

MD softwares : Espressomd

Espressomd is a open-source software published under the GNU General Public License (GPL3). It is parallelized and can be employed on desktop machines, clusters as well as on supercomputers with hundreds of CPUs, and some modules have also support for GPU acceleration. The parallel code is controlled via the scripting language Python, which gives the software its great flexibility.

To illustrate the abilities of the software, one starts from sample python scripts that should be modified for the exercices

1 Lennard-Jones model : single CPU

To start the exercices, one first needs to check that espressomd is available on your computer.

- 1) First, edit the python file **lj_M2.py** and analyze the different parts of the script file. For running a simulation on your local computer, type **ipypresso**. Once you have obtained the prompt, type run **lj_M2.py** (the script filename).

When the simulation is finished, analyze the png file of the radial distribution function. Does the figure correspond to the expected results?

One now modifies the original script file in order to obtain different results.

- 2) Increase the number of bins by a factor 2 or 4 for $g(r)$ in order to obtain a better resolution. In order to keep the original figure, change in the python script the figure filename
- 3) Increase the number of particles by a factor 2 or 4. How is $g(r)$ modified ?
Increase the the minimum value r_{min} to 0.7 in order to have a more accurate distribution function.
- 4) Change the density from 0.7 to 0.4. In order to collect simulation results
We now modify the original script file in order to obtain different results
- 5) Increase the strength of the viscous force of Langevin dynamics.

2 Lennard-Jones model : parallel simulation

In order to use the MPI abilities of the software, one needs to modify the script file as follows : edit the python file **lj_M2.py** and change the seed variable in a list with a number of elements equal to the number of cores required for the simulation

- 6) To start a simple simulation on your local computer with two cores, adapt your python file and type **mpirun -n 2 pypresso lj_M2.py**
- 7) By using the time function, compare the efficiency of your code running on a single core with a parallel simulation with 2 cores.
- 8) Write a slurm script file for running a simulation on the cluster by using 4 cores.

A How to install espressomd

- 9) Download the stable package espressomd
- 10) Update required packages

```
sudo apt install build-essential cmake cython3 python3-numpy libboost-all-dev openmpi-  
common fftw3-dev libhdf5-dev libhdf5-openmpi-dev doxygen python3-opengl python3-sphinx  
python3-pip libgsl-dev
```
- 11) Unpack the archive espressomd (tar xvf espresso-4.1.0.tar.gz)
- 12) In the directory espresso type :

```
mkdir build  
cd build
```
- 13)

```
cmake ..  
make
```
- 14) Once the compilation is over , **ipypresso** is located in the directory build. Enjoy !