

# Exercise Session 4

## MDMC Spring 2023

Lorenzo Agosta, Virginia Carnevali, Simon Dürr, Sophia Johnson, Nikolaos Lempesis, Andrea Levy

April 25, 2023

## Notebooks Reminder

- Always access the notebooks via the rocket button on the top right of the code files and choose JupyterHub to launch [noto.epfl.ch](https://noto.epfl.ch)
- **Make sure to access noto this way each time you begin the exercise to ensure you have the latest version!**



# Exercise Structure

- Learning goals
  - Derive a time evolution integrator (e.g. Verlet)
  - Understand importance of periodic boundary conditions
  - Run a molecular dynamics simulation for a small molecule ( $\text{CO}_2$ ) in *NVE* ensemble
- Chapter in script
  - Chapter 4 - Molecular Dynamics Simulations
- Resources
  - Understanding Molecular Simulation, Frenkel & Smit, 2nd Edition - Chapter 3 & Chapter 5 (extra)
  - Computer Simulation of Liquids, Allen & Tildesley, 2nd Edition - Chapter 4

## Exercise 4 - Intro & Tips

Today we will provide you a simple Molecular Dynamics (**Toy MD**) code in python and you will extend it to run a MD simulation.

- The theoretical part introduces you to MD

Be sure to understand what we mean by:

- ergodicity
- phase space sampling
- MD propagation algorithm
- periodic boundary conditions
- In the practical part you will implement:
  - Velocity verlet algorithm
  - Periodic boundary conditions (PBS)

Tip: quickly go through the theoretical part for now, focusing on understanding the algorithms in order to be able to implement them in the second part (leave the theoretical derivations for later)

## Exercise 4 - ToyMD structure

ToyMD code structure:

- main code (propagation of MD steps) in `toy_md.py` script
- additional code for specific tasks, i.e. `toy_forces.py`
- parameters for the system in separate files (carbon-dioxide folder)



## Exercise 4 - Run ToyMD

ToyMD is a python script, which can be run

1. via terminal
2. via jupyter notebook

(see instructions in the exercise). In both cases, you will execute a bash command, passing files as arguments to the `toy_md.py` script with the following structure (paths may change):

```
python3 toy_md.py -c co2.pdb -p params.txt -ff  
force_field.txt -o traj.pdb -w co2-output.pdb
```