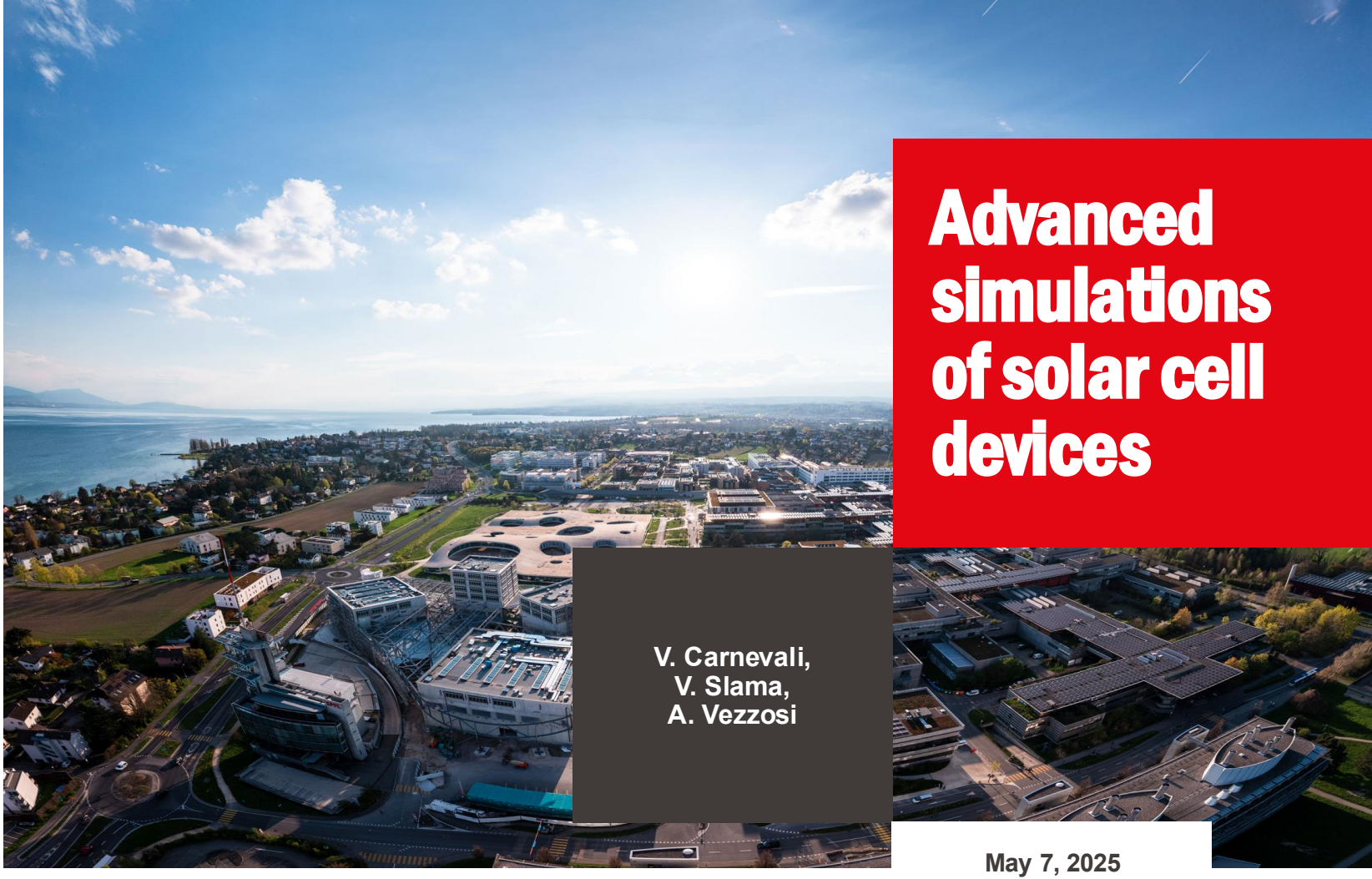
Advanced simulations of solar cell devices



V. Carnevali,
V. Slama,
A. Vezzosi

École Polytechnique fédérale de Lausanne

May 7, 2025

Why classical molecular dynamics?

Simulation of large systems - ab-initio hundreds of atoms, classical MD millions of atoms

- Interfaces
- Systems in solution
- Disordered systems
- Better ensemble averages

Longer propagation times - ab-initio few picoseconds, classical MD microseconds (with enhanced sampling even more)

- “Full" conformational sampling
- Diffusion processes
- Absorption processes
- Possible to obtain equilibrium properties

Much lower hardware requirements

- Very well parallelized on GPUs
- Usually, single workstation is enough for fairly large systems

Good for obtaining initial structures for ab-initio modelling or for studying mechanistic properties

Classical molecular dynamics

Ab-initio molecular dynamics :

$$V(R) = \langle \psi_R(r) | H_{el}(R,r) | \psi_R(r) \rangle$$

H_{el}=electronic Hamiltonian

$\psi_R(r)$=electronic wavefunction

$\boldsymbol{R}$= nuclear positions

$\boldsymbol{r}$= electronic positions

Classical molecular dynamics :

$$V(r) = \sum_{bonds} k_r (r - r_{eq})^2 \quad \textit{bond}$$

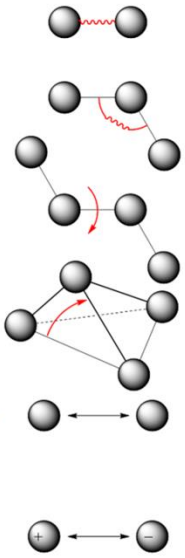
$$+ \sum_{angles} k_\theta (\theta - \theta_{eq})^2 \quad \textit{angle}$$

$$+ \sum_{dihedrals} k_\phi (1 + \cos[n\phi - \gamma]) \quad \textit{dihedral}$$

$$+ \sum_{impropers} k_\omega (\omega - \omega_{eq})^2 \quad \textit{improper}$$

$$+ \sum_{i<j}^{atoms} \varepsilon_{ij} \left[\left(\frac{r_m}{r_{ij}} \right)^{12} - 2 \left(\frac{r_m}{r_{ij}} \right)^{6} \right] \quad \textit{van der Waals}$$

$$+ \sum_{i<j}^{atoms} \frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}} \quad \textit{electrostatic}$$



One of the possible forms

- Simpler nuclear potential description
- Set of formulas for each interaction together with the parameters = **Force-Field**
- For evolution in time, the same equations of motion have to be solved

Forcefield types

- Simple description of the potential
  - fast
  - force-field tuned to specific properties and systems – we cannot have general description for all the systems as in ab-initio approaches
  - Selection of forcefield usually requires some information about the system or the processes we would like to study => It can lead to biased results depending on the quality of the forcefield

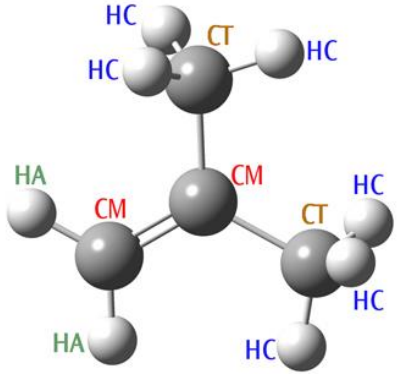
Classical forcefields:
- Fixed charges (atomic charges, multipoles)
- No chemical reactions
- Examples: AMBER (DNA and proteins), CFF (organic compounds and metals), CHARMM (small molecules and macromolecules), OPLS, IFF (Interface Force Field), …

Polarizable forcefields:
- Fixed charges + mutual polarization of atoms or molecules
- No chemical reactions
- Usually atomic polarizabilities or Drude model (charges on a spring)
- Examples: AMBER(atomic polarizabilities), AMOEBA (atomic polarizabilities and multipoles), CHARMM (Drude), …

Reactive forcefields:
- Bond breaking and chemical reactions (change of the parameters during the reaction
- Slower than classical forcefield and parameters sometimes not interpretable
- Examples: ReaxFF, EVB, …

Machine learning force-fields

Forcefield - parameters



Atom types (force-field atom types)

- Contrary to ab-initio approaches atomic or interatomic parameters are dependent on the chemical environment
- Instead of atom elements we have atom types – define atoms with the same chemical properties and same parameters
  - Usually, the first letter is element and then some unique identifier

Atom names

- Usually, all atoms within a molecule have unique names
- Important for example for the assigning atomic charges/multipoles – Even the atoms belonging to same force-field type can have different charges

Obtaining the force-field parameters

- Available force-field databases – if your system is well parametrized within some force-field it is strongly recommended to use it
- Scientific papers – new and refined parameters can be often found in literature
- Own parametrization
  - Online tools with generalized FF parameters (LibParGen, CGenFF), within MD software (antechamber for AMBER)
  - Refining parameters based on ab-initio results
  - Force matching (efficient way how to obtain FF parameters from the ab-initio MD, machine learning
  - Charge fitting (RESP, semiempirical methods e.g. AM1, CM1A, …)
  - When parametrizing only part of the system it is **important to follow the parametrization procedure**

Forcefield parameters example

(FAPbI3/01_parametrize/in.FAPbI3)

VdW interaction

```
pair_coeff    1  1  lj/cut/coul/long 0.076 3.55 #C C
pair_coeff    2  2 lj/cut/coul/long 0.0157 1.06910 # H-H
pair_coeff    8  8 buck/coul/long 72791130.3845458   0.126795228    0.0          # Pb-Pb
```

Pair of atom-type indexes

Buckingham potential

$$\Phi_{12}(r) = A\exp(-Br) - \frac{C}{r^6}$$

Bonding interaction

```
bond_coeff 1 340.4000 1.0800 # C-Ha
bond_coeff 2 448.0000 1.3650 # C-N
bond_coeff 3 434.0000 1.0100 # H-N
```

Bond type index

Strength of the harmonic potential k_r

Optimal distance r_{eq}

Optimal angle O_{eq}

Angular motion

```
angle_coeff 1 35.000 120.000 # C-N-H
angle_coeff 2 35.000 113.000 # H-N-H
angle_coeff 3 63.000 104.520 # HW-OW-HW
```

Forcefield parameters example

(FAPbI3/01_parametrize/in.FAPbI3)

Dihedral torsion

$$E = \frac{1}{2}K_1[1+\cos(\phi)] + \frac{1}{2}K_2[1-\cos(2\phi)] + \frac{1}{2}K_3[1+\cos(3\phi)] + \frac{1}{2}K_4[1-\cos(4\phi)]$$

dihedral_coeff 1 opls 3.900 0.000 0.000 0.000 # Ha-C-N-H
dihedral_coeff 2 opls 3.900 0.000 0.000 0.000 # N-C-N-H

Improper interaction (energy penatly for out-of-plane distortion)

improper_coeff 1 2.500 -1 2 # C-H-N-H
improper_coeff 2 2.500 -1 2 # Ha-N-C-N

$$E = K[1 + d\cos(n\phi)]$$

Software

- As for the force-field - different packages for different properties

LAMMPS *(Large-scale Atomic/Molecular Massively Parallel Simulator)*

- Open source
- Possible to combine different force-field types for different parts of system
- Large variability of approaches and methods
- Good for simulation of periodic systems (crystals, surfaces, …)
- Not very user friendly – Preparation of simulation takes some time

AMBER

- Commercial
- Good molecular systems in solution (mostly for bio-systems or molecular crystals)
- Many tools to help with simulation setup and parametrization – pretty quick and easy

GROMACS

- Open-source (possible to implement new features)
- Easy to use
- Mainly for the biological systems

Packmol

- Open source (python)
- Good for initial system structure preparation (no MD)

⋮

LAMMPS input

Link with files for today's lecture 10Lect_shared

```
# cell

dimension      3
boundary       p p p
processors     * * *
#neighbor       1.0 bin
neigh_modify   every 1 delay 0 check yes
units          real
atom_style     full
box tilt       large


# DATA INPUT SECTION -----------------------
include         in.temp
include         in.pressure
read_data       FAPbI3.data
include         in.setup

info system
info coeffs
# ----------------------------------------

# FIX PB AND I ATOMS
# group pb_i type 5 7
# group mobile subtract all pb_i
# fix 4 pb_i setforce 0.0 0.0 0.0
 group mobile union all
# ----------------------------------------


# ENERGY MINIMIZATION ----------------------
minimize        1.0e-5 1.0e-6 5000 50000
#write_data     data.min
# ----------------------------------------
```

LAMMPS input

```
# EXECUTION OPTIONS ----------------------------------------
timestep        0.2
velocity        mobile create 5.0 875698 mom yes rot yes dist gaussian
dump            myDump1 all dcd 500 out.1.dcd
fix             2 mobile npt temp 5.0 300.0 100.0 iso 1.0 1.0 350.0
#fix             2 mobile nvt temp 100.0 300.0 100.0
#fix             1 mobile wall/reflect zhi 85
fix             3 mobile momentum 10000 linear 1 1 1
## ------------------------------------------------------
#
#
## ---------------- PRODUCTION RUN ----------------------------------
restart           20000 cell1.rst cell2.rst

run               500000


write_restart     restart.file
write_data        data.heating_all nocoeff
# ------------------------------------------------------------------
```

LAMMPS input

Force field definition

```
# !!!CHECK AND MODIFY THIS PART ACCORDING TO YOUR SYSTEM!!!
bond_style        harmonic
angle_style       harmonic
dihedral_style    hybrid opls multi/harmonic
improper_style    cvff
pair_style        hybrid lj/cut/coul/long 10.0 8.0 buck/coul/long 10.0 8.0
pair_modify       pair lj/cut/coul/long mix geometric
kspace_style      pppm 1e-4
#kspace_modify     slab 3.0
special_bonds     amber
```

LAMMPS input

Non-bonded interactions

```
# vdW perovskite
pair_coeff      6  6 buck/coul/long 72791130.3845458   0.126795228     0.0             # Pb-Pb
pair_coeff      4  6 buck/coul/long 172493.555016667   0.3109909842    0.0             # Pb-I
pair_coeff      4  4 buck/coul/long 18994.4488183333   0.44363964      641.1935584     # I-I
pair_coeff      5  6 buck/coul/long 32690390.937995 0.150947 0.000000 # Pb-N
pair_coeff      2  6 lj/cut/coul/long 0.0140 2.26454 # Pb-H
pair_coeff      1  6 buck/coul/long 32690390.937995 0.150947 0.000000 # Pb-C
pair_coeff      3  6 lj/cut/coul/long 0.0140 2.70999 # Pb-Ha
pair_coeff      4  5 buck/coul/long 112936.714213 0.342426 0.000000 # I-N
pair_coeff      2  4 lj/cut/coul/long 0.0574 2.75000 # I-H
pair_coeff      1  4 buck/coul/long 112936.714213 0.342426 0.000000 # I-C
pair_coeff      3  4 lj/cut/coul/long 0.0574 3.10000 # I-Ha
pair_coeff      5  5 lj/cut/coul/long 0.1700 3.25000 # N-N
pair_coeff      2  2 lj/cut/coul/long 0.0157 1.06910 # H-H
pair_coeff      1  1  lj/cut/coul/long 0.076 3.55 #C C
pair_coeff      3  3 lj/cut/coul/long 0.030 2.50 #Ha Ha
```

LAMMPS input

Bonded interactions

```
#Bond Coeffs
bond_coeff    1    340.4000     1.0800  # C-Ha
bond_coeff    2    448.0000     1.3650  # C-N
bond_coeff    3    434.0000     1.0100  # H-N

#Angle Coeffs
angle_coeff    1     35.000    120.000   # C-N-H
angle_coeff    2     35.000    113.000   # H-N-H
angle_coeff    3     35.000    119.100   # Ha-C-N
angle_coeff    4     70.000    120.000   # N-C-N

# Dihedral coefficients
dihedral_coeff   1    opls   3.900      0.000      0.000      0.000   # Ha-C-N-H
dihedral_coeff   2    opls   3.900      0.000      0.000      0.000   # N-C-N-H

#impropers
improper_coeff   1    2.500      -1       2    # C-H-N-H
improper_coeff   2    2.500      -1       2    # Ha-N-C-N
```

Application

Prepare perovskite surface

- Rename hydrogens on C to Ha and hydrogens on N closer to the C to Hb in your initial cell (00_FAPbI3_2x2x2.pdb or 00_FAPbI3_1x1x1.pdb)
- Create larger super cell: "./create_supercell.py -I 01_FAPbI3_2x2x2_renamed.pdb -o 02_FAPbI3_10x10x6_renamed.pdb -s 5 5 3
- Create LAMMPS data file with system structural information:
  - Open larger system in vmd
  - Open tk console
  - Write: "source data_file_FAPbI_CHA.tcl" script sould create system.data file
  - source data_file_FAPbI_CHA.tcl contain information about the charges and atomic masses for all atoms in the system

Prepare perovskite FF parameter file

- Force-field types, bond types, angle types, ... indexes have to correspond to the generated data file
- For the perovskite parameter file is provided "in.FF_style". Take a look if you understand how it is structured

Application

Generate parameter file for your system

- Run "./Parametrize_lammps_Mattoni.py -d FAPbI3.data -o in.FAPbI3"
- In case of additives or water it is possible to specify the structure and parameter files and produce FF for the full system as: "./Parametrize_lammps_Mattoni.py -d FAPbI3.data -l WAT.lmp WAT.pdb -o in.FAPbI3"
- The perovskite is described by both Buckingham and LJ potential. For interaction with additives only LJ is used => code in current version compute all the pair interactions with LJ. It is important to replace the parameters for the FAPbI3 with the parameters specified in FF_vdW_perovskite.dat:

```
# vdW perovskite
pair_coeff 6 6 buck/coul/long 72791130.3845458 0.126795228 0.0 # Pb-Pb
pair_coeff 4 6 buck/coul/long 172493.555016667 0.3109909842 0.0 # Pb-I
pair_coeff 4 4 buck/coul/long 18994.4488183333 0.44363964 641.1935584 # I-I
pair_coeff 5 6 buck/coul/long 32690390.937995 0.150947 0.000000 # Pb-N
pair_coeff 2 6 lj/cut/coul/long 0.0140 2.26454 # Pb-H
pair_coeff 1 6 buck/coul/long 32690390.937995 0.150947 0.000000 # Pb-C
pair_coeff 3 6 lj/cut/coul/long 0.0140 2.70999 # Pb-Ha
pair_coeff 4 5 buck/coul/long 112936.714213 0.342426 0.000000 # I-N
pair_coeff 2 4 lj/cut/coul/long 0.0574 2.75000 # I-H
pair_coeff 1 4 buck/coul/long 112936.714213 0.342426 0.000000 # I-C
pair_coeff 3 4 lj/cut/coul/long 0.0574 3.10000 # I-Ha
pair_coeff 5 5 lj/cut/coul/long 0.1700 3.25000 # N-N
pair_coeff 2 2 lj/cut/coul/long 0.0157 1.06910 # H-H
pair_coeff 1 1 lj/cut/coul/long 0.076 3.55 #C C
pair_coeff 3 3 lj/cut/coul/long 0.030 2.50 #Ha Ha
```

Application

Run initial FA orientation equilibration

- Surfaces are in general more prone to defects compared to the bulk due to the cut and exposure to vacuum (in the simulation)
- It is important to slowly equilibrate the surface to have reasonable starting structure
- First, we run NPT simulation to equilibrate the box dimension together with atomic coordinates. For NPT run we need 3D periodic boundary condition =>
  - Comment line: "# kspace_modify slab 3.0" in FF parameter file (in.FAPbI3)
  - Submit LAMMPS job

Application

Equilibration at NVT

- Uncomment line: "kspace_modify slab 3.0" in FF parameter file (in.FAPbI3)
- Change LAMMPS input file

NEW

```
# cell
dimension       3
boundary        p p f
processors      * * *
#neighbor       1.0 bin
neigh_modify    every 1 delay 0 check yes
units           real
atom_style      full
box tilt        large


# DATA INPUT SECTION -----------------------------------
include         in.temp
include         in.pressure
read_data       data.heating_all
include         in.setup

info system
info coeffs
# ----------------------------------------------------

+-- 10 lines: FIX_PB_AND_I_ATOMS----------------------------------------------
#write_data     data.min
# ----------------------------------------------------------------------------


# EXECUTION OPTIONS ------------------------------------
timestep        0.2
#velocity       mobile create 5.0 875698 mom yes rot yes dist gaussian
dump            myDump1 all dcd 500 out.1.dcd
#fix            2 mobile npt temp 5.0 300.0 100.0 iso 1.0 1.0 350.0
fix             2 mobile nvt temp 300.0 300.0 100.0
#fix            1 mobile wall/reflect zhi 85
fix             3 mobile momentum 10000 linear 1 1 1
## ----------------------------------------------------
#
#
## ---------------- PRODUCTION RUN ----------------------------------------
restart         20000 cell1.rst cell2.rst

run             500000


write_restart   restart.file
write_data      data.equil nocoeff
# ----------------------------------------------------------------------------
```

OLD

```
# cell
dimension       3
boundary        p p p
processors      * * *
#neighbor       1.0 bin
neigh_modify    every 1 delay 0 check yes
units           real
atom_style      full
box tilt        large


# DATA INPUT SECTION -----------------------------------
include         in.temp
include         in.pressure
read_data       FAPbI3.data
include         in.setup

info system
info coeffs
# ----------------------------------------------------

+-- 10 lines: FIX_PB_AND_I_ATOMS----------------------------------------------
#write_data     data.min
# ----------------------------------------------------------------------------


# EXECUTION OPTIONS ------------------------------------
timestep        0.2
velocity        mobile create 5.0 875698 mom yes rot yes dist gaussian
dump            myDump1 all dcd 500 out.1.dcd
fix             2 mobile npt temp 5.0 300.0 100.0 iso 1.0 1.0 350.0
#fix            2 mobile nvt temp 100.0 300.0 100.0
#fix            1 mobile wall/reflect zhi 85
fix             3 mobile momentum 10000 linear 1 1 1
## ----------------------------------------------------
#
#
## ---------------- PRODUCTION RUN ----------------------------------------
restart         20000 cell1.rst cell2.rst

run             500000


write_restart   restart.file
write_data      data.heating_all nocoeff
# ----------------------------------------------------------------------------
```

Not periodic in z

Read .data file from previous simulation

Do not reassign velocities (use the ones from previous simulation)

Use NVT instead of NPT

Rename the output .data file

Application

Setting up the system with water (FAPbI3_WAT/00_prepare folder)

- Solvate the system with water
  - Useful tool is packmol. (optional you can try to download "https://m3g.github.io/packmol/", install and use it with the provided input script: " packmol < packmol.inp"
  - You should obtain solvated system with water layer – files provided: "FAPbI_WAT_10x10x6.pdb"
  - After generating the fille add box definition from the initial perovskite structure (line starting with CRYST1)
- Generate data file with VMD
  - Open pdb file with water layer (vmd FAPbI_WAT_10x10x6.pdb)
  - Open tk console
  - Execute "*source data_file_FAPbI_CHA.tcl*"
  - Rename the system.data to FAPbI_WAT_10x10x6.data

Parametrization (FAPbI3_WAT/01_prepare folder)

- Run parametrization with "./Parametrize_lammps_Mattoni.py -d FAPbI_WAT_10x10x6.data -i WAT_tip3p.lmp WAT_tip3p.pdb -o in.FAPbI3_WAT"
- Correct vdW potential parameters in in.FAPbI3_WAT parameter file (copy the ones in FF_vdW_perovskite.dat and remove the existing ones in in.FAPbI3_WAT)

Application: Water on $FAPbI_3$ slab

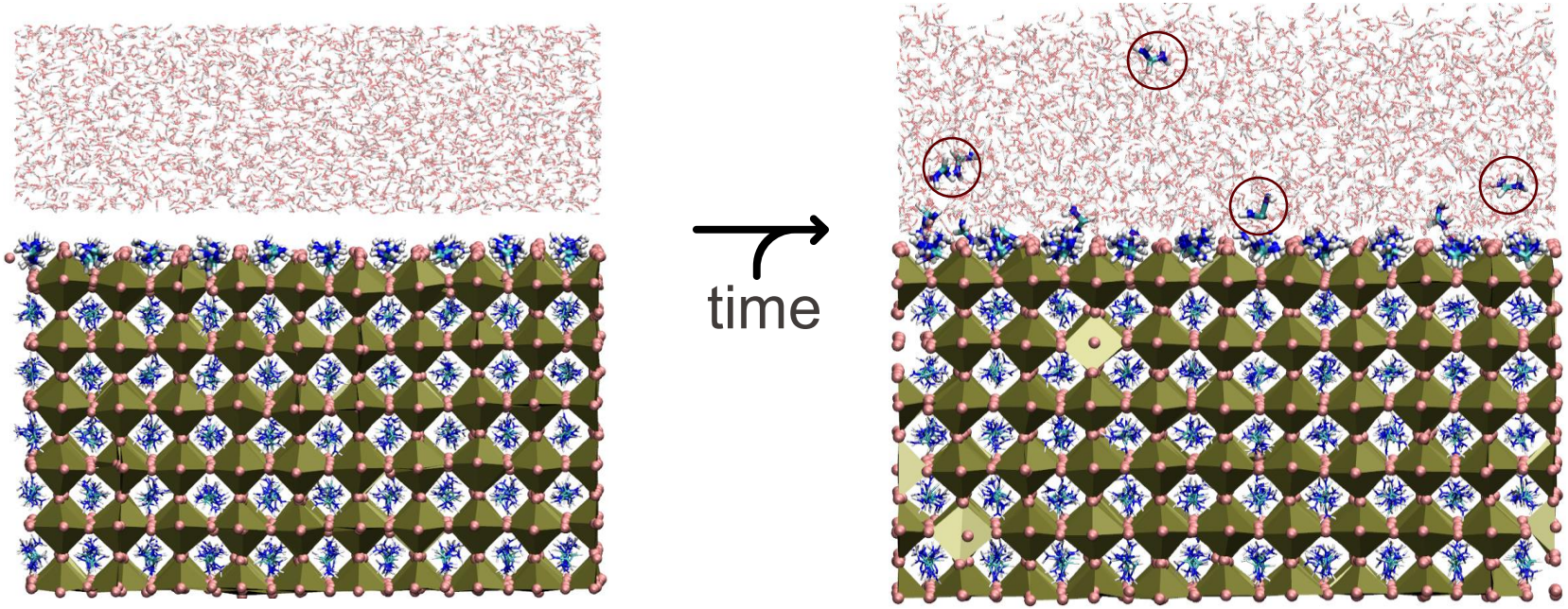
Initial minimization and short equilibration (FAPbI3_WAT/02_equilibrate_NVT)

- After building the system we need to minimize the geometry to avoid close contacts:
  - Uncomment line "minimize 1.0e-5 1.0e-6 5000 50000" in lammps.inp file
  - Because we don't have periodic system in z we need to make sure that the water is not going toward the boundary of the box => adding reflection wall in lammps.in file:
    "fix 1 mobile wall/reflect zhi 90"

Running simulation with SHAKE (FAPbI3_WAT/03_equilibrate_shake_NVT)

- SHAKE algorithm fixes the bond length and angle of the water molecules => rigit body
  - Allow longer propagation timesteps 1-2 fs (without 0.5 fs or lower)
  - Turned on by adding line in lammps.in file: "fix 8 all shake 0.001 20 10 b 4 a 3"
    - 0.001 = tolerance for the SHAKE algorithm
    - 20 maximum iteration of SHAKE
    - 10 frequency of printing SHAKE info (in steps)
    - b 4 = fix bond type with index 4 (OW-HW)
    - a 3 = fix angle type with index 3 (HW-OW-HW)

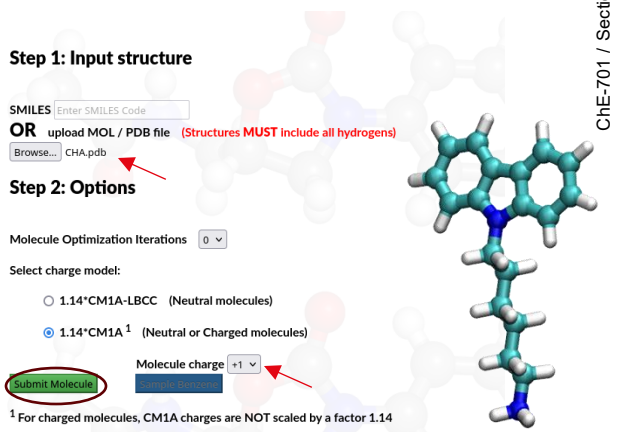
Application: Water on $FAPbI_3$ slab



Surface passivation with Carbazole molecules

Setting up the system (FAPbI3_CHA/00_prepare folder)

- Carbazole molecule – file provided "FAPbI3-CHA_10x10x6.pdb"
  - You can also build the molecule yourself (see prev. lesson)
  - Build the system by replacing surface FA with CHA (file is provided)
- Generate data file with VMD
  - Open pdb file with CHA molecules (vmd FAPbI3-CHA_10x10x6.pdb)
  - Open tk console
  - Execute "*source data_file_FAPbI_CHA.tcl*"
  - Rename the system.data to FAPbI3-CHA_10x10x6.data



LAMMPS

CHA.pdb

LAMMPS

Parametrization (FAPbI3_WAT/01_parametrize)

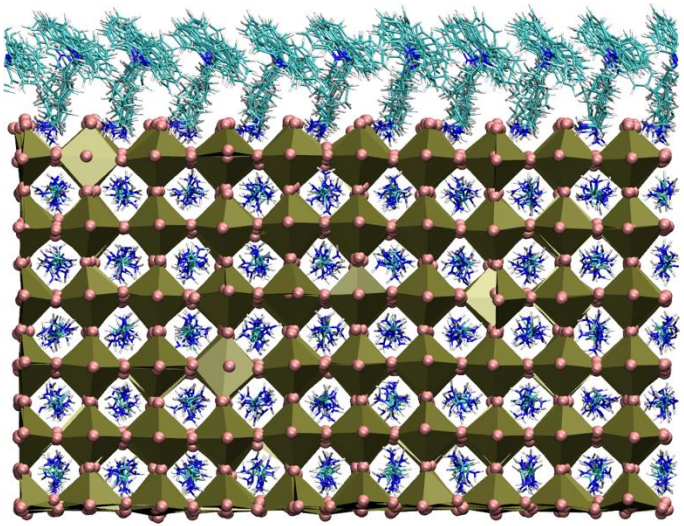
- Get generalized FF parameters from LigParGen (https://traken.chem.yale.edu/ligpargen/)
- Run parametrization with "python Parametrize_lammps_Mattoni.py -d FAPbI3-CHA_10x10x6.data -i CHA.lmp CHA.pdb -o in.FAPbI3_CHA_tmp"
- Correct vdW potential parameters in in.FAPbI3_CHA parameter file (copy the ones in FF_vdW_perovskite.dat and remove the existing ones in FF_vdW_perovskite.dat)
- Copy final parameter file to the simulation folder 02_equilibrate_NVT

10Lect_shared

Surface passivation with Carbazole molecules

Perform minimization and short equilibration (FAPbI3_CHA/02_equilibrate_NVT)

Save the last snapshot of the simulation for the next task !



Surface passivation with Carbazole molecules

Setting up the system with water (FAPbI3_CHA_WAT/00_prepare folder)

- Solvate the system with water
  - Using packmol solvate the last snapshot of the NVT simulation "FAPbI3-CHA_10x10x6_NVT_final.pdb"
  - You should obtain solvated system with water layer – files provided: "FAPbI_WAT_10x10x6.pdb"
  - After generating the file add box definition from the initial structure (line starting with CRYST1)
- Generate data file with VMD
  - Open pdb file with water layer (vmd FAPbI_WAT_10x10x6.pdb)
  - Open tk console
  - Execute "*source data_file_FAPbI_CHA.tcl*"
  - Rename the system.data to FAPbI_CHA_WAT_10x10x6.data

Parametrization (FAPbI3_CHA_WAT/01_parametrize)

- Run parametrization with "python3 Parametrize_lammps_Mattoni.py -d FAPbI_CHA_WAT_10x10x6.data -i CHA.lmp CHA.pdb -i WAT_tip3p.lmp WAT_tip3p.pdb -o in.FAPbI3_CHA_WAT_tmp"
- Correct vdW potential parameters in in.FAPbI3_CHA_WAT parameter file (copy the ones in FF_vdW_perovskite.dat and remove the existing ones in in.FAPbI3_CHA_WAT)

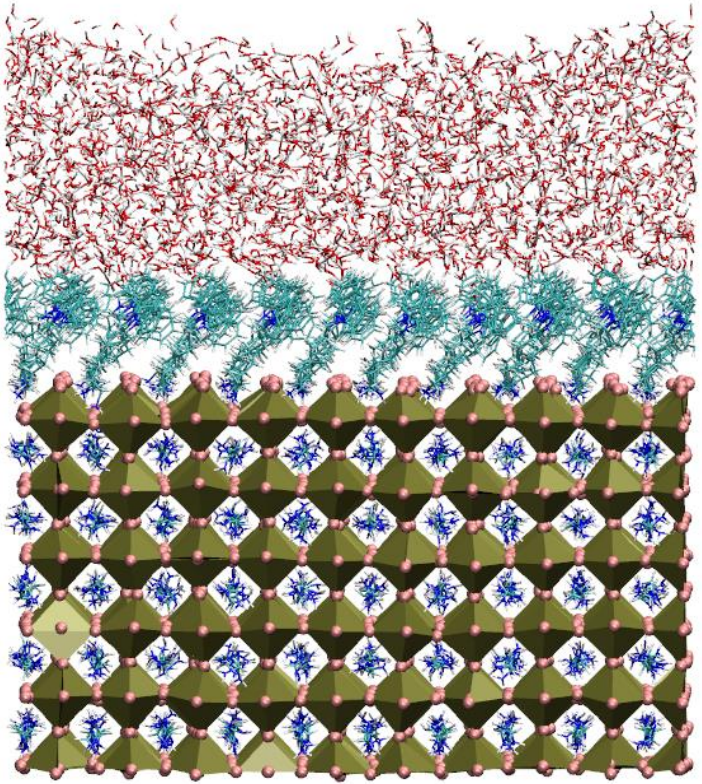
Surface passivation with Carbazole molecules

- Now solvate the $FAPbI_3$ passivated with CHA molecule that you have simulated in the previous task. You can follow the same procedure used for the pristine $FAPbI_3$ surface.

- Run a short NVT simulation and check that the organic hydrophobic layer protect the perovskite surface from degradation !

…

Surface passivation with Carbazole molecules

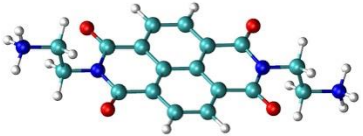


Projects

1- AIMD – band alignment

$FAPbI_3$ surface passivated with NDI molecules

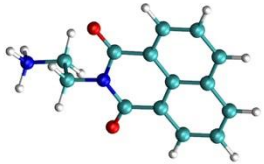
- NDI binding energy compared with FA
- Understand SAM conformation
- Band alignment: PBE vs PBE0: is NDI electroactive?



2- AIMD – band alignment

$FAPbI_3$ surface passivated with NI molecules

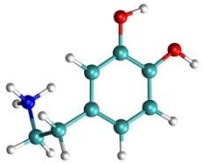
- NI binding energy compared with FA
- Understand SAM conformation
- Band alignment: PBE vs PBE0: is NI electroactive?

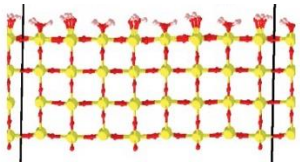


3- AIMD – ETL/additives/AL interface

SnO_2/Dopamonium/$FAPbI_3$ interface

- What happen to the dopamonium when in contact with SnO_2?
- Can you imagine any preferential binding patter of dopamonium at the interface?



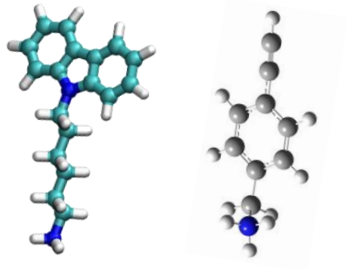


Projects

4- MD – interplay between 2 different additives

$FAPbI_3$ surface passivated with carbazoles and BMAA

- SAM with percentages according to the binding energies
- Surface protection in presence of water layer: comparison with pure carbazole layer



5- MLFF – Training an MLP for $FAPbI_3$ (Incoming lecture!)

The aim of the project is to train an MLP model for photoactive cubic phase of $FAPbI_3$ and validate it.

- Data conversion using dpdata
- Train the force field using DeepMD
- Compare accuracy and performances for different cut-off radii
- Test the FF with actual MD simulations and validate structural properties (RDF, Octahedra tilting etc.)