https://github.com/laurobbraz/iMD-VR-project
2. LAMMPS iMD documentation page - https://bit.ly/2Rr2GFA
3. UnityMolX 1.0.18 - https://bit.ly/2RX5HTm
4. IP address on Linux - https://bit.ly/2DI1Z78

References

[1] - Plimpton, S. Fast Parallel Algorithms for Short-Range Molecular Dynamics *J. Comput. Phys.* 1995, 117, 119.

[2] - Doutreligne, S., Gageat, C., Cragnolini, T., Taly, A., Pasquali, S., Derreumaux, P., Baaden, M. UnityMol: interactive and ludic visual manipulation of coarse-grained RNA and other biomolecules Virtual and Augmented Reality for Molecular Science (VARMS@IEEEVR), 2015 IEEE 1st International Workshop.