

# Introduction to Molecular Dynamics (MD) simulations

Objectives:

- (1) Learn the general algorithm of MD
- (2) Learn the general procedure of doing MD

# Algorithm for Molecular Simulations

## 1. Input initial conditions

Potential interaction  $V$  as a function of atom positions

Positions  $\mathbf{r}$  of all atoms in the system

Velocities  $\mathbf{v}$  of all atoms in the system

↓

---

repeat 2,3,4 for the required number of steps:

---

## 2. Compute forces

The force on any atom

$$\mathbf{F}_i = -\frac{\partial V}{\partial \mathbf{r}_i}$$

is computed by calculating the force between non-bonded atom pairs:

$$\mathbf{F}_i = \sum_j \mathbf{F}_{ij}$$

plus the forces due to bonded interactions (which may depend on 1, 2, 3, or 4 atoms), plus restraining and/or external forces.

The potential and kinetic energies and the pressure tensor may be computed.

↓

## 3. Update configuration

The movement of the atoms is simulated by numerically solving Newton's equations of motion

$$\frac{d^2 \mathbf{r}_i}{dt^2} = \frac{\mathbf{F}_i}{m_i}$$

or

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{v}_i; \quad \frac{d\mathbf{v}_i}{dt} = \frac{\mathbf{F}_i}{m_i}$$

↓

## 4. if required: Output step

write positions, velocities, energies, temperature, pressure, etc.

# Algorithm of MD simulations

- So, we need
  - initial configurations (position and velocities)
  - Forces (we will talk in detail about this in the force field session)
  - Way to propagate our system
- We will see how to set up our initial configuration in the example we do.
- We will talk little about the forces and related details
- We will then talk about our MD parameters that relate to how we propagate our system.

## MD → Gromacs translation

- Initial positions and velocities:
  - This is called system initialization
  - Ultimately, we want a coordinate file that has the positions of all our atoms that we can use as a starting configuration.
- Forces
  - We will specify details of this in the topology file
  - We will give some more details in the MDP file (gromacs molecular dynamics parameter file)
- Propagation
  - Mostly done in the MDP file.

Topology file: Defining molecules

# Gromacs topology file

Types of atoms in the system



Define the molecules in your system: atoms, and how they are connected



Water is defined a bit special to keep it rigid

Define the system: how many molecules of each type are in your system?



```
(base) [ss@~/Gromacs-Tutorial/spcesims] more spce_MASTER.top
[ defaults ]
; nbfunc      comb-rule      gen-pairs      fudgeLJ  fudgeQQ
1             2             no             1.0      1.0

[ atomtypes ]
OW      15.9994  -0.8476  A   3.16557e-01  6.50629e-01
HW      1.0080   0.4238  A   0.00000e+00  0.00000e+00

[ moleculetype ]
; molname      nrexcl
SOL            2

[ atoms ]
; id  at type      res nr  res name  at name  cg nr  charge  mass
1    OW      1      SOL      OW       1      -0.8476 15.99940
2    HW      1      SOL      HW1      1      0.4238  1.00800
3    HW      1      SOL      HW2      1      0.4238  1.00800

[ settles ]
; OW      funct  doh      dhh
1         1      0.1      0.16330

[ exclusions ]
1         2      3
2         1      3
3         1      2

[ system ]
Pure SPCE water

[ molecules ]
SOL      4055
```

# Flexible water topology

Types of atoms in the system →

Define the molecules in your system: atoms, and how they are connected →

Water is defined a bit special to keep it rigid

Define the system: how many molecules of each type are in your system? →

```
; nbfunc      comb-rule      gen-pairs      fudgeLJ fudgeQQ
1             2             no             1.0    1.0

[ atomtypes ]
; name      mass      charge      ptype      sigma      epsilon
; Water parameters are those of TIP3P model
OW      15.9994      -0.834      A      3.15061e-01  6.36386e-01
HW      1.008        0.417      A      0.000        0.000

; molecules in the system are defined here. These molecules comprise
; the above-listed atomtypes

[ moleculetype ]
; name of molecule
; molname nrexcl
SOL      2
; atoms that form the molecule
[ atoms ]
1      OW      1      SOL      OW      1      -0.834
2      HW      1      SOL      HW1     1      0.4170
3      HW      1      SOL      HW2     1      0.4170

[ bonds ]
; i      j      funct      length      force
1      2      1      0.09572  502416.0
1      3      1      0.09572  502416.0
2      3      1      0.15139  502416.0

; further details
[ exclusions ]
1      2      3
2      1      3
3      1      2

; define the system here
[ system ]

TIP3P water

[ molecules ]
SOL 216
```

What would the topology file for butane look like?



## Butane topology

- [defaults]
- [atomtypes]
- [moleculetype]
  - [atoms]
  - [bonds]
  - [angles]
  - [dihedrals]
- [system]
  - [molecules]

## Butane+water topology

- [defaults]
- [atomtypes]
- [moleculetype]
  - [atoms]
  - [bonds]
  - [angles]
  - [dihedrals]
- [moleculetype]
  - [atoms]
  - [bonds]
  - [angles]
  - [dihedrals]
- [system]
  - [molecules]

Molecular dynamics parameters (MDP)

# Input parameters for MD simulations

```
dt                = 0.002           ; time step
nsteps            = 50000            ; number of steps
nstcomm          = 10                ; reset c.o.m. motion
nstcalcenergy     = 10
nstxout           = 100              ; write coords
nstvout           = 100              ; write velocities
nstlog            = 100              ; print to logfile
nstenergy         = 100              ; print energies
compressed-x-grps = System
nstxout-compressed = 100
nstlist           = 10                ; update pairlist
coulombtype       = PME              ; cut-off for vdw
rvdw              = 0.85              ; cut-off for coulomb
rcoulomb          = 0.85              ; cut-off for coulomb
rlist             = 0.85
Tcoupl            = v-rescale
ref_t             = 300.0
tc-grps           = System
tau_t             = 0.5
Pcoupl            = Berendsen
Pcoupltype        = isotropic        ; pressure geometry
tau_p             = 0.5               ; p-coupling time
compressibility    = 4.5e-5           ; compressibility
ref_p             = 1.0               ; ref pressure
DispCorr          = EnerPres         ; long range correction
gen_vel           = yes               ; generate init. vel
gen_temp          = 300               ; init. temp.
gen_seed          = 372340            ; random seed
constraints       = hbonds            ; constraining bonds with H
constraint_algorithm = shake
```

# Input parameters for MD simulations

- What time steps should we use?
  - Depends on the fastest timescale in the system ←
  - Examples of some timescales:

Motion	Timestep (1 fs = $10^{-15}$ s)
Molecules (bond vibration)	0.5-1.0 fs
Molecules (rigid bonds)	2 fs
Atoms (translation)	5-10 fs

- Usually for most of our simulations we use 2 fs – especially because we use rigid water molecules.
- Constraints – LINCS, SHAKE, RATTLE, (SETTLE)

# Input parameters for MD simulations: Integration

- Integrators: What options and which do we select?
  - Duplicate classical trajectory as closely as possible
  - Satisfy known conservation laws for energy and momentum
  - Be time reversible
  - Show be fast and require little memory
  - Simple in form and easy to program
- Default in Gromacs: Leap frog
- You can also use velocity-verlet algorithm

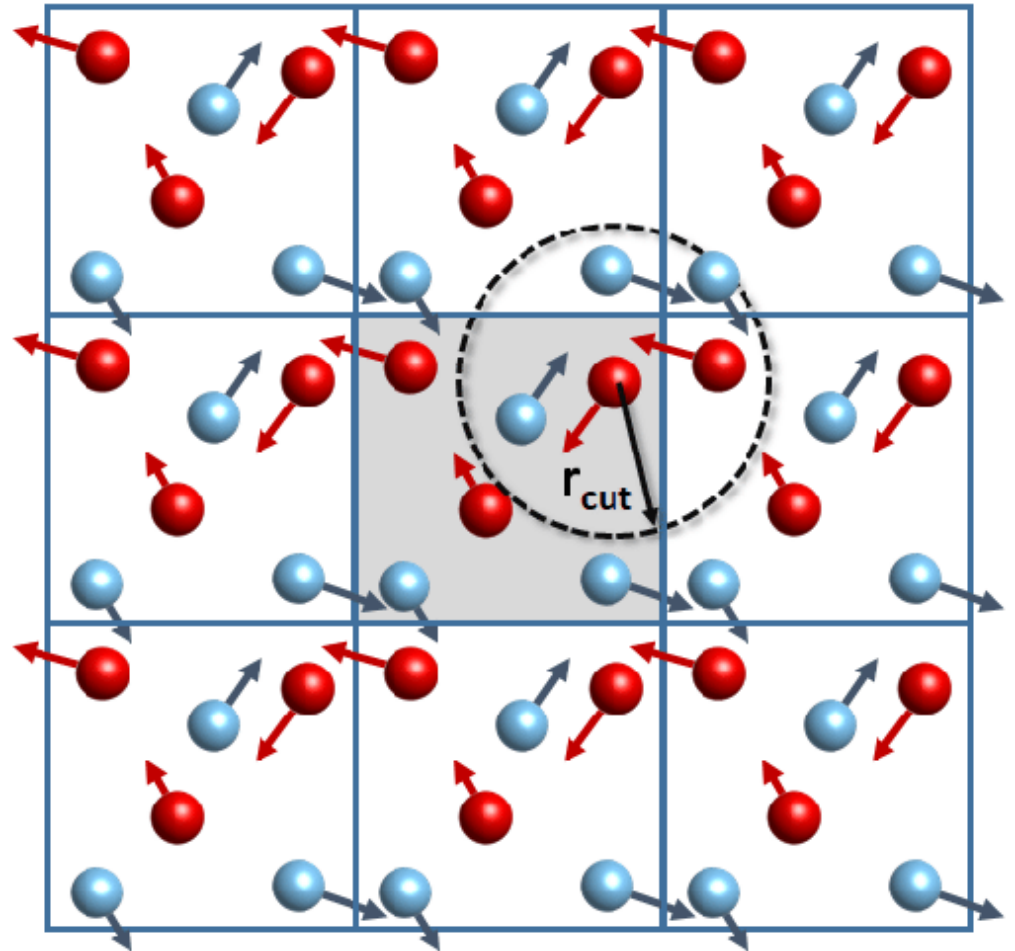
$$\begin{aligned}\mathbf{r}(t + \Delta t) &= \mathbf{r}(t) + \Delta t \mathbf{v} + \frac{\Delta t^2}{2m} \mathbf{F}(t) \\ \mathbf{v}(t + \Delta t) &= \mathbf{v}(t) + \frac{\Delta t}{2m} [\mathbf{F}(t) + \mathbf{F}(t + \Delta t)]\end{aligned}$$

- How big a system should we simulate?
- What should we do with the walls of the box?

# Input parameters for MD simulations

- Periodic boundary conditions:
  - Captures the bulk like behavior of liquids
  - This takes care of the walls.
  - Finite-size effects
- $U_{tot} = \sum_{i,j,n} u(|r_{ij} + nL|)$

Infinite sum!



# Input parameters for MD simulations

- How big a system should we simulate? → can we use a cut-off and get away with it?

- Consider a potential of type  $u(r) = \frac{a}{r^n}$ 
  - We can write the potential of energy after some cut-off distance  $r_c$  as

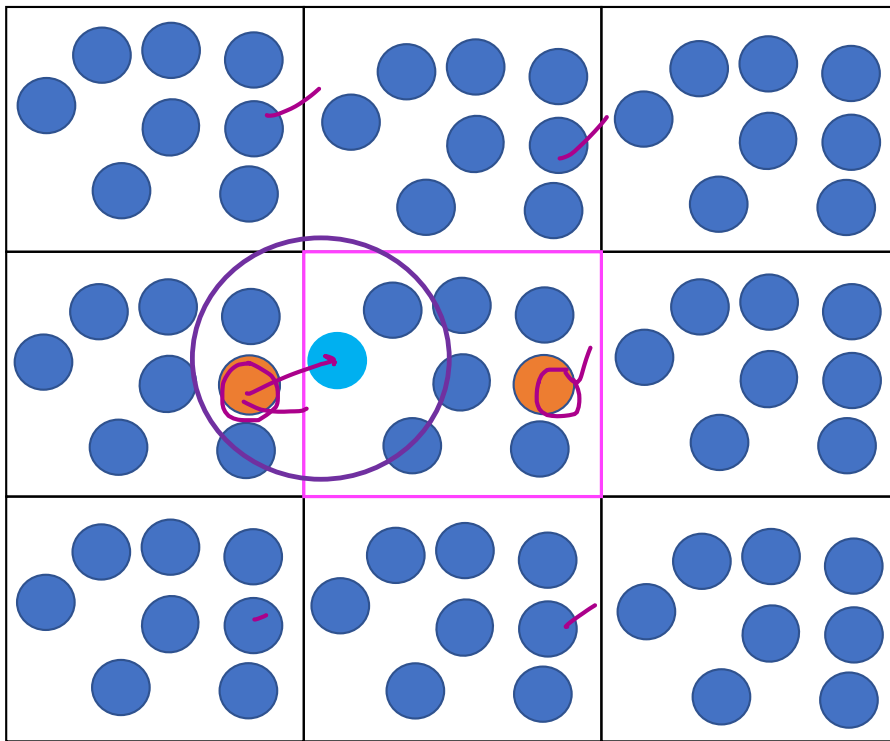
$$U_{tail} = \frac{2\pi N^2}{V} \int_{r_c}^{\infty} u(r) r^2 dr$$

This will diverge for  $n \leq 3$ ; and short range for  $n > 3$

- LJ is short-range and we can use a correction to account for the cut-off – it works for most part.
  - There are different approaches to handle the discontinuity at the cut-off
- Long range – Electrostatics – often a problem. We have special methods to handle this.
  - Ewald summation
  - Particle mesh Ewald



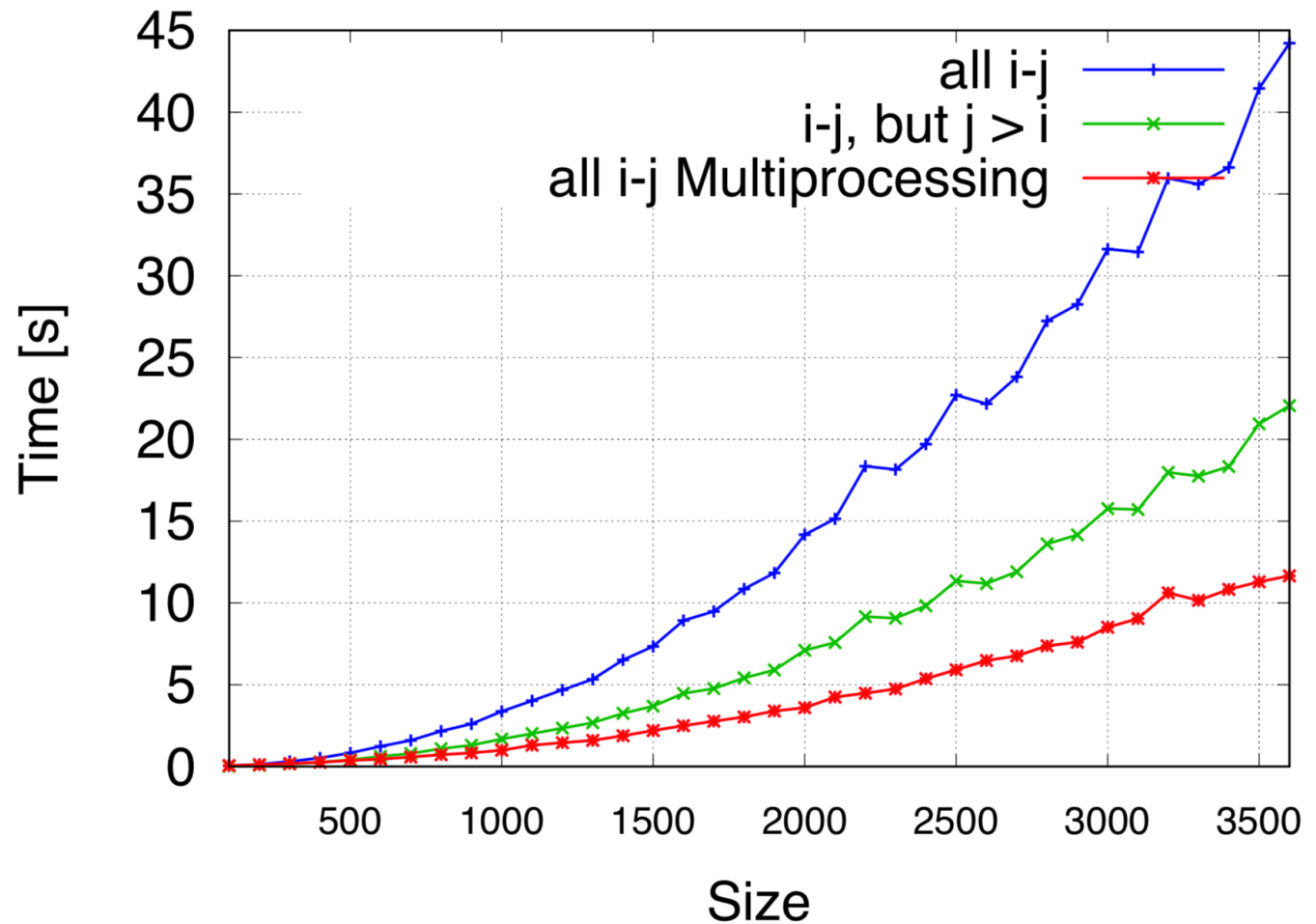
# Input parameters for MD simulations: Minimum Image Convention



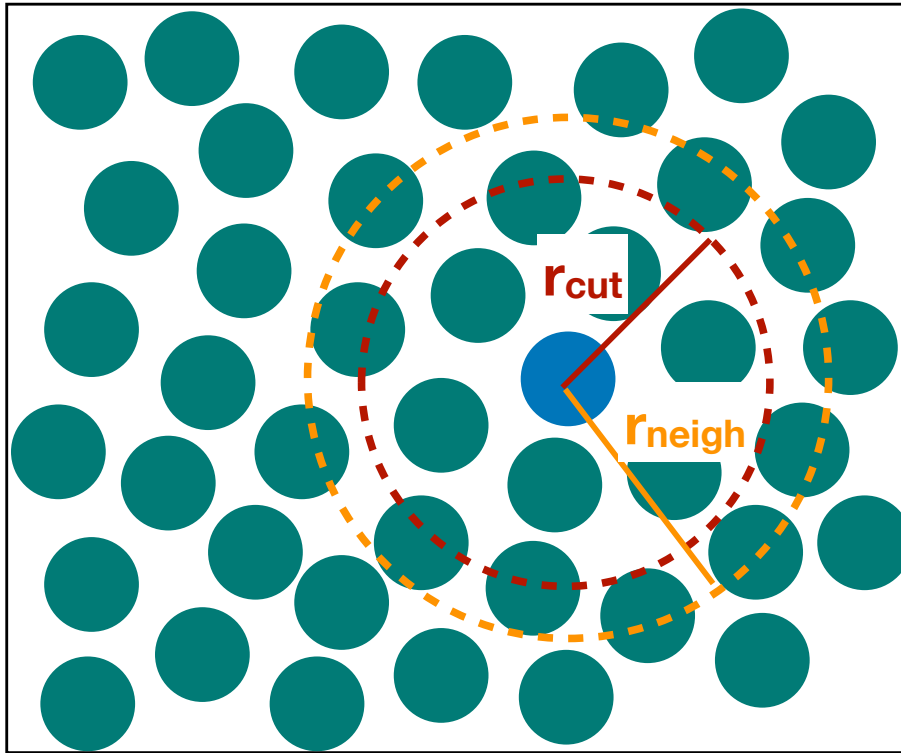
We find the nearest image of a molecule to calculate interactions with.

# Input parameters for MD simulations: Neighbor calculations

- We still have to calculate the distances between the atom pairs. So how does it help?
- We have algorithms to speed this up – neighbor lists



# Input parameters for MD simulations: Neighbor calculations



- Pick some  $r_{\text{neigh}} > r_{\text{cut}}$
- Build neighbor list using  $r_{\text{neigh}}$  every  $x$  (e.g. 10) steps
- At each step, only calculate pairwise distances between atoms on neighbor list

How do we pick  $r_{\text{neigh}}$  and the frequency to update neighbor lists?

# Input parameters for MD simulations

```
dt                = 0.002      ← ; time step
nsteps            = 50000      ; number of steps
nstcomm          = 10         ; reset c.o.m. motion
nstcalcenergy     = 10
nstxout           = 100       ; write coords
nstvout           = 100       ; write velocities
nstlog            = 100       ; print to logfile
nstenergy         = 100       ; print energies
compressed-x-grps = System
nstxout-compressed = 100
nstlist           = 10        ; update pairlist
coulombtype       = PME      ←
rvdw              = 0.85     ; cut-off for vdw
rcoulomb          = 0.85     ; cut-off for coulomb
rlist             = 0.85     ; cut-off for coulomb
Tcoupl            = v-rescale
ref_t             = 300.0
tc-grps           = System
tau_t             = 0.5
Pcoupl            = Berendsen
Pcoupltype        = isotropic ; pressure geometry
tau_p             = 0.5       ; p-coupling time
compressibility    = 4.5e-5    ; compressibility
ref_p             = 1.0       ; ref pressure
DispCorr          = EnerPres ← ; long range correction
gen_vel           = yes       ; generate init. vel
gen_temp          = 300       ; init. temp.
gen_seed          = 372340    ; random seed
constraints       = hbonds    ; constraining bonds with H
constraint_algorithm = shake
```

Input parameters for MD simulations:

Thermostats and Barostats

