Setup instructions

In preparation for the molecular simulation workshop, you will need to have software installed on your computer. The installation might take a considerable amount of time, so please plan accordingly.

This workshop will require the use of the Python programming language and many libraries that are available in that ecosystem. One of the easiest ways to get started with Python, its libraries and molecular simulation software, is via the conda package manager. You can get access to these tools via Anaconda or Miniconda.

Anaconda is a distribution of Python, the conda package manager, and several third-party libraries which are commonly used in data science. Miniconda contains only Python and the conda package manager. You will be able to install any package you would like later using miniconda. Miniconda will take up a lot less space on your computer. We will be using conda environments and install the packages we need, so we consider miniconda to be the better option between the two. If you already have Anaconda installed, however, there is no need to install miniconda.

If you have previously installed anaconda or miniconda in your machine, feel free to skip to the section *Setting up conda environments*.

Notes for Windows users

Installation of WSL2

It is strongly recommended that Windows users install the Windows Subsystem for Linux (WSL) in advance. You should install WSL2 and have Windows 10 or 11 in your local machine. If you have an older version of Windows, please contact the workshop organizers.

The instructions for installing WSL2 can be found in the official Windows documentation. If you do not have a preference on Linux distribution, we recommend installing Ubuntu.

Once WSL2 is installed, open your “Start” menu and choose “Ubuntu”. This will open a terminal window. You may see a message about finishing the installation. After the installation is done, you will have to create a username and password. After this, you should be able to use the terminal.

WSL2 is like having a Linux distribution inside your Windows machine. You can run Linux programs and Windows programs simultaneously. You can access files from Linux from

Windows and vice versa. For more information, we recommend reading the excellent WSL2 documentation.

The instructions after this section need to be entered in the terminal.

Installing a text editor

There are many excellent editors you could use. For example, vim, emacs, nano, gedit, sublime, visual studio code, etc. Gedit is an editor that might be familiar to some, as it is similar to Notepad on Windows. If you would like to install gedit, for example, do the following

WSL2 and Linux

```
> sudo apt-get install gedit
```

You will be asked to enter your password.

MacOS

You might need to install homebrew, which is a package manager for Mac. To install it, follow the instructions at this web page or open the terminal and type the following:

```
> /bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Follow the instructions on the screen. You may be asked for the password of your machine, so have that ready.

Then you can install many packages. You can search for them here.

The installation command will be given. For example, to install gedit, you would type

```
> brew install gedit
```

Installing miniconda

The installers for miniconda can be obtained from the download section of the conda website. Below we provide the instructions for convenience.

WSL2 and Linux

Open the terminal and download the miniconda scripts from the terminal. If you use Linux or WSL

If you do not have curl already installed:

```
> sudo apt-get install curl
```

Otherwise, you can skip the latter step.

```
> curl -O https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
> bash Miniconda3-latest-Linux-x86_64.sh
```

Follow the instructions. After the installation is complete, close and open the terminal. You should see a (base) before your username in the command line. Please type

```
> conda init
```

MacOS

If you use MacOS, please go to the website and pick the appropriate installer for your computer architecture. If you have a newer Mac with an M1 processor, please do:

```
> curl -O https://repo.anaconda.com/miniconda/Miniconda3-latest-MacOSX-arm64.sh
> bash Miniconda3-latest-MacOSX-arm64.sh
```

If you have an x86 architecture (Intel):

```
> curl -O https://repo.anaconda.com/miniconda/Miniconda3-latest-MacOSX-x86_64.sh
> bash Miniconda3-latest-MacOSX-x86_64.sh
```

Follow the instructions. After the installation is complete, close and open the terminal. You should see a (base) before your username in the command line. Please type

```
> conda init
```

Setting up conda environments

Before we start, we need to add conda-forge to the channels available to our conda installation. Conda channels are the locations of the libraries and programs that we will be using.

Conda-forge is a channel that contains many packages that are frequently used in fields such as molecular simulations or data science. These packages are maintained by the developers and are relatively up to date.

Conda-forge does not come as a default package in conda, so we need to add it first

```
> conda config  --add channels conda-forge
> conda config --set channel_priority strict
```

Optional and suggested: you might want to install mamba, a C++ reimplementation of conda. This will make it fast to install the required packages. Install mamba in the base environment as:

```
> conda install mamba -n base -y
```

This may take several minutes to complete - be patient.

You are now ready to create the conda environments that we will need for the workshop. First, we will create the conda environment for the MD part (first day), followed by the MC part (second day). If you did NOT install mamba, type:

```
> conda create -n lammps-tutorial lammps mbuild foyer matplotlib jupyter pylat -y
> conda create -n cassandra-tutorial mosdef_cassandra matplotlib jupyter garnett
pycifrw -y
```

Alternatively, if you decided to install mamba in the base environment:

```
> mamba create -n lammps-tutorial lammps mbuild foyer matplotlib jupyter pylat -y
> mamba create -n cassandra-tutorial mosdef_cassandra matplotlib jupyter garnett
pycifrw -y
```

Activate the *lammps-tutorial* environment and do a quick test of the installation:

```
> conda activate lammps-tutorial
> python -c “import mbuild”
> lmp_serial
> pylat -h
```

The second command will run the *import mbuild* command in Python. If everything went well, you should get a warning message like:

```
Warning: importing 'simtk.openmm' is deprecated.  Import 'openmm' instead.)
```

You can safely ignore this particular warning message.

The third command will output the LAMMPS version that you have downloaded from conda. To terminate LAMMPS, press Ctrl-C.

The fourth command will output the documentation of the program PyLAT.py

Similarly, you can do a test for the second environment:

```
> conda activate cassandra-tutorial
> python -c “import mosdef_cassandra”
> cassandra.exe
```

You can alter between environments freely using the following command:

```
> conda activate <environment_name>
```

To return to “base” use:

```
> conda deactivate
```

Installing ovito

Ovito is a visualization tool that allows you to see how molecules move in the simulation.

Please go to the Ovito website and download the required executable for your platform. Open the executable and follow the instructions.

To test the installation of Ovito, please download the following file and unzip it by writing these commands in the terminal

```
> wget https://github.com/emarinri/UNICAMP-utils/archive/refs/heads/main.zip
> unzip main.zip
```

Open the installation of Ovito and see if you can visualize this system by clicking on the “File” menu and the “Load file” option.