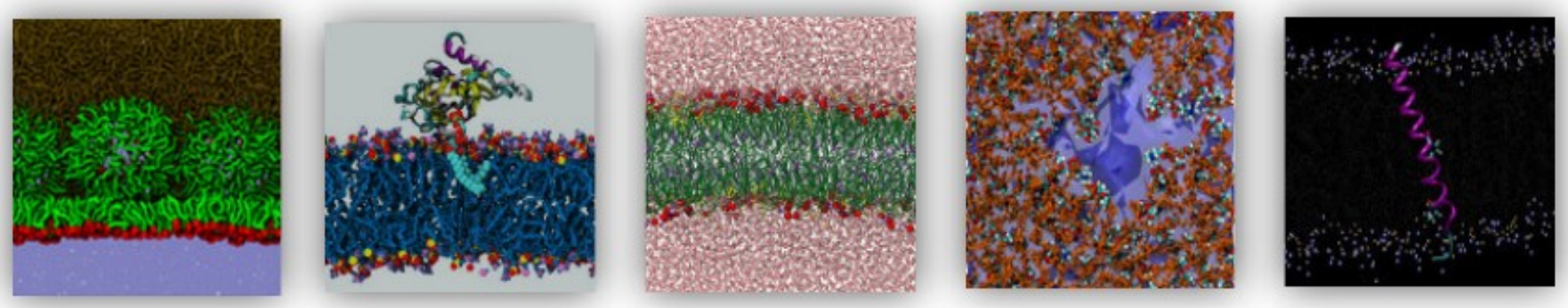


CSC spring school

26th April 2023

Molecular dynamics simulations in biosystems

MD introduction

Hector Martinez-Seara

Institute of Organic Chemistry and Biochemistry of the Czech Academy of Sciences

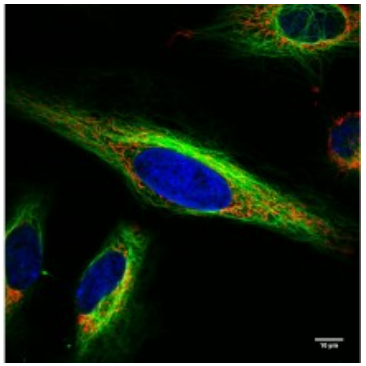
Acknowledge

Based on a course by **Prof. Lukasz Cwiklik** group leader at
J.Heyrovsky Institute of Physical Chemistry
Czech Academy of Sciences at Prague, Czech Republic

http://cwiklik.net

Computer simulations

Physical system
(cluster of atoms, water-air interface,
cell membrane, protein in water)

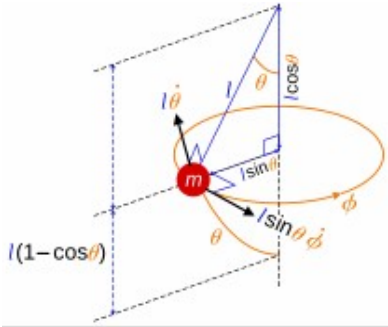


8x5is at Wikimedia Commons
CC BY-SA

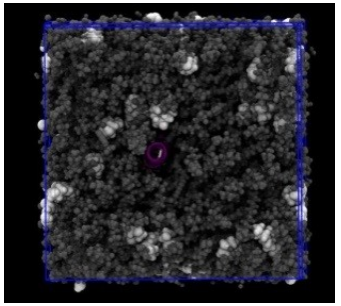
Experiment
(IR spectroscopy, NMR,
X-ray diffraction,
fluorescence microscopy,
enzymatic kinetics)

Zeiss at Wikimedia Commons
CC BY-SA

Theory
(quantum mechanics,
thermodynamics,
chemical kinetics)



Simulations
(Molecular Dynamics, Monte Carlo, Dissipative Particle Dynamics, Computational Fluid Dynamics)

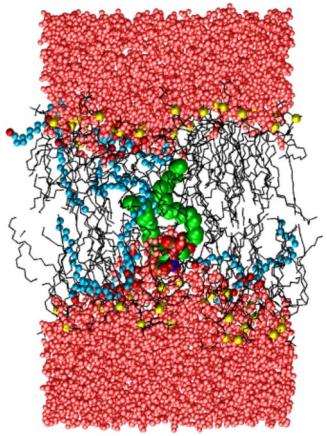
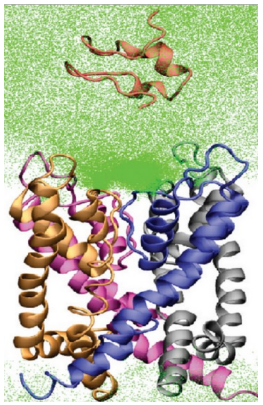
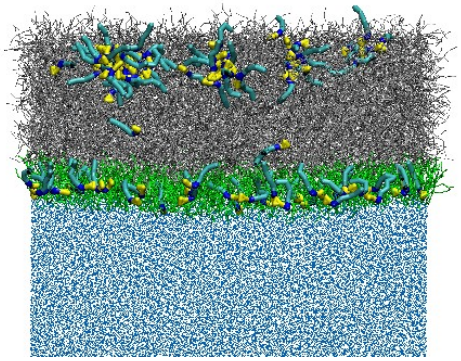
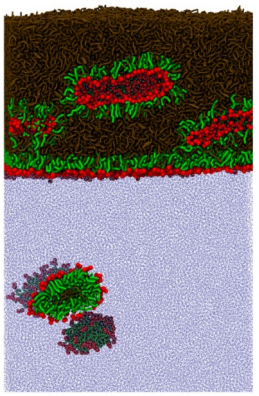
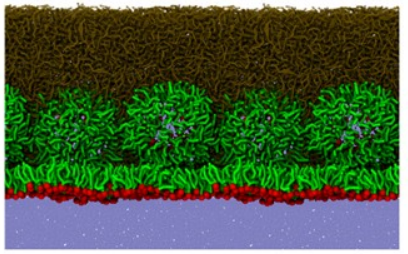


Simulations:

- theory – based on a model
- “computational experiment”
- “computational microscope”

The idea

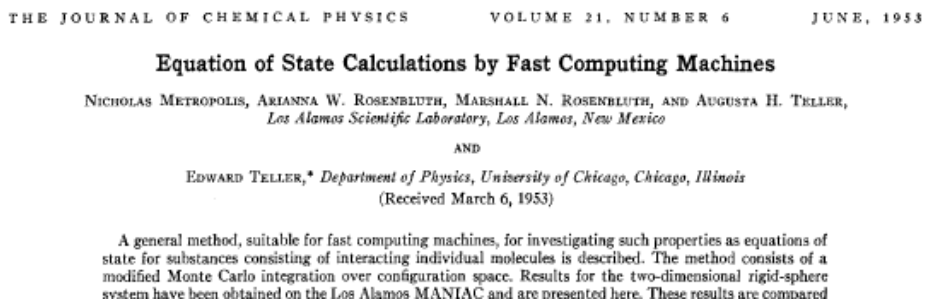
Molecular simulations = **simulations of molecules/atoms**

- What they are good for:
  - Condensed phases & interfaces (bulk of water, surface of solid,...)
  - Biological systems (proteins, membranes, sugars, biomolecules)
    - Interactions between biomolecules, drugs.
    - Provide mechanistic details about the microscopic behavior

Computer simulations - History

- Monte Carlo:
  Ulam, von Neumann,
  Metropolis
  Los Alamos labs, 1953
- Molecular Dynamics:
  Alder and Wainwright
  Livermore labs, 1957
  (hard spheres dynamics)
- Molecular Dynamics, Nobel Prize in Chemistry 2013:
  Karplus, Levitt, Warshel

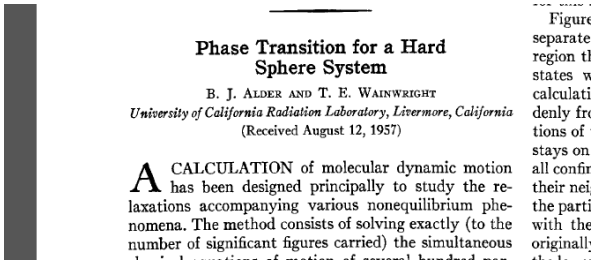


THE JOURNAL OF CHEMICAL PHYSICS VOLUME 21, NUMBER 6 JUNE, 1953

Equation of State Calculations by Fast Computing Machines

NICHOLAS METROPOLIS, ARIANNA W. ROSENBLUTH, MARSHALL N. ROSENBLUTH, AND AUGUSTA H. TELLER,
Los Alamos Scientific Laboratory, Los Alamos, New Mexico
AND
EDWARD TELLER,* *Department of Physics, University of Chicago, Chicago, Illinois*
(Received March 6, 1953)

A general method, suitable for fast computing machines, for investigating such properties as equations of state for substances consisting of interacting individual molecules is described. The method consists of a modified Monte Carlo integration over configuration space. Results for the two-dimensional rigid-sphere system have been obtained on the Los Alamos MANIAC and are presented here. These results are compared

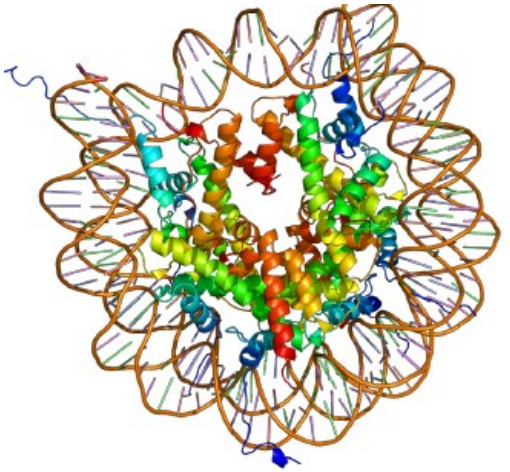


Phase Transition for a Hard Sphere System

B. J. ALDER AND T. E. WAINWRIGHT
University of California Radiation Laboratory, Livermore, California
(Received August 12, 1957)

A CALCULATION of molecular dynamic motion has been designed principally to study the relaxations accompanying various nonequilibrium phenomena. The method consists of solving exactly (to the number of significant figures carried) the simultaneous

Figure
separate
region t
states w
calculati
denly fr
tions of
stays on
all confir
their nei
the parti
with the
originall

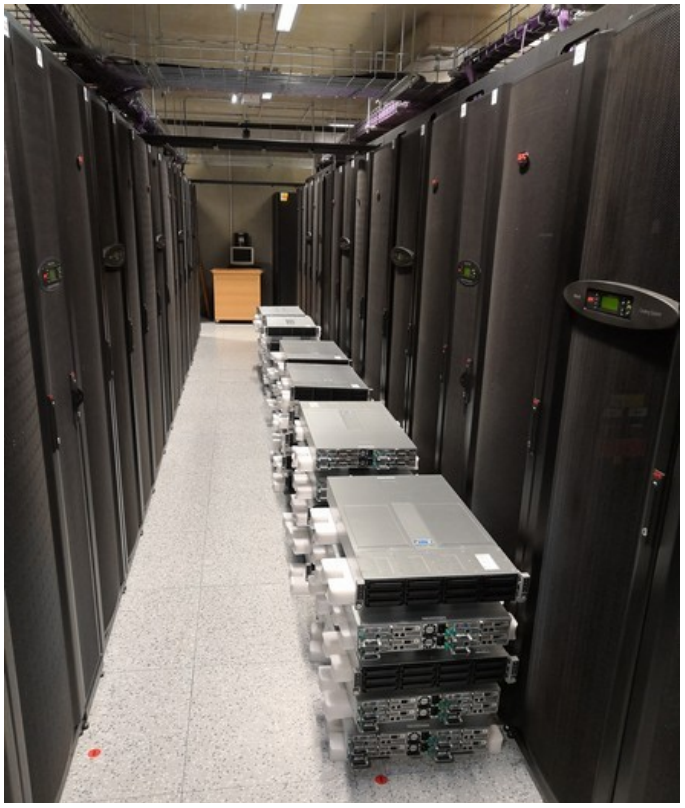


Emw at Wikimedia Commons
CC BY-SA

MD simulations literature

- Alen & Tildesley “Computer Simulations of Liquids”
- Frenkel & Smit “Understanding Molecular Simulation”
- H. Martinez-Seara, L. Cwiklik, “Introductory tutorial for GROMACS”
  DOI: 10.5281/zenodo.1230441
- ChatGTP

Computer simulations: Hardware



- Supercomputer or cluster:
  - Allows to simulate BIGGER, but not much “longer”?
- “Super”-desktop with powerful GPU and plenty of Memory:
  - to prepare, analyze and visualize simulations

Buy the proper hardware (HW)

- Different simulations has different HW needs.
  - Amount of memory
  - Number and type of cores
    - AMD: cheaper, faster, less issues?
  - GPUs: Nvidia (Cuda) vs AMD/Lumi.
  - Local disk SSD/NVME an backups (Please).
- Fight for the proper hardware.
  - Rarely, desktops targeted to MS Office are useful for simulations.

Computer Codes

- Molecular dynamics (MD) simulation
  - GROMACS (www.gromacs.org)
  - Amber (www.ambermd.org)
  - CHARMM (www.charmm.org)
  - NAMD (www.ks.uiuc.edu/Research/namd)
  - OpenMM (http://openmm.org/)
  - LAMMPS (https://lammps.sandia.gov/)
- Molecular visualization
  - VMD (https://www.ks.uiuc.edu/Research/vmd/)
  - -Pymol (https://pymol.org/)
- Analysis
  - Python, R, Matlab
  - MDAnalysis (https://www.mdanalysis.org/)
  - MDTraj (mdtraj.org)

- Optimal simulation and working environments for MD simulations:

Linux

python™

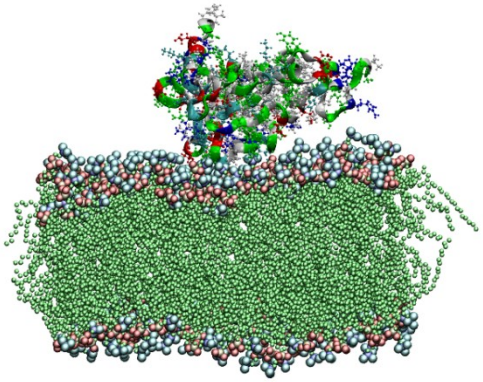
archlinux

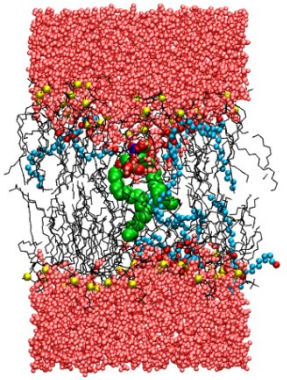
Never use windows !!

Pros and Cons of MD?

Advantages

Atomistic resolution!

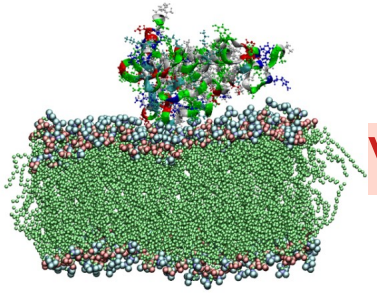




Disadvantages/limitations

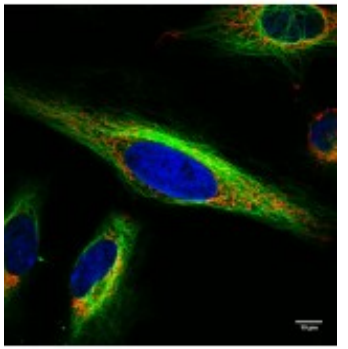
System size (~10^{-8} m)

~1 nm
(10^{-9} m)



Vs

~1 μm
(10^{-6} m)



8x5is at Wikimedia Commons
CC BY-SA

Timescale (~10^{-6} μs)

1 μs
(10^{-9} m)

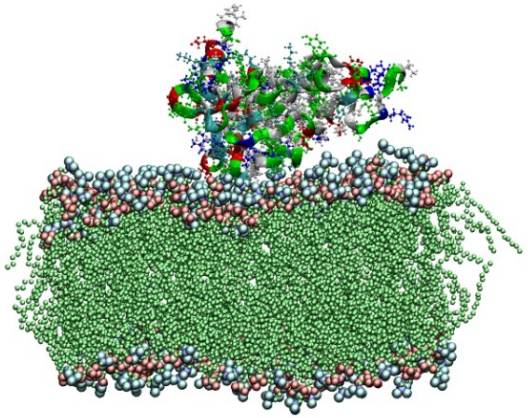
Vs

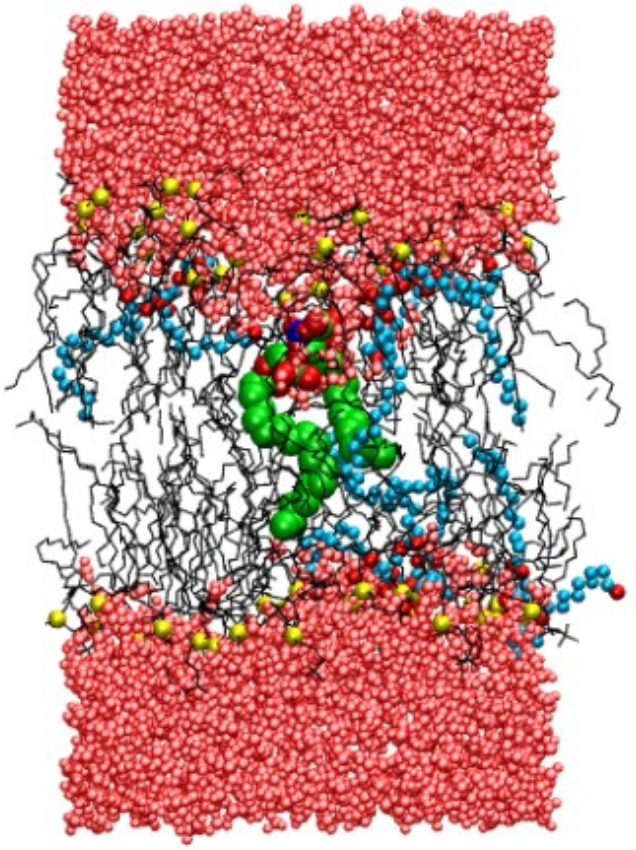
1 s
(10^{0} s)

Limited time- and length-scale
No chemical reactions!

Classical or Quantum physics

Atoms as quantum particles or billiard balls?





Do we need to consider quantum mechanics when dealing with molecules in biosystems?

Why we can use MD at all?

Born-Oppenheimer approximation

Born-Oppenheimer approximation:

- $M_e << M_{nucl}$ ⇒ nuclei move in an averaged field of electrons ⇒ motion of electrons can be separated from motion of nuclei

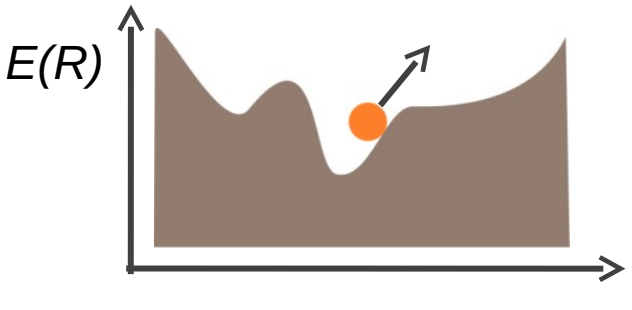
$$\Psi_{tot} = \psi_{el}(r;R)\psi_n(R)$$

for electrons:

$$H_{el}(r;R)\psi_{el}(r;R) = E_{el}(R)\,\psi_{el}(r;R)$$

for nuclei:

$$[T_n + E_{el}(R)]\psi_n(R) = \mathrm{E}\psi_n(R)$$



- Nuclei move on a potential energy surface $E_{el}(R)$
- System remains in one electronic state
- Fails: near-degeneracy

Classical approximation

- de Broglie wavelength: $\lambda = h/p$
- Thermal de Broglie wavelength:
  $\lambda_{th} = h/p$, where at given temperature:
  $p = \sqrt{2\pi m k_b T}$
  so: $\lambda_{th} = h/\sqrt{2\pi m k_b T}$

- **Classical mechanics may be used for nuclei**
  Thermal de Broglie wavelength (298K):
  H: ~10^{-10}m
  H_2: ~10^{-11}m
  He: ~10^{-11}m

MD: equations of motion, forces and potential

- In MD, classical Newton's equations of motion are solved for atoms in the considered system.
- Trajectory (positions of atoms in time) is the result of MD simulation. System properties calculated from the trajectory.
- Interaction potential must be assumed (**force field**).
- Initial conditions must be set (initial positions and velocities of all atoms).

$$V = -\int_{ref}^{r} \vec{F}\,\overrightarrow{dr}$$

$V(\vec{r})$ - force field

Gradient definition:

$$\nabla f = \frac{\partial f}{\partial x}\mathbf{i} + \frac{\partial f}{\partial y}\mathbf{j} + \frac{\partial f}{\partial z}\mathbf{k}$$

$$\overrightarrow{F_i} = -\overrightarrow{\nabla V}$$

$$\vec{a} \equiv \frac{d^2\overrightarrow{r_i}}{dt^2}$$

$$\overrightarrow{F_i} = m_i\overrightarrow{a_i}$$

Equations to be solved in MD:

$$-\frac{dV}{d\overrightarrow{r_i}} = m_i \frac{d^2\overrightarrow{r_i}}{dt^2}$$

Assuming the force field:

$V(\vec{r})$ - force field

$\overrightarrow{r_i}(t=0), \overrightarrow{v_i}(t=0)$

Ab initio MD - Interaction potential from quantum mechanics

Born-Oppenheimer approximation:

- $M_e << M_{nucl}$ ⇒ nuclei move in an averaged field of electrons ⇒ motion of electrons can be separated from motion of nuclei

$$\Psi_{tot} = \psi_{el}(r;R)\psi_n(R)$$

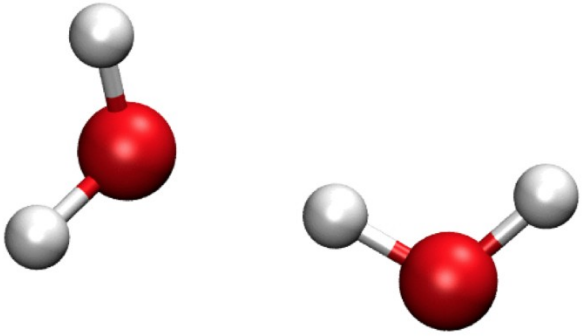
for electrons:

$$H_{el}(r;R)\psi_{el}(r;R) = E_{el}(R)\,\psi_{el}(r;R)$$

for nuclei:

$$[T_n + E_{el}(R)]\psi_n(R) = \mathrm{E}\psi_n(R)$$

Ab initio MD
(e.g., water dimer, picoseconds)



$V(\vec{r})$ – interaction potential, calculated on-the-fly (in each step) from the electronic wave function

$\vec{r_i}(t=0), \vec{v_i}(t=0)$

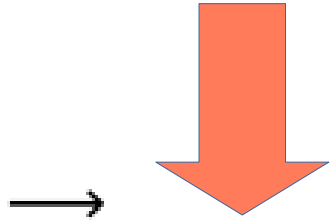
$$\vec{F_i} = m_i\vec{a_i}$$

$$-\frac{dV}{d\vec{r_i}} = m_i\frac{d^2\vec{r_i}}{dt^2}$$

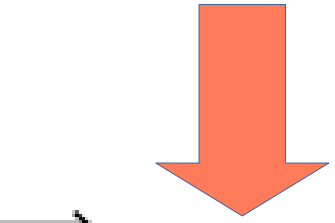
Limitations – computationally very expensive (short time and/or small systems only)!

Classical MD – The force field: The molecular model (empirical)

$V(\vec{r})$ - force field



$$\overrightarrow{F_i} = -\nabla V$$



$$\overrightarrow{F_i} = m_i \overrightarrow{a_i}$$

The force field (V) and the environment (r) determine the force that an atom feels

The force experienced by an atom and its state determine how an atom moves

The classical MD force field

$$V_i(r_1, r_2, .., r_N) = V_{bond} + V_{angle} + V_{dihedral} + V_{LJ} + V_{coulomb}$$

- The force field determines the force that an atom feels.
- The force field is “invariable” during a simulation.
- The force fields are usually additive.
- The force field is the collection of functions and its parameters used to compute the final force experienced by an atom.
- There are many kind of force fields to simulate biosystems, most of the use similar functional forms but different parameters
  - Is there the **ultimate** force field?

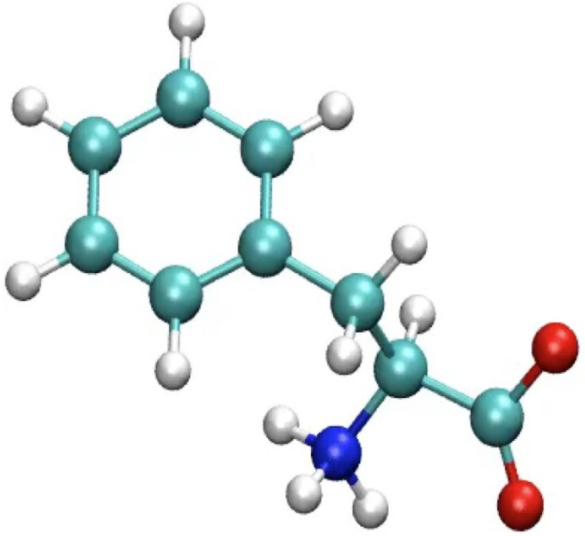
Finding an approximate interaction potential

$V(\vec{r})$ - force field

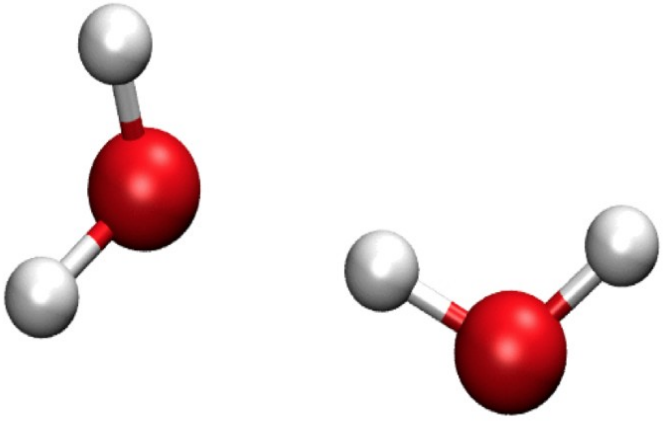
- Force field – derived by fitting to experimental and/or ab initio data
- Force fields are called “empirical potentials”
- Force fields are not ‘purely’ empirical, as ab initio data are also used for their derivation!
- Often they contain a lot of magic too :(

Force field: intra- and intermolecular interactions

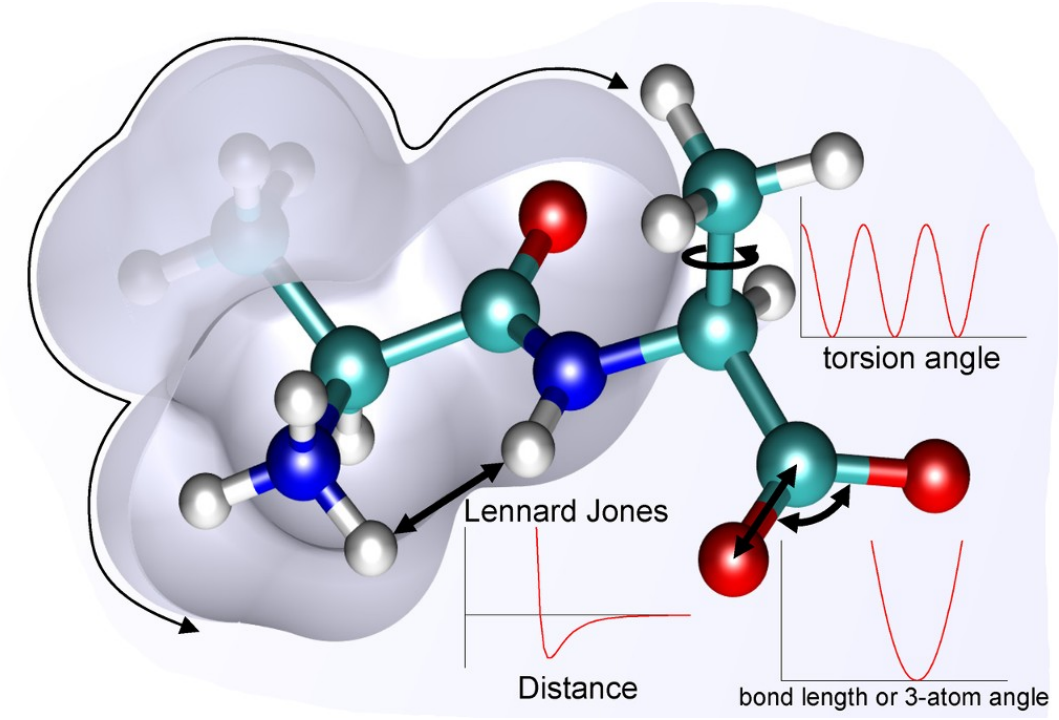
Intramolecular interactions
(*chemical*, bonding)



Intermolecular interactions
(*physical*, non-bonding)

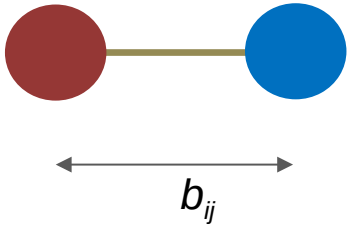


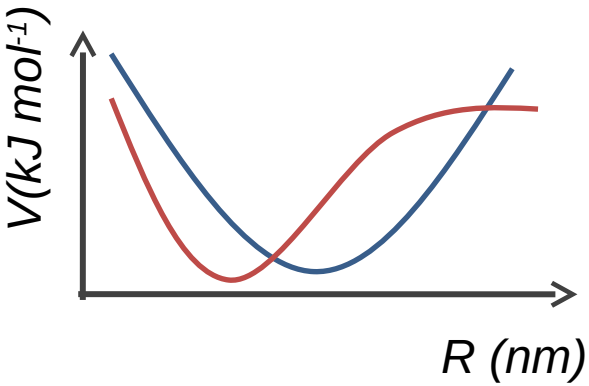
Force field: Intramolecular interactions



- Intramolecular interaction can be approximated by the additive terms:
  - Bonds
  - Valence angles
  - Torsion (dihedral) angles
  - Intramolecular electrostatics
  - Intramolecular van der Waals

Intramolecular interactions – bond potential



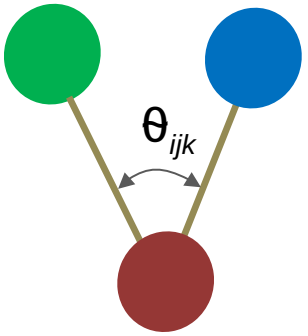


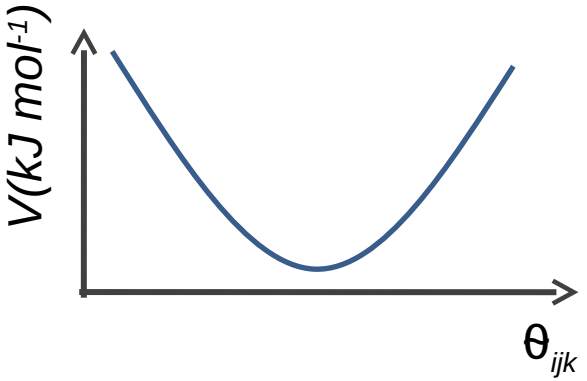
- Bond potential usually approximated as:
  - Harmonic potential $V_b\left(r_{ij}\right) = \frac{1}{2} k_{ij}^b (r_{ij} - b_{ij})^2$
  - Morse potential $V_{morse}(r_{ij}) = D_{ij}[1 - \exp(-\beta_{ij}(r_{ij} - b_{ij}))]^2$

Errors due to differences between quantum and classical oscillators!

IN PRACTICE: it is better to constraint the bonds (it also allows larger Δt)

Intramolecular interactions – valence angle potential





- Valence angle is usually approximated as:
  - Harmonic angle potential

$$V_a(\theta_{ijk}) = \frac{1}{2}k^{\theta}_{ijk}(\theta_{ijk} - \theta^0_{ijk})^2$$

Sometimes also as:

  - Cosine-based potential (in GROMOS)

$$V_a(\theta_{ijk}) = \frac{1}{2}k^{\theta}_{ijk}\left(\cos(\theta_{ijk}) - \cos(\theta^0_{ijk})\right)^2$$

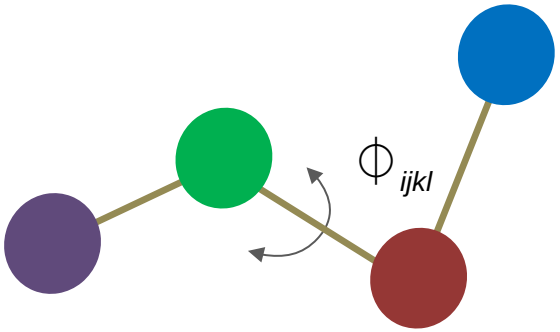
  - Restricted bending potential

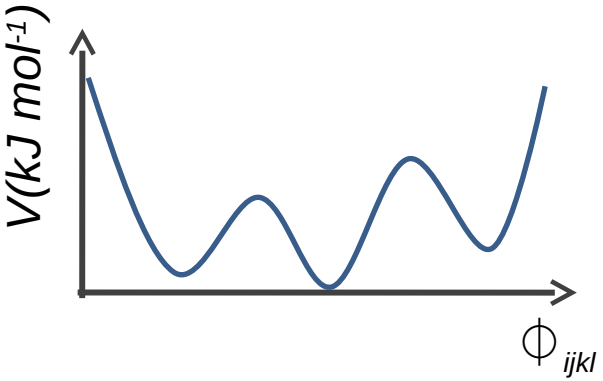
$$V_{\mathrm{ReB}}(\theta_i) = \frac{1}{2}k_{\theta}\frac{(\cos\theta_i - \cos\theta_0)^2}{\sin^2\theta_i}$$

  - Urey-Bradley potential (in CHARMM)

$$V_a(\theta_{ijk}) = \frac{1}{2}k^{\theta}_{ijk}(\theta_{ijk} - \theta^0_{ijk})^2 + \frac{1}{2}k^{UB}_{ijk}(r_{ik} - r^0_{ik})^2$$

Intramolecular interactions – torsion angle potential





- Different (equivalent) functional forms:
  - Periodic (Fourier, proper)

$$V_d(\phi_{ijkl}) = \sum_{n=0}^{5} k_{\phi_n}(1 + \cos(n\phi_n - \phi_{sn})$$

k_{ϕ_n} - dihedral force constant

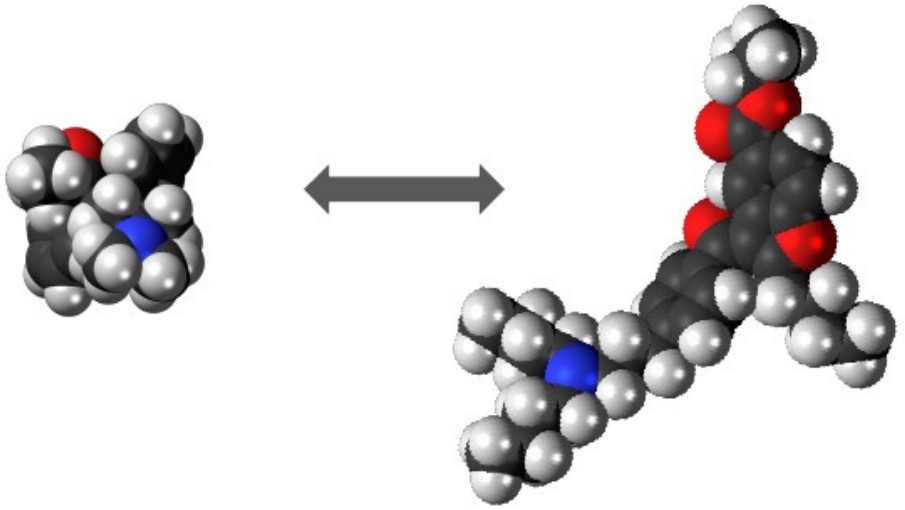
n – number of minimum

ϕ_{sn} - angular offset
(usually 0° for odd n, 180° for even n)

  - Ryckaert-Bellemans potential

$$V_{rb}(\phi_{ijkl}) = \sum_{n=0}^{5} C_n(\cos(\psi))^n$$

Intermolecular interactions



Intramolecular interactions classification:

- **Electrostatic interactions:** charge-charge, charge-dipole, charge-induced dipole (or higher multipoles)
- **van der Waals interactions** (all but the above, attractive & repulsive, between uncharged molecules):
  - dipole-dipole (or higher quadrupoles, sometimes classified as part of electrostatic interactions)
  - dipole-induced dipole (Debye force, 'polarization')
  - between two instantaneously induced dipoles (London dispersion force)
  - Pauli repulsion

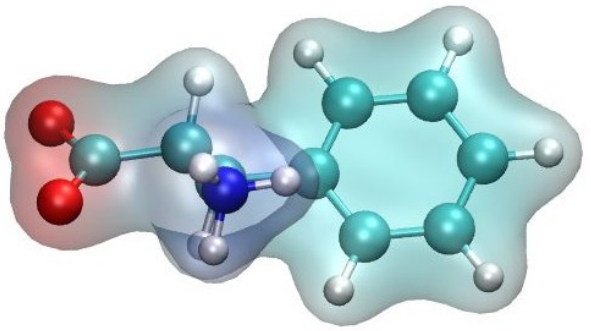
Electrostatic interactions – modeling via point charges

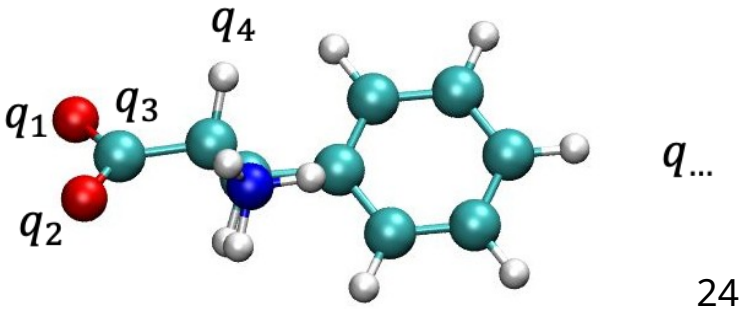
All interactions at the molecular level have their origin in electrostatics.

In non-polarizable classical MD simulation, electrostatic interactions in the force field are reduced (approximated) to **charge-charge interactions**!

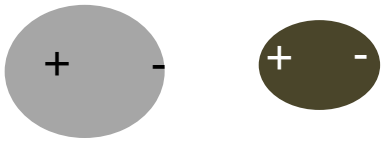
In force fields, in practice, this is realized by the use of **point charges**:

- We use Point charges to generate approximate electrostatic potential
- Atomic charges are not observables, i.e., their definition is arbitrary.
- Derived usually from quantum mechanics (different schemes exist: ESP, RESP, etc.)

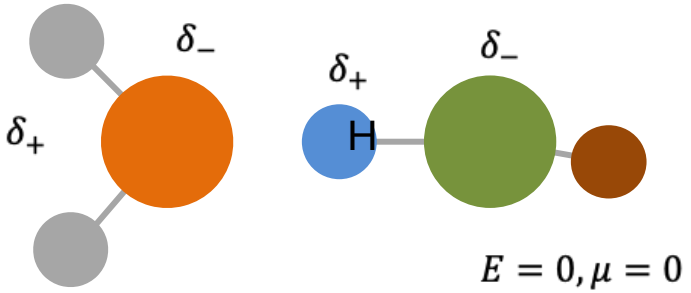




Hydrogen bond & electrostatics



$E \neq 0, \mu \neq 0$



$E = 0, \mu = 0$

Hydrogen bond – not longer explicitly included in typical force fields

Attractive intermolecular interactions:
A fair approximation?

Interaction	V(r) dependence	Typical energy (kJ/mol)
ion-ion	$1/r$	250
ion-dipole	$1/r^2$	15
dipole-dipole	$1/r^3$	2
dipole-dipole (rotating)	$1/r^6$	0.6
London (dispersion)	$1/r^6$	2
hydrogen bond		20

based on Atkins, Phys Chem

The **always** attractive van der Waals iteractions in condensed phases can be **approximated as**:

$$V_{ij} = -\frac{C_6}{r^6}$$

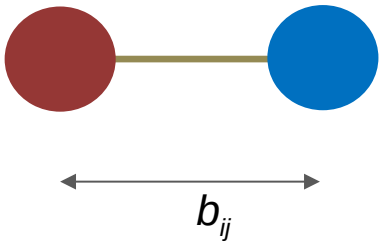
Pauli repulsion

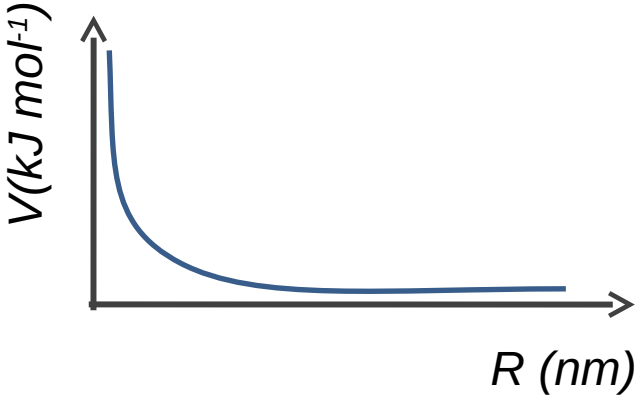
Short-range strong repulsive forces due to overlap of electronic densities

$$V_{ij} = \frac{C_{12}}{r^{12}}$$

(Atoms don't fuse easily, do they?)

Origin:
Quantum mechanical exchange energy between fermions

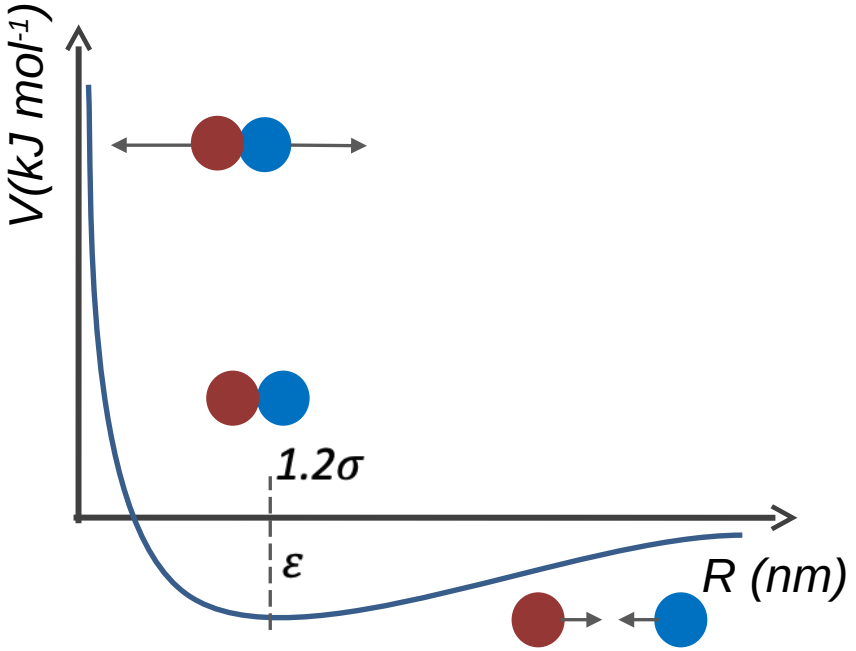




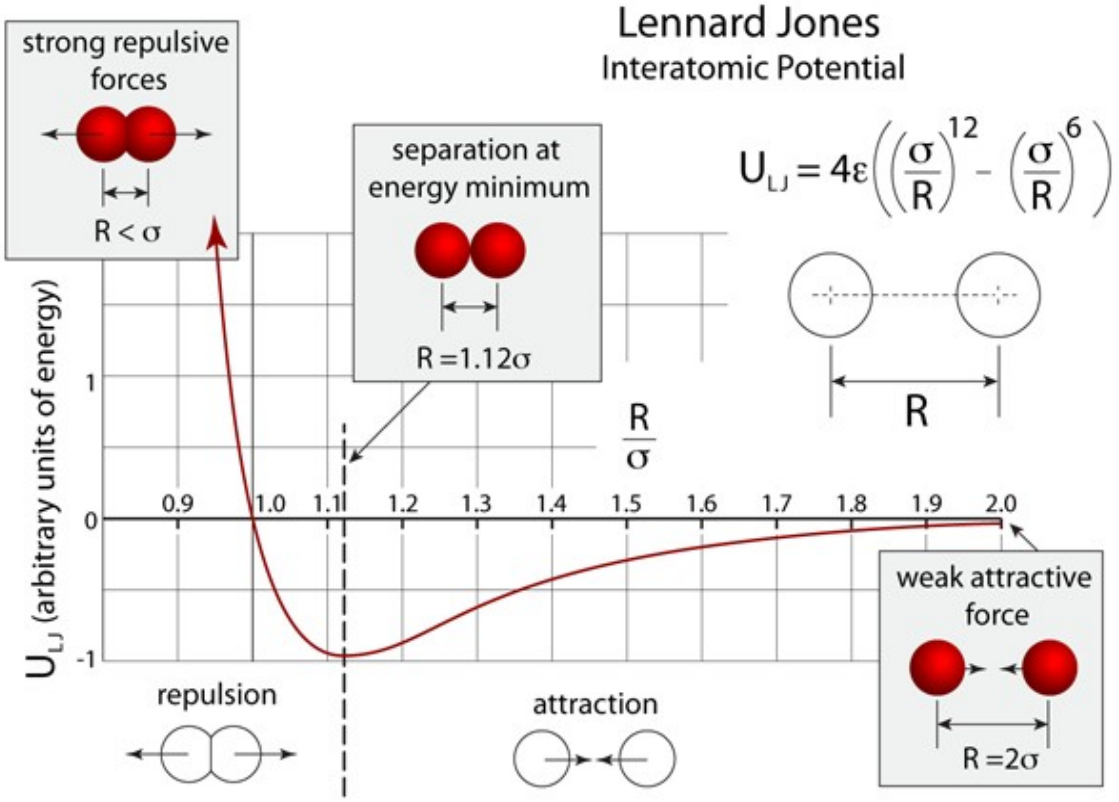
Lennard-Jones potential

An approximate model for the isotropic part of a total (repulsion plus attraction) van der Waals force as a function of distance

$$V_{LJ}(\boldsymbol{r}_{ij}) = 4\epsilon_{ij}\left(\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right) = V_{LJ}(r_{ij}) = \frac{C_{ij}^{(12)}}{r_{ij}^{12}} - \frac{C_{ij}^{(6)}}{r_{ij}^{6}}$$



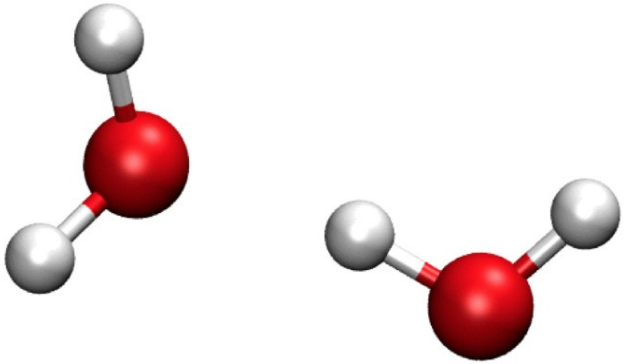
Understanding the Lennard-Jones potential



The graph above plots the Lennard–Jones potential function, and indicates **regions of attraction and repulsion**. Atoms try to minimize their potential energy and at the lowest temperatures are sitting at the bottom of the potential curve. When the atomic separations are to the left of the minimum the atoms repel, otherwise they attract one another.

http://atomsinmotion.com/book/chapter5/md

Lennard-Jones potential – mixing rules



$$V_{LJ}(\boldsymbol{r}_{ij}) = 4\epsilon_{ij}\left(\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right)$$

atomic (not pair) property

Geometric average:

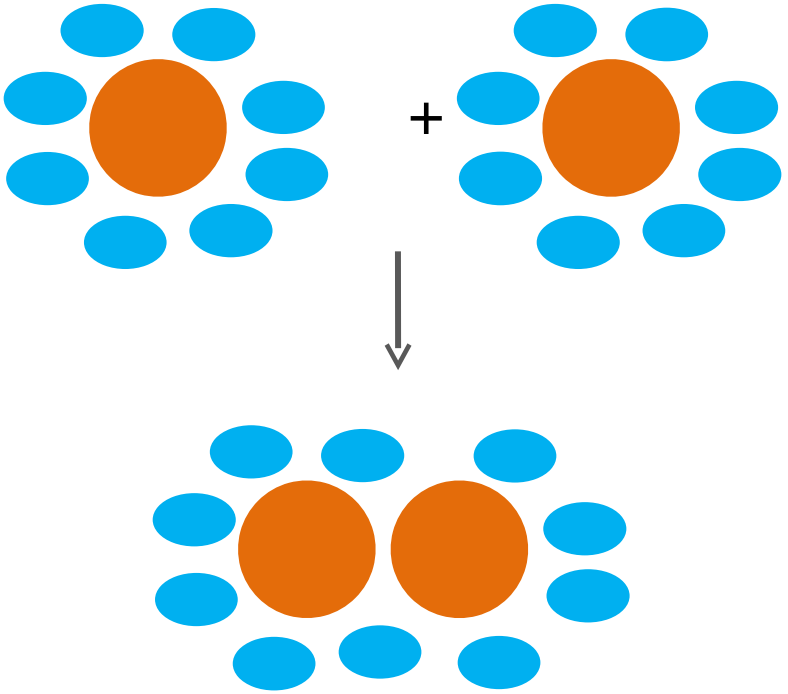
$$C_{ij}^{(6)} = \left(C_{ii}^{(6)}\, C_{jj}^{(6)}\right)^{1/2}$$
$$C_{ij}^{(12)} = \left(C_{ii}^{(12)}\, C_{jj}^{(12)}\right)^{1/2}$$

Lorentz-Berthelot rules:

$$\sigma_{ij} = \tfrac{1}{2}(\sigma_{ii} + \sigma_{jj})$$
$$\epsilon_{ij} = (\epsilon_{ii}\, \epsilon_{jj})^{1/2}$$

Hydrophobic effect

The apparent tendency of nonpolar molecules in a polar solvent (e.g., in water) to interact with one another



- Hydrophobic effect results from electrostatic and van der Waals interactions
- The term ‘hydrophobic interaction’ is disputable

An empirical force field – overall potential energy scheme

$V_{total} = E_{bonded} + E_{nonbonded}$

$E_{vdW} = E_{dip\text{-}dip} + E_{dip\text{-}ind\ dip} + E_{ind\ dip\text{-}ind\ dip} + E_{charge\text{-}ind.dip}$

$E_{bonded} = E_{bond} + E_{angle} + E_{dihedral}$

$E_{nonbonded} = E_{electrostatic} + E_{van\ der\ Waals} + E_{Pauli}$

$$V_b(r_{ij}) = \frac{1}{2}k_{ij}^b(r_{ij} - b_{ij})^2$$

$$V_a(\theta_{ijk}) = \frac{1}{2}k_{ijk}^\theta(\theta_{ijk} - \theta_{ijk}^0)^2$$

$$V_{rb}(\phi_{ijkl}) = \sum_{n=0}^{5} C_n(\cos(\psi))^n$$

$E_{electrostatic} = E_{charge\text{-}charge} + E_{charge\text{-}dip} + E_{charge\text{-}induced\ dip.}$

$E_{Coulomb\ pointcharge\text{-}pointcharge}$

$$V_{Coulomb\ ij} = \frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}}$$

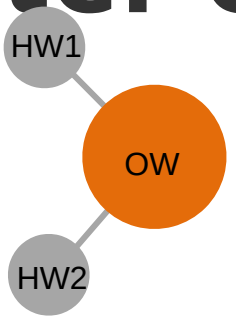
$$U = \sum_{bonds} \frac{k_i^b}{2}(\mathbf{r_i} - \mathbf{r_i^{eq}})^2 + \sum_{angles} \frac{k_i^\alpha}{2}(\theta_i - \theta_i^{eq})^2 + \sum_{torsions} \sum_n k_{\phi,n}\left(1 + \cos\left(n\phi - \phi^{ref}\right)\right) + \sum_{pairs} \left(\left\{ \frac{C_{ij}^{(12)}}{r_{ij}^{12}} - \frac{C_{ij}^{(6)}}{r_{ij}^6} \right\}_{vdw} + \left\{ k_e \frac{q_i q_j}{r_{ij}} \right\}_{coul} \right)$$

$$-\frac{dV_{total}}{d\vec{r_i}} = m_i \frac{d^2\vec{r_i}}{dt^2}$$

$\sim E_{Lennard\text{-}Jones}$

$$V_{L-J\ ij} = \frac{C_{ij}^{(12)}}{r_{ij}^{(12)}} - \frac{C_{ij}^{(6)}}{r_{ij}^{(6)}}$$

Force field in Gromacs – TIP3P water example



Charges & intramolecular terms
(tip3p.itp file)

```
[ 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      -0.834     16.00000
  2   HW          1       SOL       HW1      1       0.417      1.00800
  3   HW          1       SOL       HW2      1       0.417      1.00800

#ifndef FLEXIBLE

[ settles ]
; OW    funct   doh dhh
1       1       0.09572 0.15139

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

#else

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


[ angles ]
; i     j       k       funct   angle   force_constant
2       1       3       1       104.52  628.02      104.52  628.02
```

Intermolecular terms (Lennard-Jones)
(ffnonbonded.itp file)

```
[ atomtypes ]
; name      at.num   mass      charge  ptype   sigma       epsilon
Br          35       79.90     0.0000  A   0.00000e+00  0.00000e+00
C            6       12.01     0.0000  A   3.39967e-01  3.59824e-01
CA           6       12.01     0.0000  A   3.39967e-01  3.59824e-01
CB           6       12.01     0.0000  A   3.39967e-01  3.59824e-01
CC           6       12.01     0.0000  A   3.39967e-01  3.59824e-01
CK           6       12.01     0.0000  A   3.39967e-01  3.59824e-01
CM           6       12.01     0.0000  A   3.39967e-01  3.59824e-01
HW           1        1.008    0.0000  A   0.00000e+00  0.00000e+00
OW           8       16.00     0.0000  A   3.15061e-01  6.36386e-01
OH           8       16.00     0.0000  A   3.06647e-01  8.80314e-01
OS           8       16.00     0.0000  A   3.00001e-01  7.11280e-01
O2           8       16.00     0.0000  A   2.95992e-01  8.78640e-01
```

Mixing rules
(forcefield.itp file)

```
#define _FF_AMBER
#define _FF_AMBER99SBILDN

[ defaults ]
; nbfunc        comb-rule       gen-pairs       fudgeLJ fudgeQQ
1               2               yes             0.5     0.8333
```

Force fields – Remarks

- Based on a) experimental data (spectroscopy, density, phase transitions), b) ab initio methods, and c) trial and error.
- fast
- often system-specific (e.g., good for proteins, bad for lipids)
- there are interaction models (functional form is assumed)
- polarizable force fields sometimes used (more and more often but have their own problems)
- no chemical reactivity, problems with radicals, excited states, weakly-interacting complexes
- implicit water models, many-body effects, anharmonicity
- problematic under non-typical biological conditions (e.g., low temperature)
- **steep learning curve, experience needed to use them properly**

Main force fields use in biosystems

- AMBER
- CHARMM
- OPLS
- GROMOS
- Glycan
- MARTINI (coarse grained)

Solution of equations of motion

$$m_i \frac{d^2 \mathbf{r_i}}{dt^2} = \mathbf{F_i} = -\nabla_{\mathbf{r_i}} U(\mathbf{r_1}, \mathbf{r_2}, \ldots, \mathbf{r_N})$$

Numerical integrators equations of motion

$$m_i \frac{d^2 \mathbf{r_i}}{dt^2} = \mathbf{F_i} = -\nabla_{\mathbf{r_i}} U(\mathbf{r_1}, \mathbf{r_2}, \ldots, \mathbf{r_N})$$

- in practical systems, too many bodies for analytical solution of equations of motion
- numerical integration of Newton's equations required
  - numerical integrators are used to approximate analytical solution
- numerical integrators suffer/may suffer of various problems:
  - low accuracy
  - time irreversibility
  - numerical instability
  - low computational efficiency
  - ...

- typical simulation times of biosystems: 100 ns - 1 μs
- note: 1 fs = 10^{-15} s
- if Δt =2 fs, then 1 μs requires 500,000,000 MD steps!

Taylor's expansion – discretization of Equations of Motion (EOMs)

$$\dot{r}_i(t) \qquad \frac{f_i(t)}{m} = a_i(t) = \ddot{r}_i(t)$$

$$\mathbf{r}_i(t+\Delta t) = \mathbf{r}_i(t) + \Delta t\ \mathbf{v}_i(t) + \frac{\Delta t^2}{2m_i}\ \mathbf{f}_i(t) + \frac{\Delta t^3}{3!}\ \dddot{\mathbf{r}}_i(t) + \mathcal{O}(\Delta t^4)$$

$$\mathbf{v}_i(t+\Delta t) = \mathbf{v}_i(t) + \frac{\Delta t}{m_i}\ \mathbf{f}_i(t) + \frac{\Delta t^2}{2}\ \ddot{\mathbf{v}}_i(t) + \frac{\Delta t^3}{3!}\ \dddot{\mathbf{v}}_i(t) + \mathcal{O}(\Delta t^4)$$

Euler algorithm

(forward difference approximation)

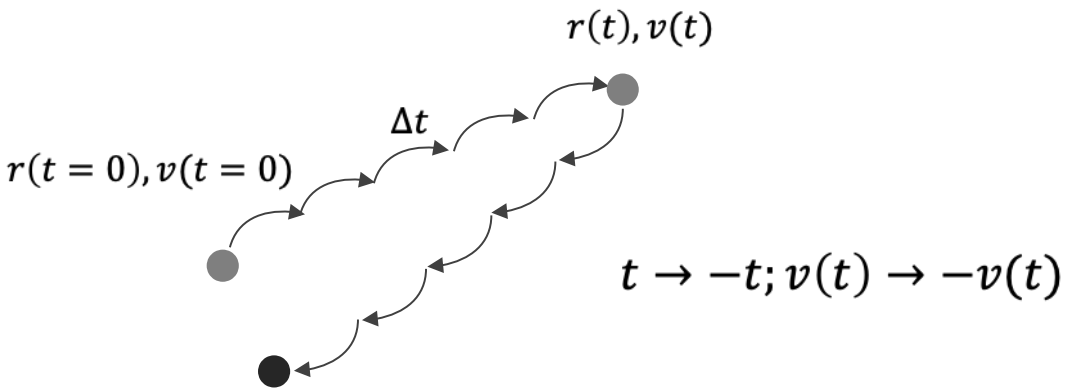
$$\mathbf{r}_i(t+\Delta t) = \mathbf{r}_i(t) + \Delta t\ \mathbf{v}_i(t) + \frac{\Delta t^2}{2m_i}\mathbf{f}_i(t) + \mathcal{O}(\Delta t^3)$$

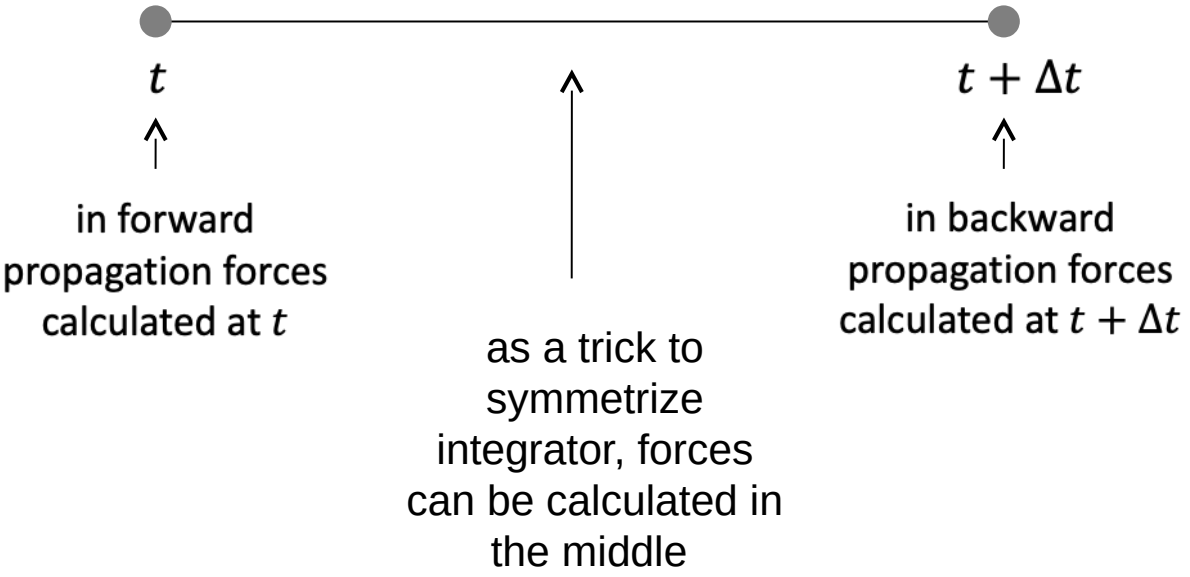
$$\mathbf{v}_i(t+\Delta t) = \mathbf{v}_i(t) + \frac{\Delta t}{m_i}\ \mathbf{f}_i(t) + \mathcal{O}(\Delta t^2)$$

NOT USED in MD:

- time irreversible (but Newton's equations are reversible)
- no phase-space preserving (Liouville's theorem violated)

Time irreversibility of integrators





Verlet algorithm

Derivation:

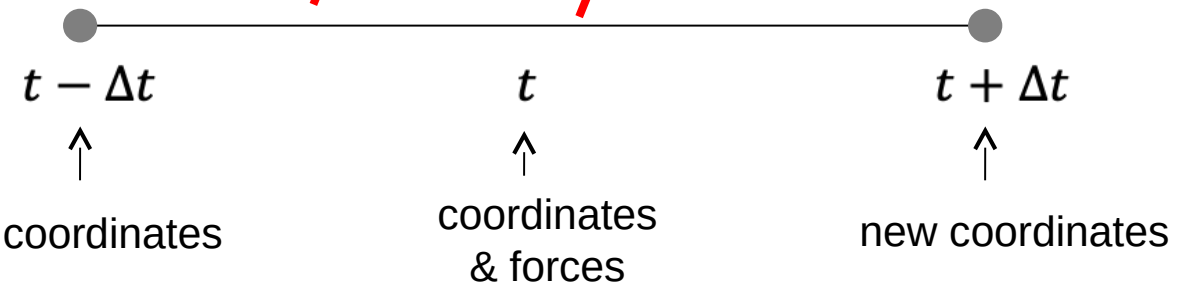
$$\mathbf{r}_i(t+\Delta t) = \mathbf{r}_i(t) + \Delta t\ \mathbf{v}_i(t) + \frac{\Delta t^2}{2m_i}\ \mathbf{f}_i(t) + \frac{\Delta t^3}{3!}\ \dddot{\mathbf{r}}_i(t) + \mathcal{O}(\Delta t^4)$$

$$\mathbf{r}_i(t-\Delta t) = \mathbf{r}_i(t) - \Delta t\ \mathbf{v}_i(t) + \frac{\Delta t^2}{2m_i}\ \mathbf{f}_i(t) - \frac{\Delta t^3}{3!}\ \dddot{\mathbf{r}}_i(t) + \mathcal{O}(\Delta t^4)$$

Positions from adding and velocities from subtracting the above eqs:

$$\mathbf{r}_i(t+\Delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t-\Delta t) + \frac{\Delta t^2}{m_i}\mathbf{f}_i(t) + \mathcal{O}(\Delta t^4)$$

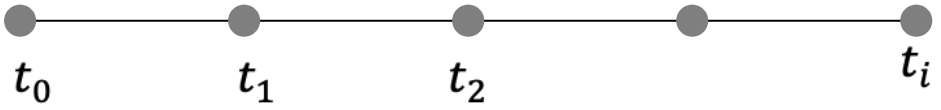
$$\mathbf{v}_i(t) = \frac{\mathbf{r}_i(t+\Delta t) - \mathbf{r}_i(t-\Delta t)}{2\Delta t} + \mathcal{O}(\Delta t^3).$$



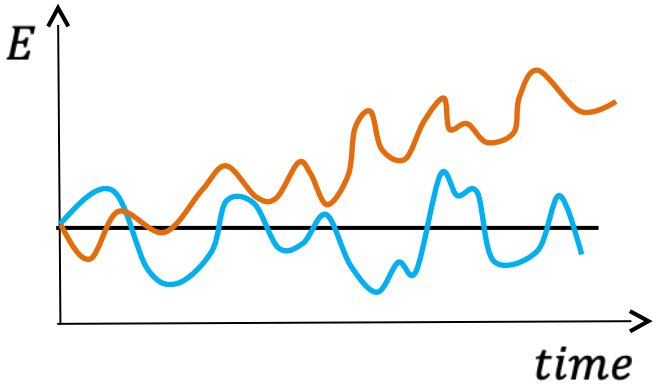
Verlet alg. is time reversible (symmetric when $t \to -t$)

Integrators - stability

Stability can be checked by conserved quantities: for instance, total energy (in NVE ensembles) can be used



ideally: $E_0 = E_1 = E_2 = E_i = const$

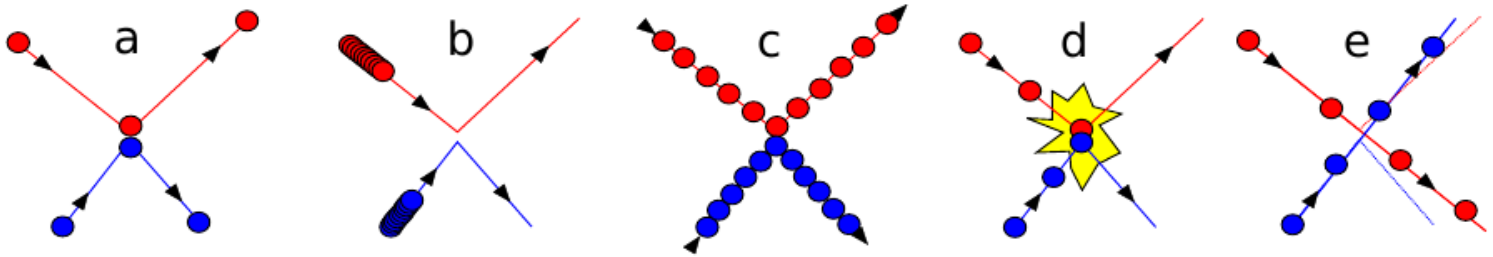


diverging – wrong!

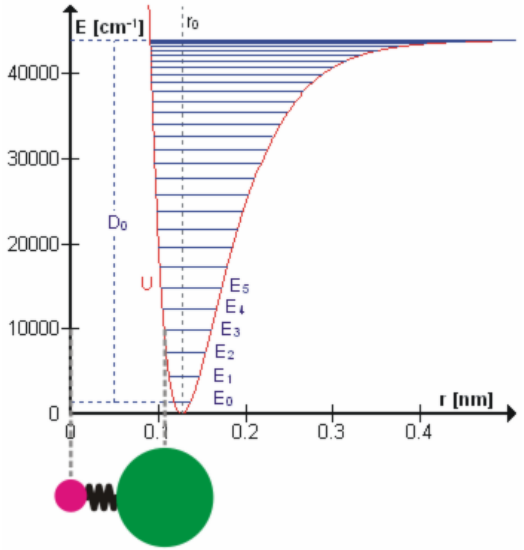
fluctuating – O.K.

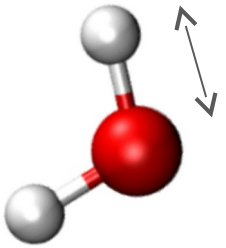
- integrators are conditionally stable
- conserved quantities diverge if Δt too large!

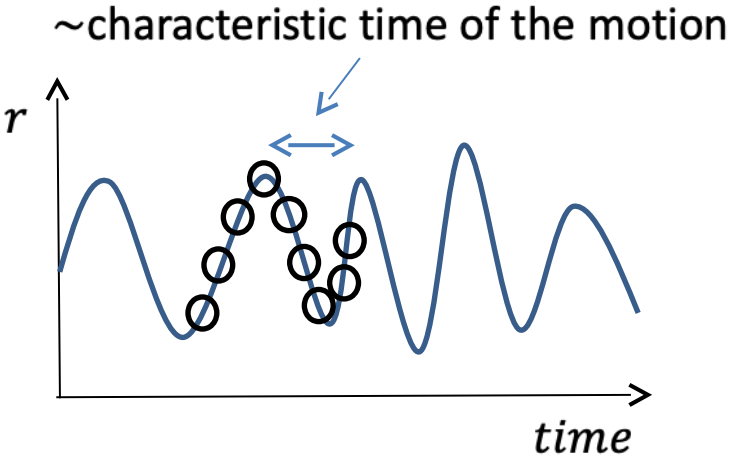
How to choose time step?



In practice, the time-step must be at least one order of magnitude shorter than a characteristic time (e.g., period) of the fastest motion in the system





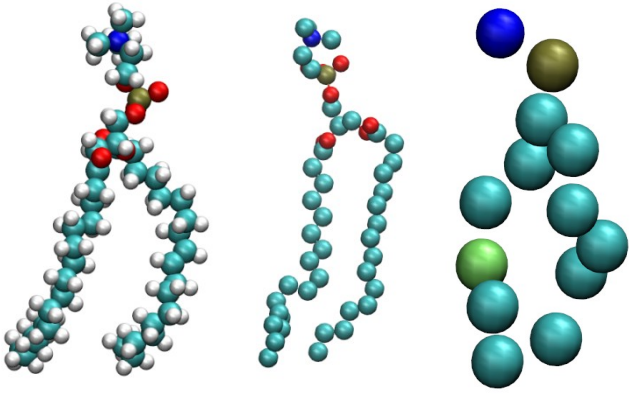


the fastest motion:

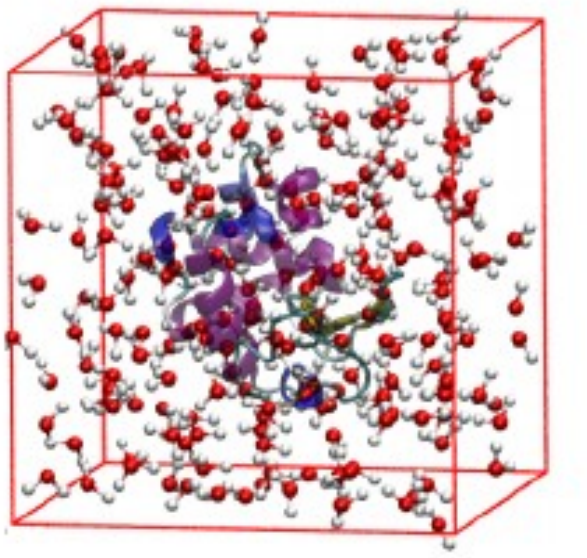
- typically - vibrations of O-H and C-H bonds with time scale of $\sim 10^{-14}$ s
- hence, typical time-step in MD Δt =1-5 fs

Time step - tricks

- most often, O-H and C-H bonds can be constrained (SHAKE algorithm for water, LINCS for C-H), then $\Delta t = 2$ fs
- united-atom force fields typically do not include C-H bonds, if water constrained then $\Delta t = 2$ fs
- coarse-grain force fields do not include atomistic motions $\Delta t = 20$ fs!

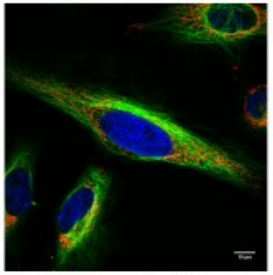


Simulation box



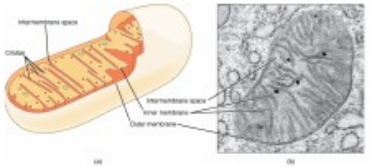
How big are biosystems of interest?

Cell
1-20 μm
($\sim 10^{-6}$ -10^{-5}m)



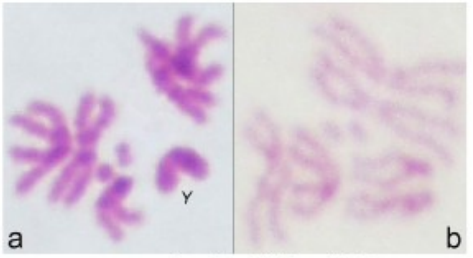
8x5is at Wikimedia Commons
CC BY-SA

Mitochondria
0.2-2 μm
($\sim 10^{-6}$ m)



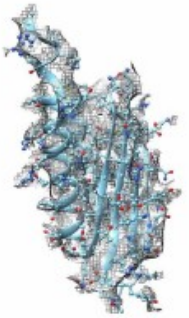
OpenStax at Wikimedia Commons
CC

Metaphase chromosome
0.2-20 μm
($\sim 10^{-6}$ -10^{-5}m)



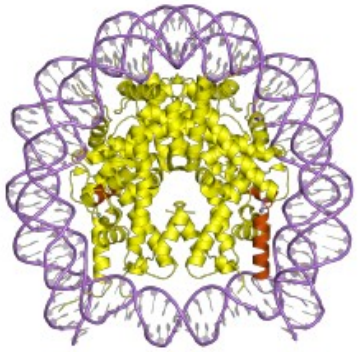
albval at Wikimedia Commons
CC

Ribosome fragment
20 nm
($\sim 10^{-8}$ m)

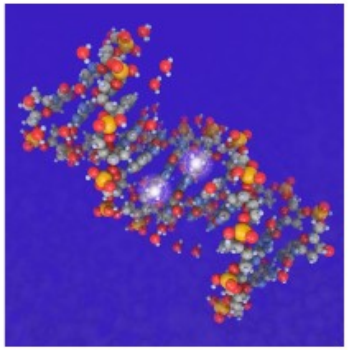


Franken et al. at Wikimedia Commons
CC

Nucleosome
10 nm
($\sim 10^{-8}$ m)

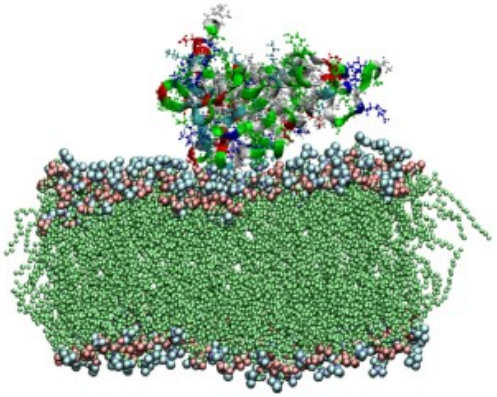


DNA fragment
10 nm
($\sim 10^{-8}$ m)

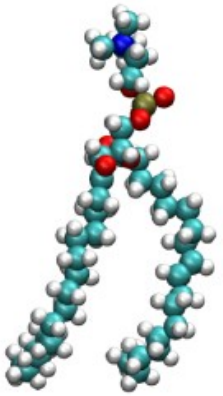


Qubeat at Wikimedia Commons
CC

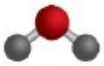
Fragment of cell membrane
with a protein
10 nm ($\sim 10^{-8}$ m)



Lipid molecule
2 nm
($\sim 10^{-9}$ m)

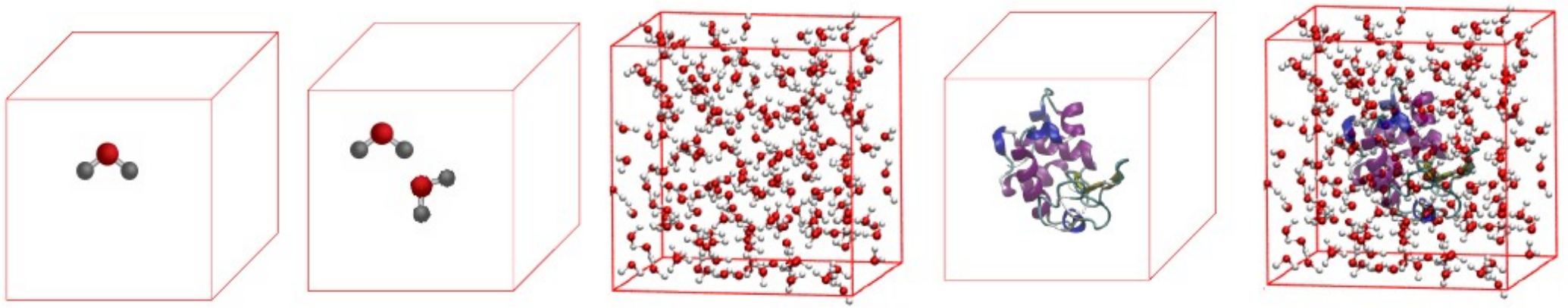


Water molecule
0.2 nm
($\sim 10^{-10}$ m)



Simulation box - limitations

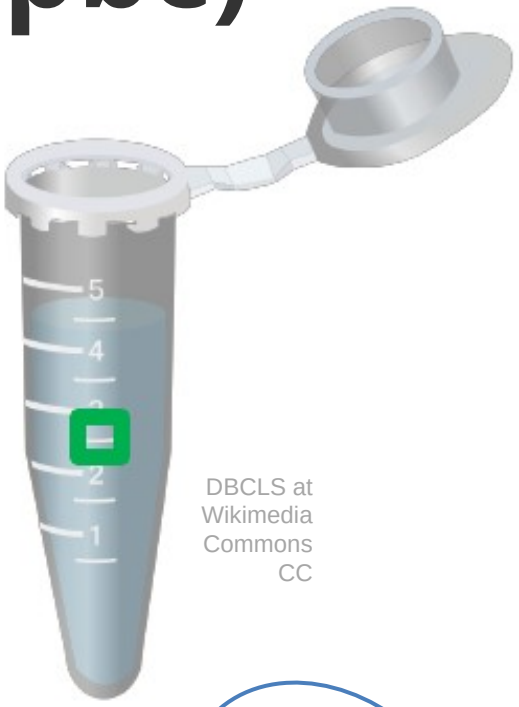
Simulation box – part of the physical space that is included in a simulation



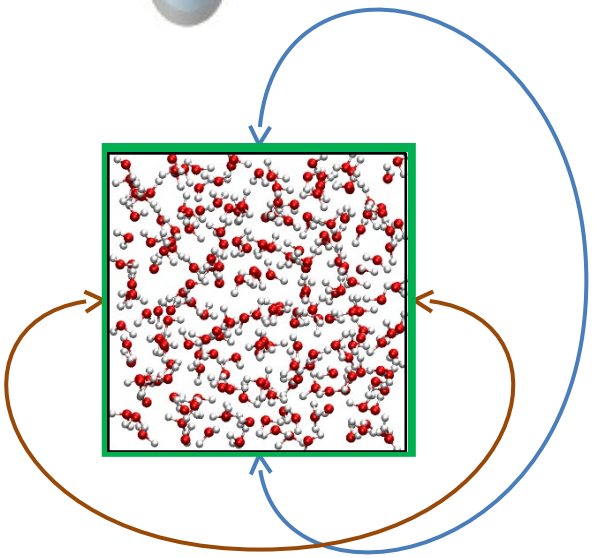
- size of simulated systems limited by computational costs
- typically 10^5-10^6 atoms; 100-1000 nm^3
- O.K. for small isolated molecules or small molecular clusters
- not proper for big molecules (e.g., big proteins) and condensed phases (e.g., small proteins in water, lipid membranes)

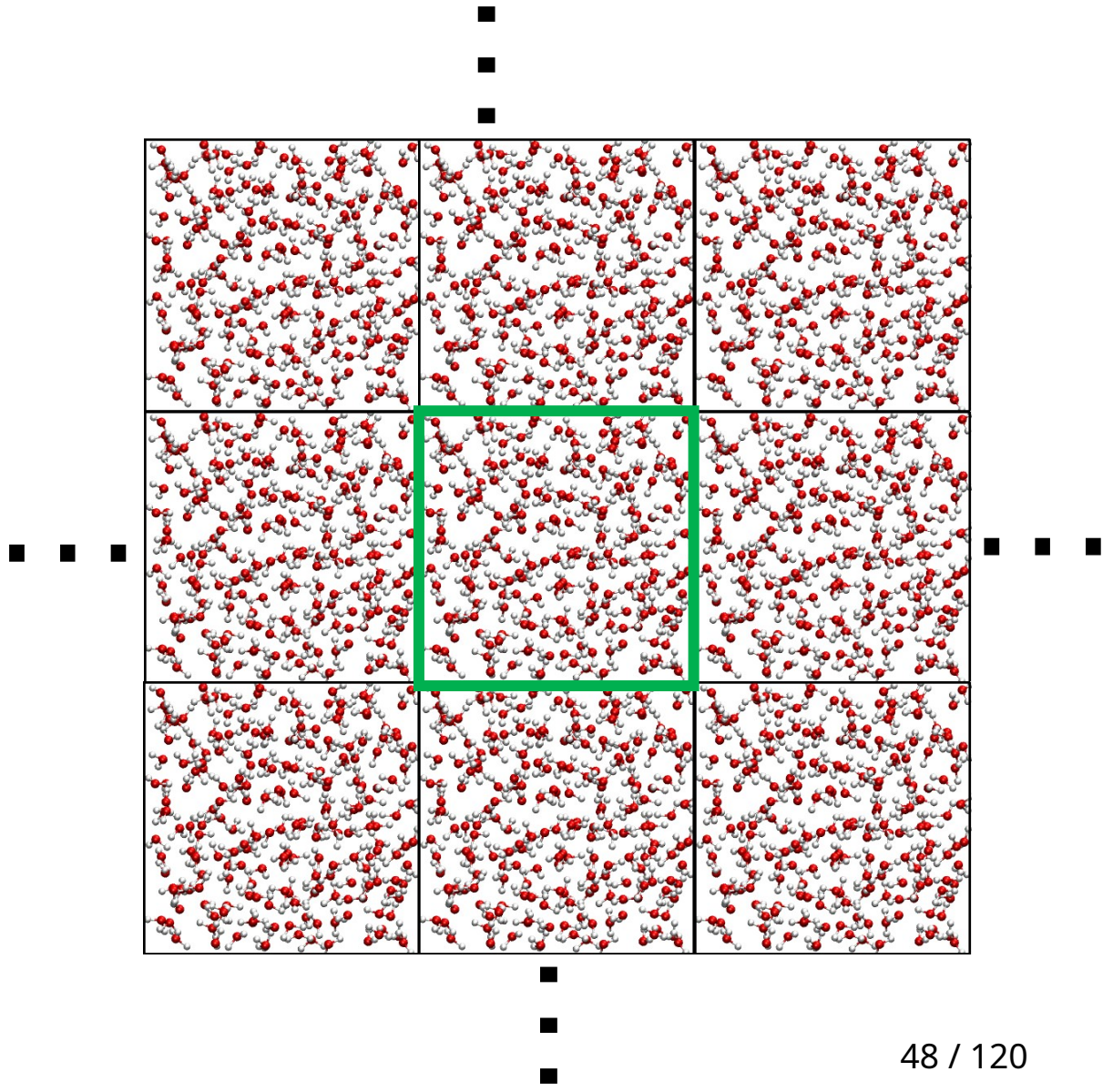
Solutions for condensed phases:
periodic boundary conditions!

Periodic Boundary Conditions (pbc)



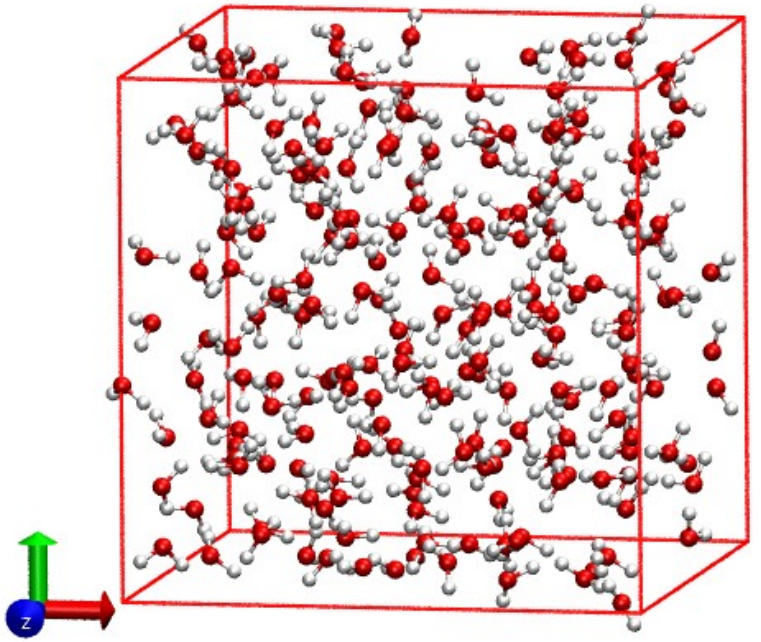
DBCLS at Wikimedia Commons CC





Periodic Boundary Conditions

- the simulated system is enclosed in so-called simulation box
- the box replicated in 3D (or 2D if needed)
- the finite system becomes infinite (but periodic!)



Benefits:

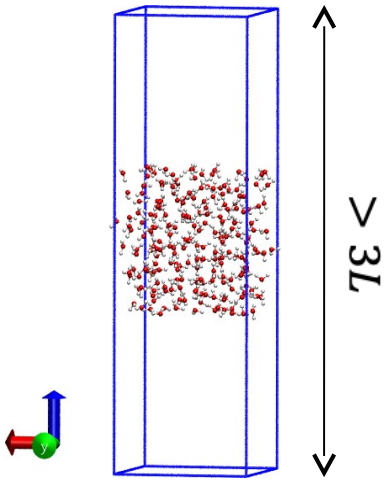
- artificial surfaces eliminated
- bulk-like behavior
- computational cost increases only slightly
- very good for periodic solid phases (crystals)
- reasonable for liquid phases

Disadvantages:

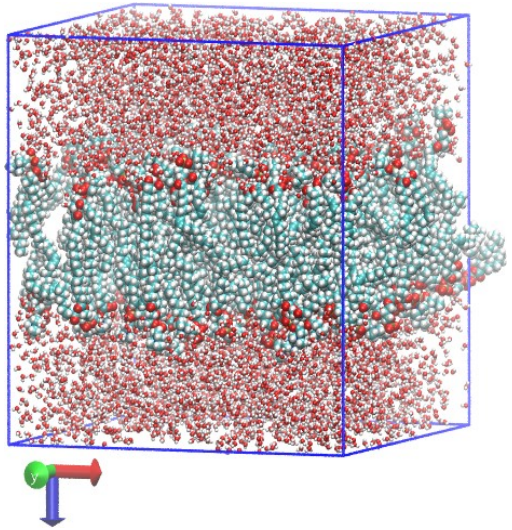
- artificial periodicity introduced (some interactions can display artifacts, some processes may be artificially induced)
- fails for effects with characteristic length > box size (phonons, membrane undulations, phase transitions-related phenomena)
- box has to be large enough (check!)

PBC special geometries

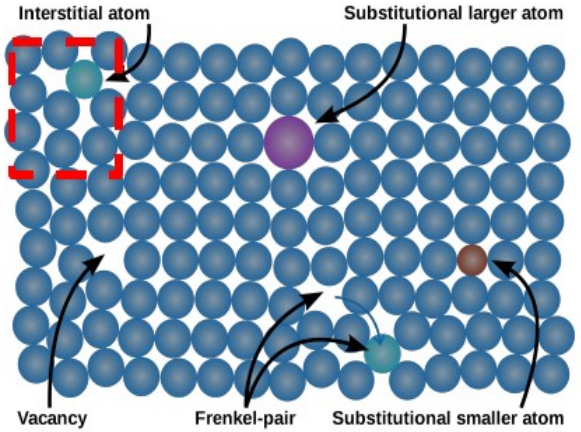
water-air interface
(slab geometry,
good for interfaces, 2x)



Lipid bilayer
(„slab-like”)



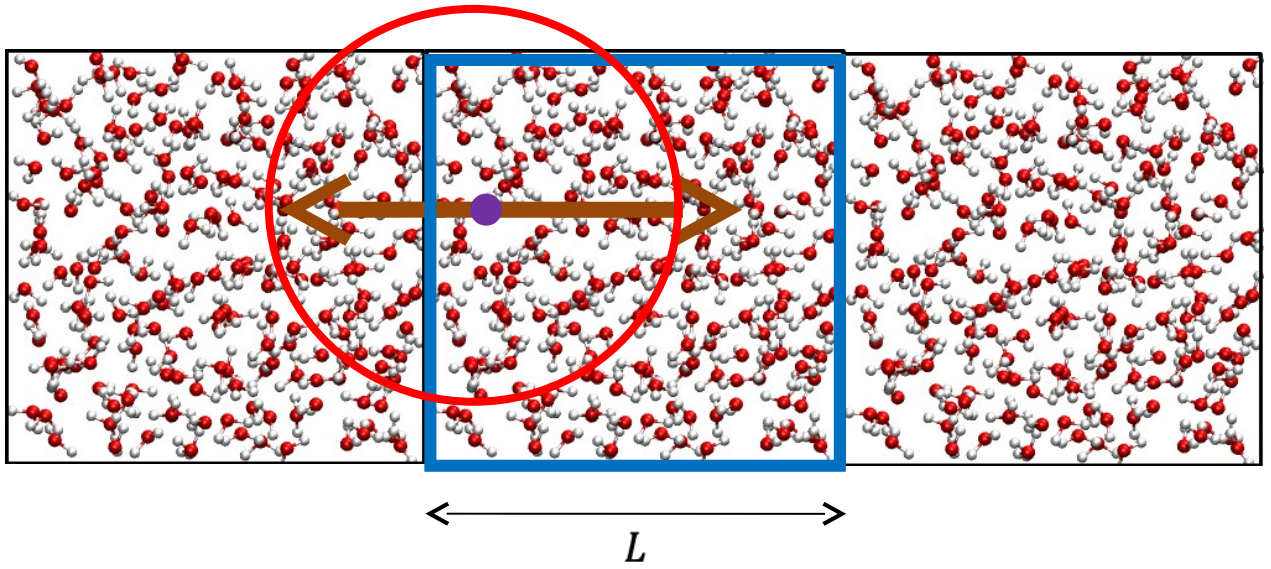
Solid crystal structures
with nonidealities



DaniFeri at Wikimedia Commons
CC

The choice of periodic boundary condition depends on a system and the phenomena studied

Periodic Boundary Conditions – minimum image convention



Minimum image convention:

Only the properties of the original box should be propagated

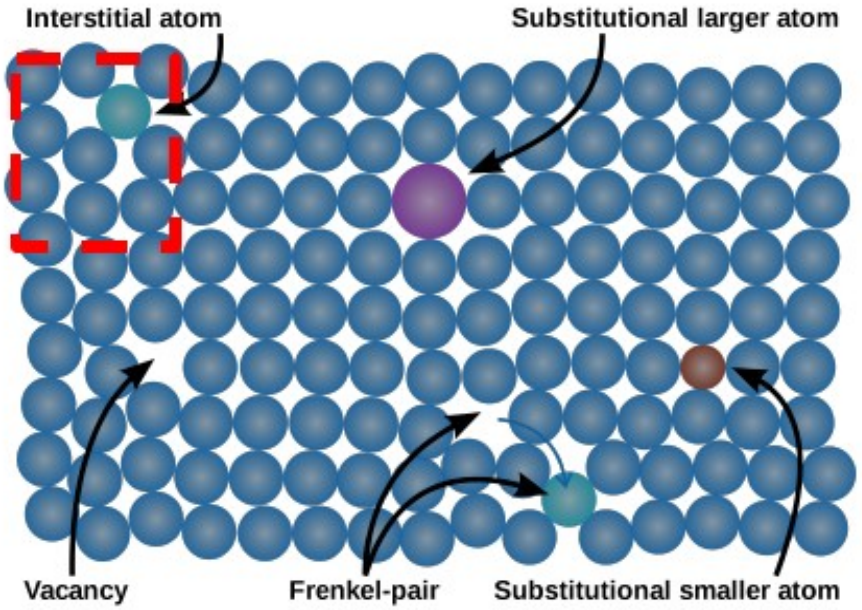
-> only interactions within $^{L}/_{2}$ radius can be considered

Benefits:

- reduced influence of periodicity
- lower computational costs

PBC – Example problems I

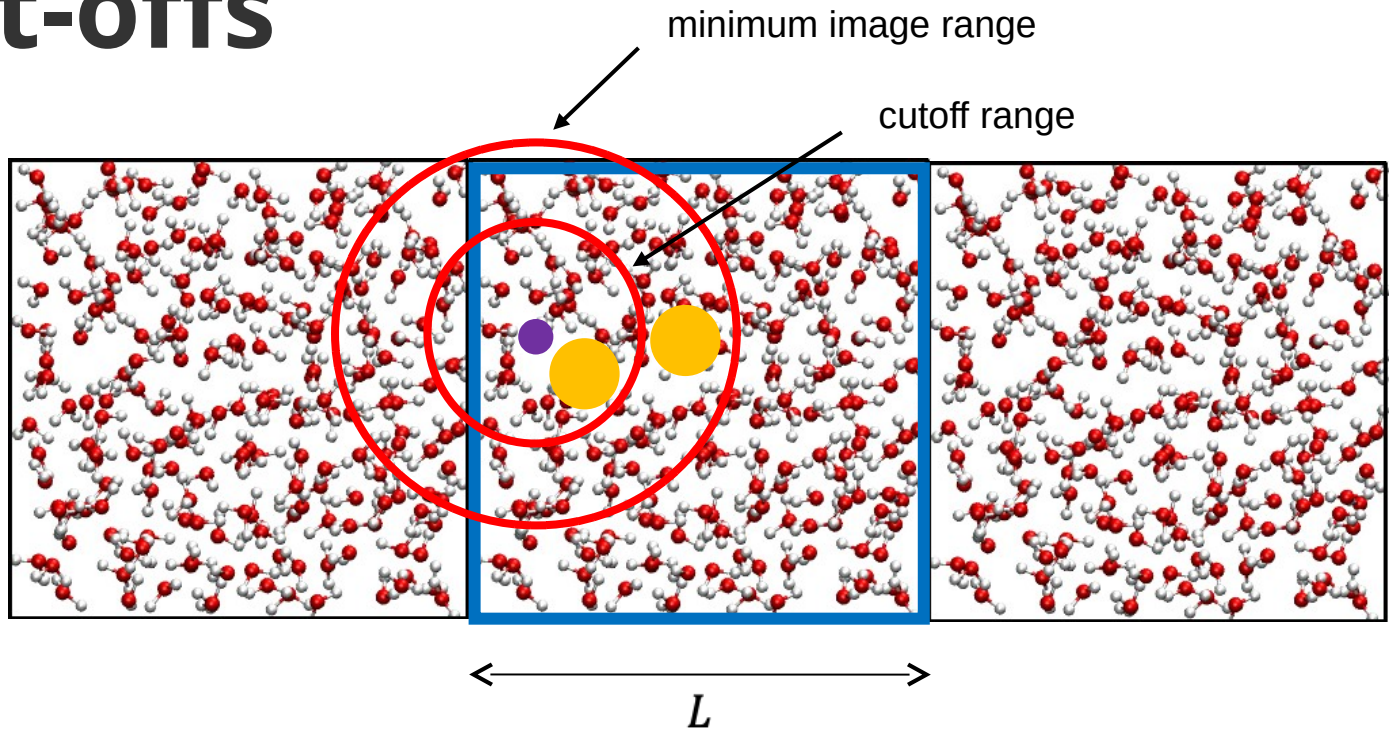
Solid crystal structures
with nonidealities



DaniFeri at Wikimedia Commons
CC

- p.b.c. introduce artificial periodicity, so not very well suitable to model defects and nonidealities in crystal structures if a single unit cell is used
- solution: bigger box (include several unit cells)

Periodic Boundary Conditions and Cut-offs

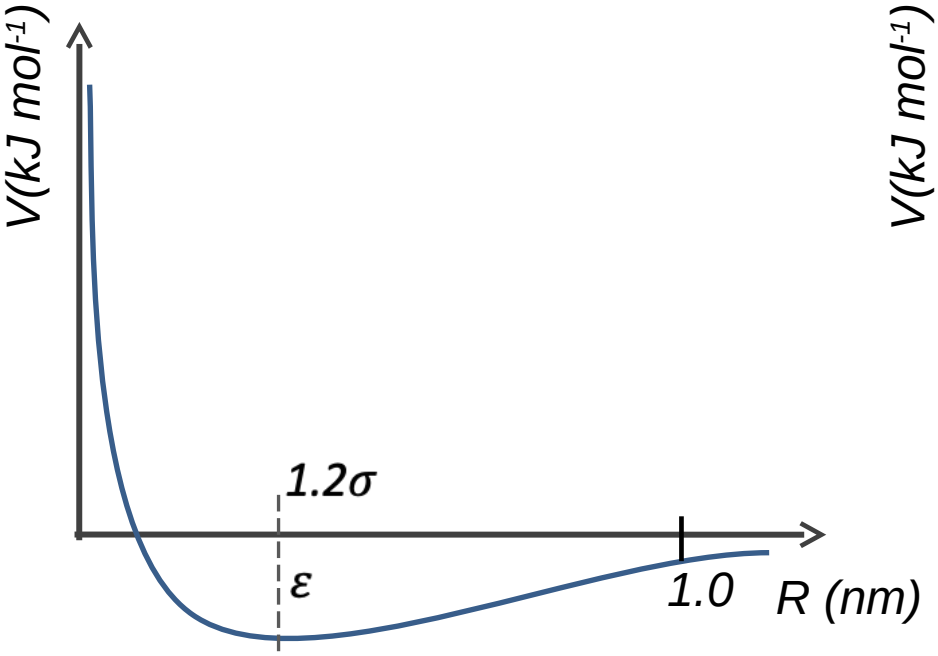


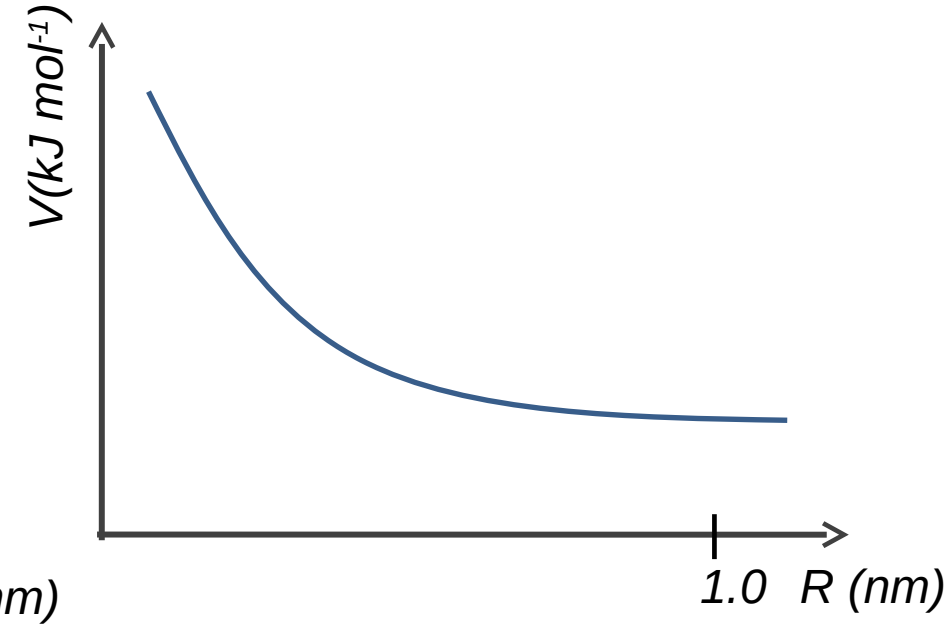
- To achieve minimum image convention, in practice, cut-offs are used:

$$\text{if } r_{ij} < cutoff \text{ then } E_{ij} = E(r_{ij}), \text{ else } E_{ij} = 0$$

- with periodic boundary conditions, cut-off $< \frac{L}{2}$!

Cut-offs of interactions





$$V_{LJ}(r_{ij}) = \frac{C_{ij}^{(12)}}{r_{ij}^{12}} - \frac{C_{ij}^{(6)}}{r_{ij}^{6}}$$

charge-charge $V_{ij} \sim \frac{1}{r}$

Lennard-Jones negligible (<1%)
at typically 0.9-1.2nm
→ cut-off can be used

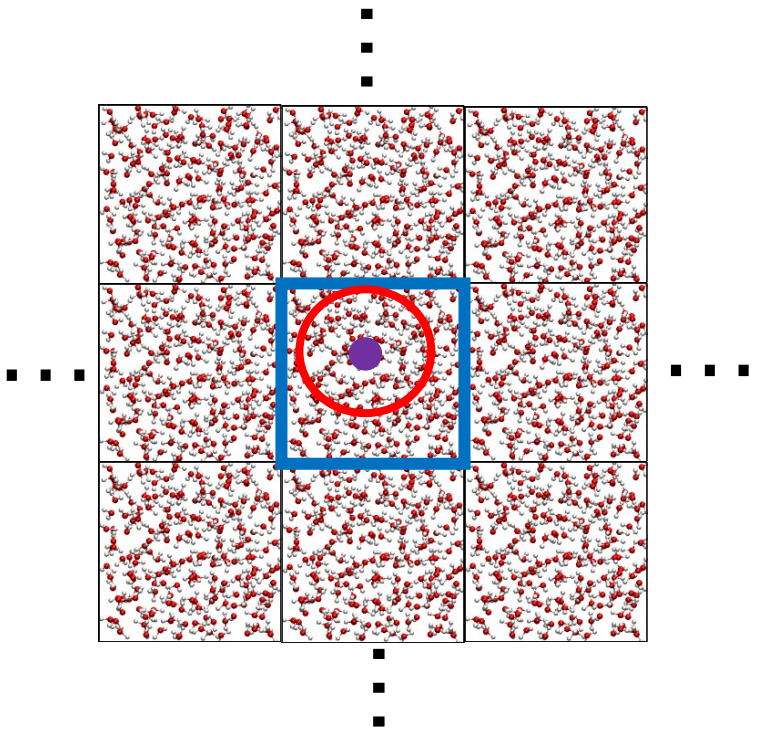
decay very slowly (ch-ch 1% at ~100nm)
→ cut-off cannot be used
(solution: Ewald –type methods)

Particle Mesh Ewald idea

Approximate calculation of electrostatic forces in periodic systems

Assumptions:

- periodic boundary conditions
- neutral system (equal number of positive and negative charges)



infinite number of periodic images

$$V(r_i) = \sum_{j}^{\infty} \frac{q_j}{r_{ij}}$$

infinite sum, conditionally convergent, slow to calculate

Particle Mesh Ewald method

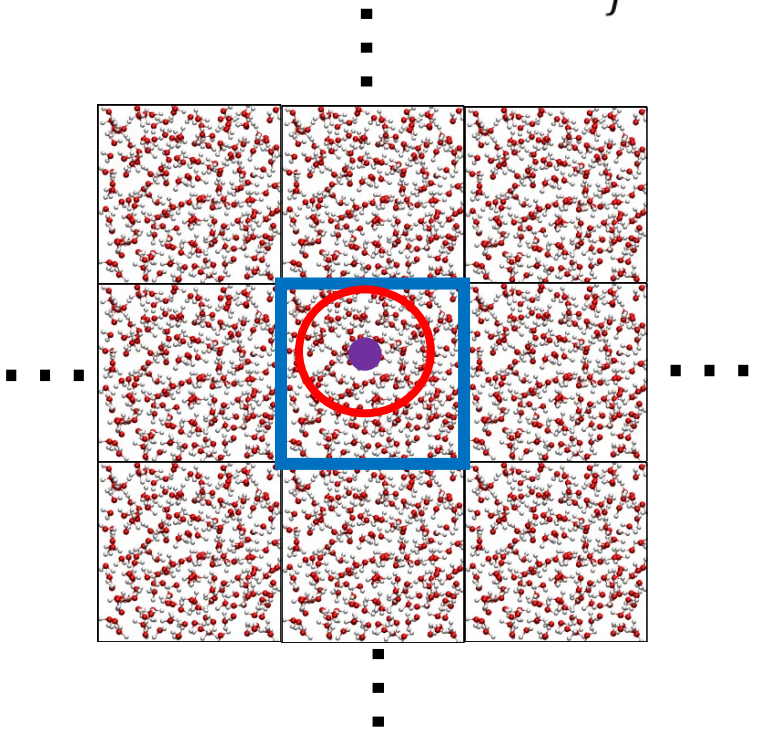
Trick – divide the sum into two rapidly converging components:

- short-range (within some relatively short cutoff) solved in real space
- long-range (originating mostly from periodic images) – this is mostly periodic so can be fast solved in reciprocal space)

$$V(r_i) = \sum_{j}^{\infty} \frac{q_j}{r_{ij}} = V_{short\ range} + V_{long\ range} = \sum_{k}^{j_{cutoff}} \frac{q_j}{r_{ij}} + V_{long\ range}$$

non-periodic but limited and easy to calculate

periodic so infinite sum can be easily calculated in the Fourier space



$$V_{long\ range} = \sum_{k} \tilde{\Phi}(k)|\tilde{\varrho}(k)|^2$$

Fourier transform of potential and charge density

How to Solve Poisson's Equation Using Fourier Transforms

Steps

Download Arti

1 **Begin with Poisson's equation.** Recall that the electric field $\mathbf{E}$ can be written in terms of a scalar potential $\mathbf{E} = -\nabla\phi$. We can then use Gauss' law to obtain Poisson's equation as seen in electrostatics.

- $\nabla^2\phi = -\frac{\rho}{\epsilon_0}$
- In this equation, it is often the case that we know the charge density ρ, called the source function, and wish to know the potential ϕ. Therefore, we need to find some way to invert this equation.

2 **Write out the FTs and inverse FTs of the potential and charge density.** Since we are dealing with three dimensions, the FTs are adjusted accordingly, with the constant factor there for normalization purposes. The bounds will differ depending on conventions on where to set the potential to 0. Although we will not explicitly write the boundaries until evaluating the integrals, we will set the potential to 0 at infinity, so that we are integrating over all space.

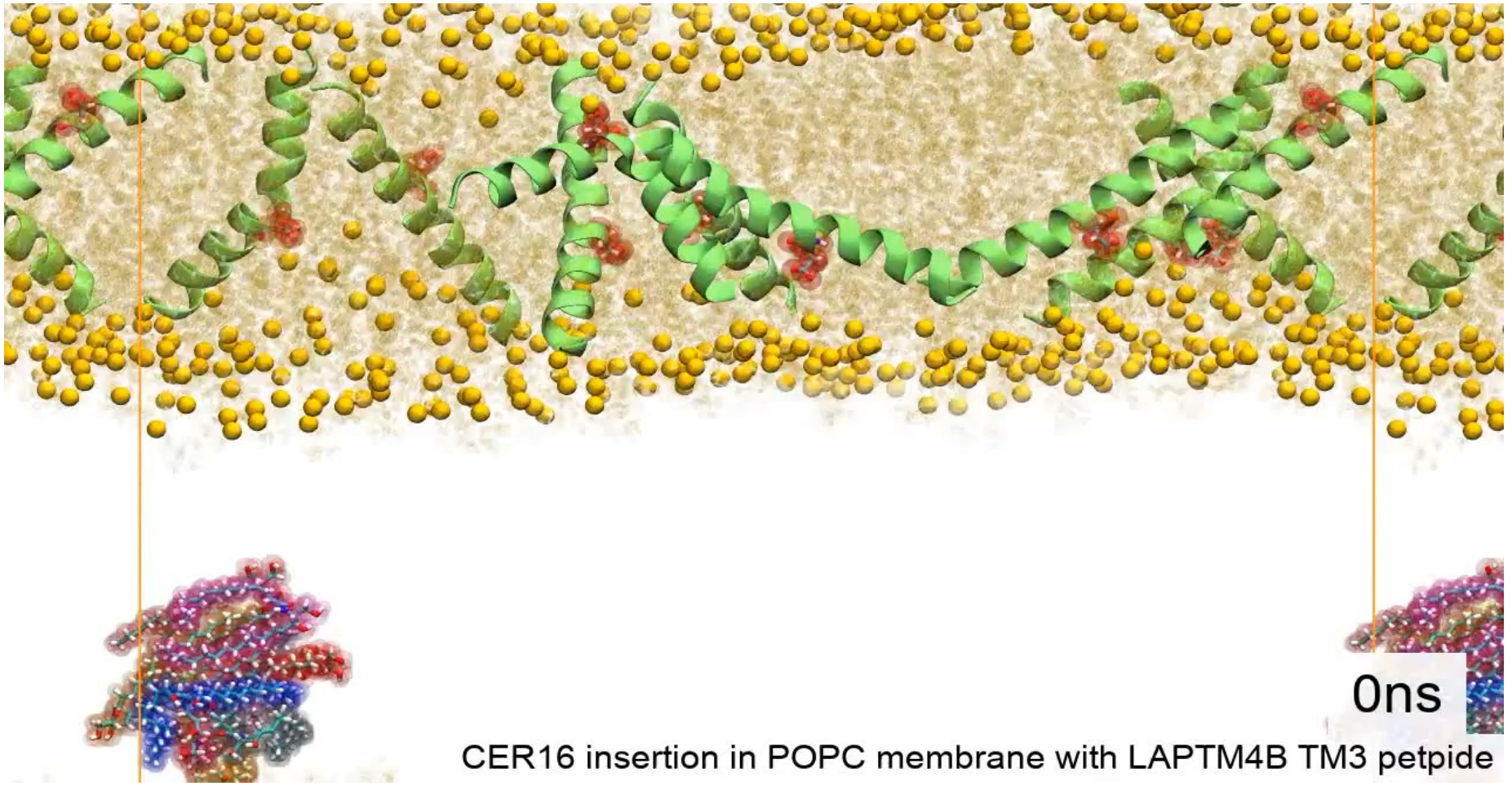
- $\tilde{\phi}(\mathbf{k}) = \frac{1}{(2\pi)^{3/2}}\int \phi(\mathbf{x})e^{-i\mathbf{k}\cdot\mathbf{x}}\mathrm{d}^3\mathbf{x}$
- $\phi(\mathbf{x}) = \frac{1}{(2\pi)^{3/2}}\int \tilde{\phi}(\mathbf{k})e^{i\mathbf{k}\cdot\mathbf{x}}\mathrm{d}^3\mathbf{k}$
- $\tilde{\rho}(\mathbf{k}) = \frac{1}{(2\pi)^{3/2}}\int \rho(\mathbf{x})e^{-i\mathbf{k}\cdot\mathbf{x}}\mathrm{d}^3\mathbf{x}$
- $\rho(\mathbf{x}) = \frac{1}{(2\pi)^{3/2}}\int \tilde{\rho}(\mathbf{k})e^{i\mathbf{k}\cdot\mathbf{x}}\mathrm{d}^3\mathbf{k}$

3 **Relate $\tilde{\phi}(\mathbf{k})$ with $\tilde{\rho}(\mathbf{k})$.** The result will relate the potential and charge density in the $\mathbf{k}$ space, and as it will turn out, the relation is algebraic, which is considerably simpler.

- Take the Laplacian of $\phi(\mathbf{x})$. We can differentiate under the integral here because the integral is being taken with respect to $\mathbf{k}$, and $\mathbf{x}$ is an independent variable.
  - $\nabla^2\phi(\mathbf{x}) = \frac{1}{(2\pi)^{3/2}}\int -k^2 e^{i\mathbf{k}\cdot\mathbf{x}}\tilde{\phi}(\mathbf{k})\mathrm{d}^3\mathbf{k} = \frac{\rho(\mathbf{x})}{\epsilon_0}$
- FT charge density so that it is also written in the $\mathbf{k}$ space.
  - $\frac{\rho(\mathbf{x})}{\epsilon_0} = \frac{1}{(2\pi)^{3/2}}\int \frac{\tilde{\rho}(\mathbf{k})}{\epsilon_0}e^{i\mathbf{k}\cdot\mathbf{x}}\mathrm{d}^3\mathbf{k}$
- By direct comparison, we see that the below relation holds.
  - $k^2\tilde{\phi}(\mathbf{k}) = \frac{\tilde{\rho}(\mathbf{k})}{\epsilon_0}$
- If we were given charge density in the $\mathbf{k}$ space and wanted to find potential in the same space, it would be very easy. However, we are interested in finding these quantities in the $\mathbf{x}$ space. Therefore, we will need to transform a second time.

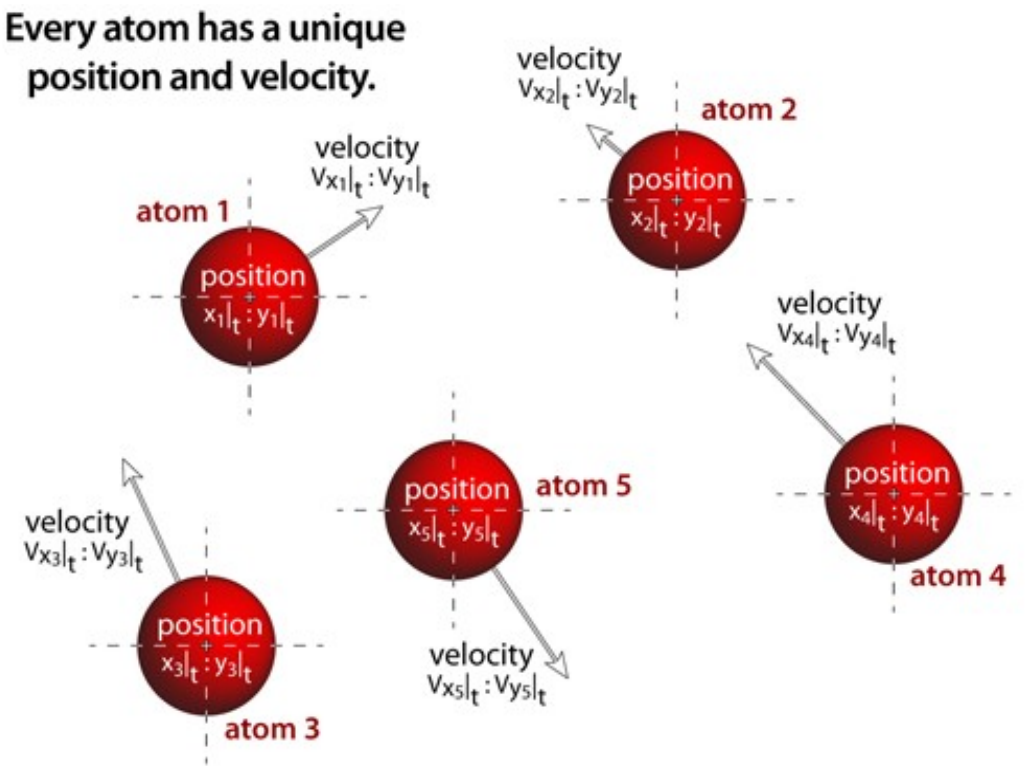
https://www.wikihow.com/Solve-Poisson%27s-Equation-Using-Fourier-Transforms#

MD simulations



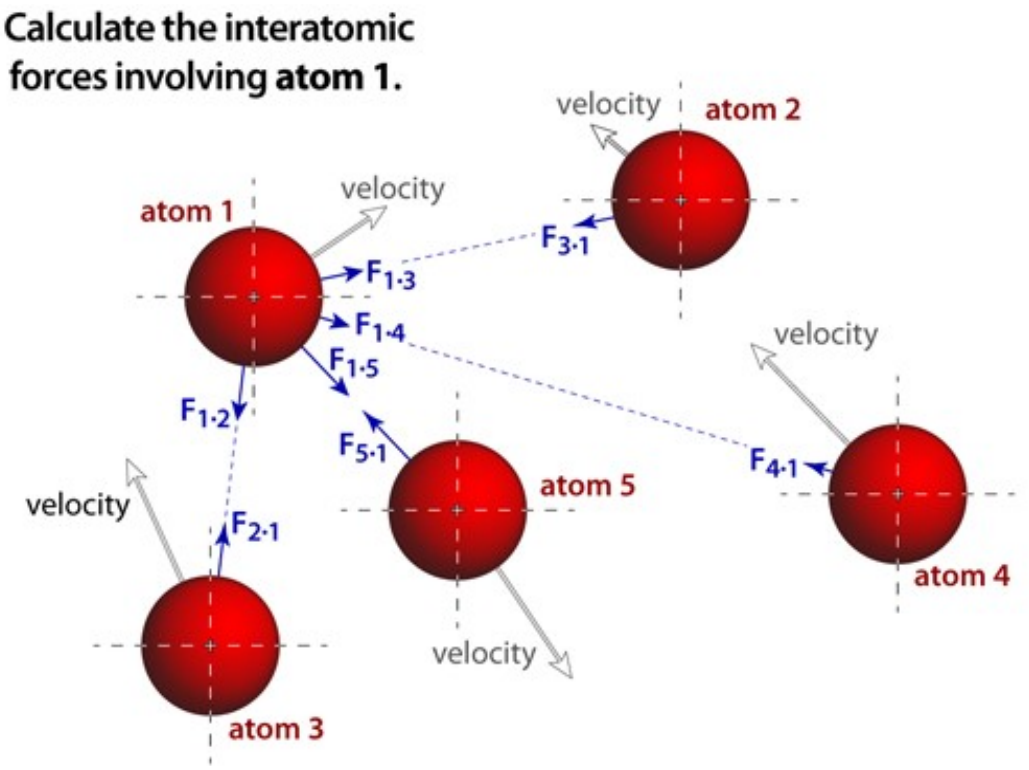
CER16 insertion in POPC membrane with LAPTM4B TM3 petpide

MD: step 1



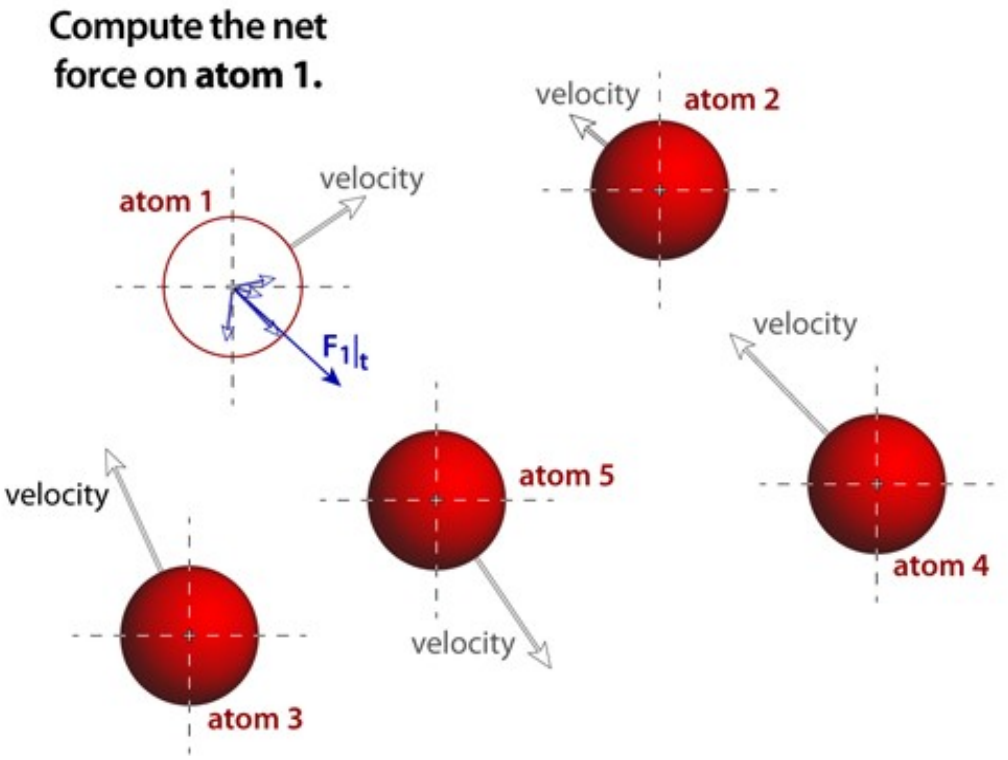
http://atomsinmotion.com/book/chapter5/md

MD: step 2



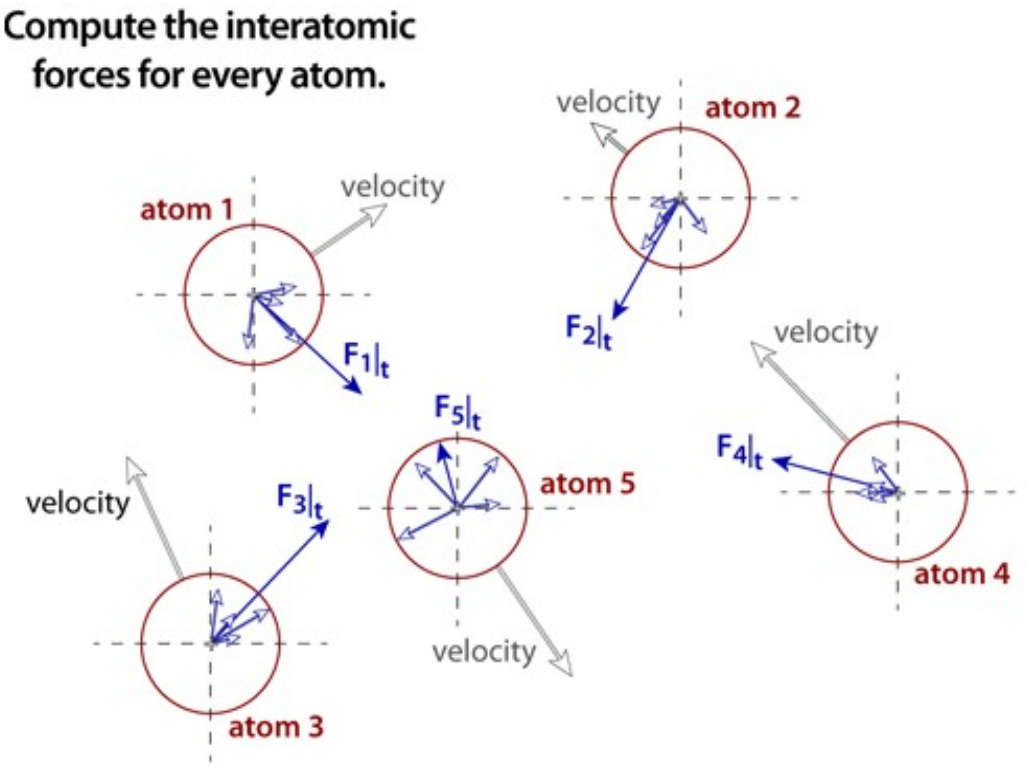
http://atomsinmotion.com/book/chapter5/md

MD: step 3



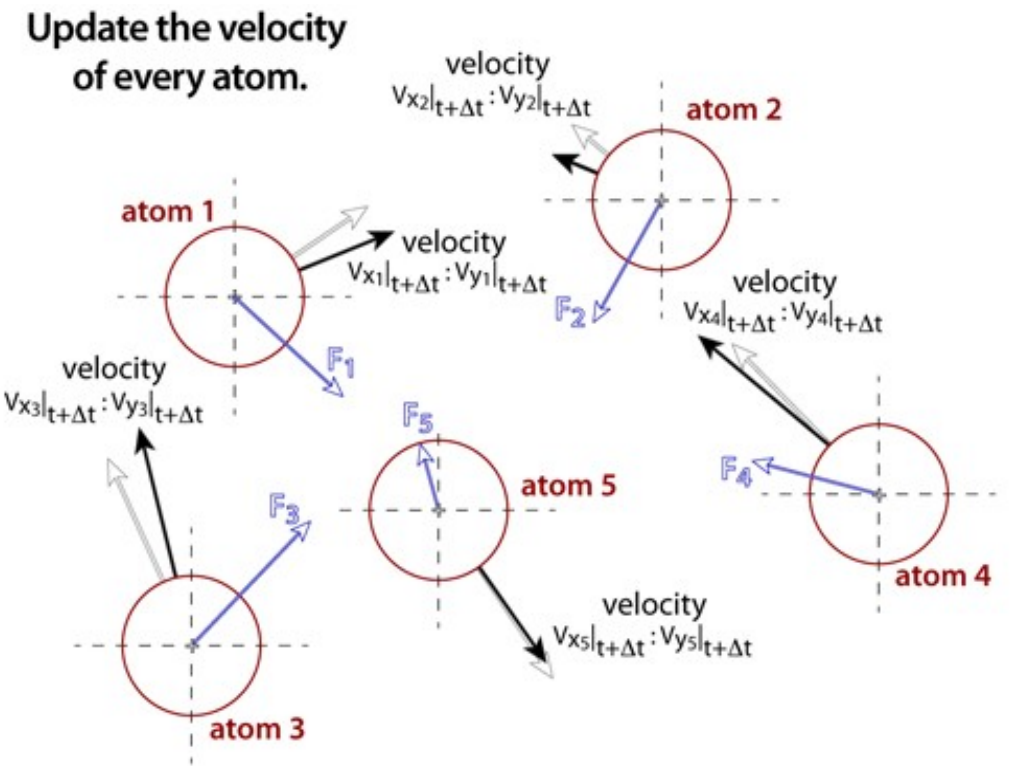
http://atomsinmotion.com/book/chapter5/md

MD: step 4



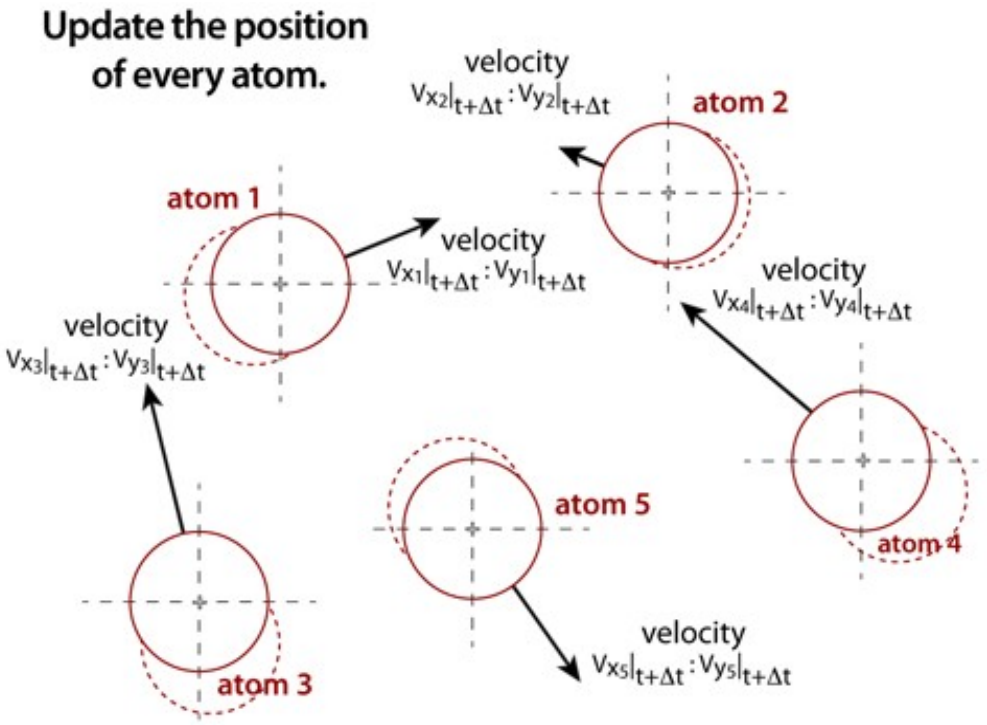
http://atomsinmotion.com/book/chapter5/md

MD: step 5



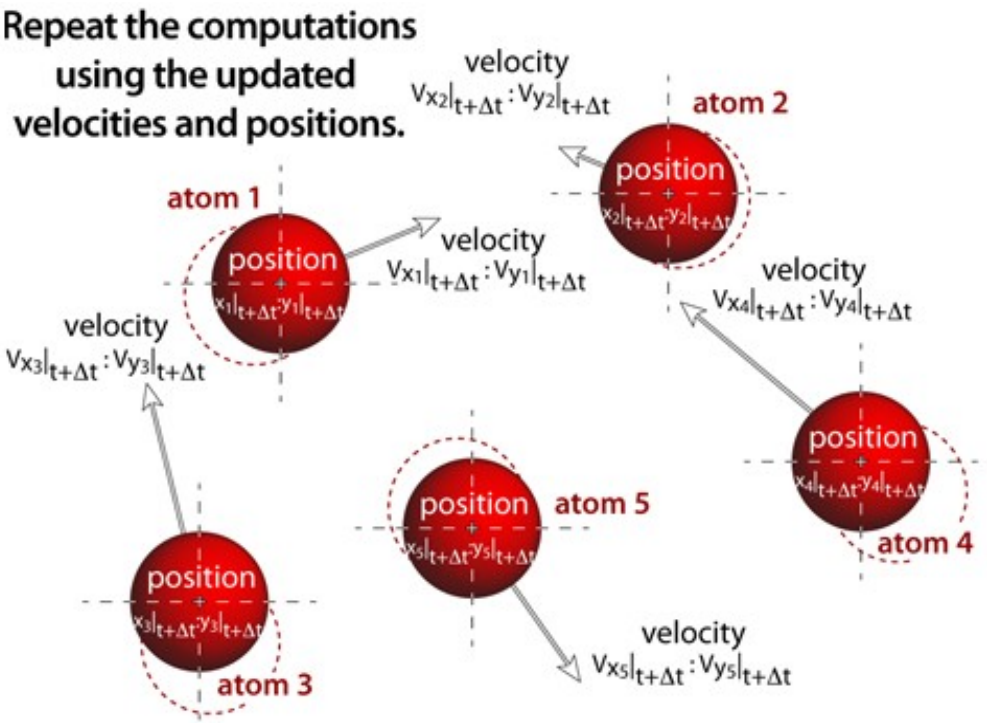
http://atomsinmotion.com/book/chapter5/md

MD: step 6



http://atomsinmotion.com/book/chapter5/md

MD: step 7



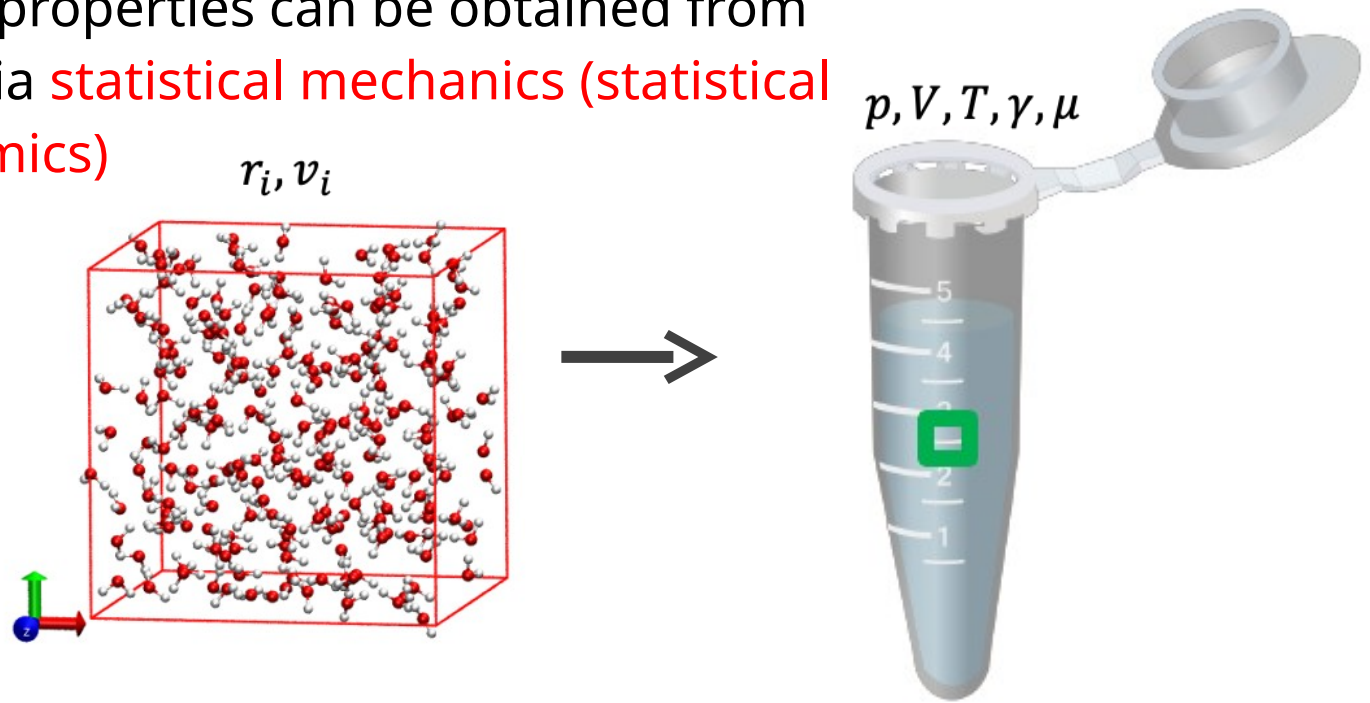
http://atomsinmotion.com/book/chapter5/md

Macroscopic properties

Newton equations of motion govern microscopic evolution and do not explicitly give macroscopic properties (e.g. temperature, pressure, surface tension)

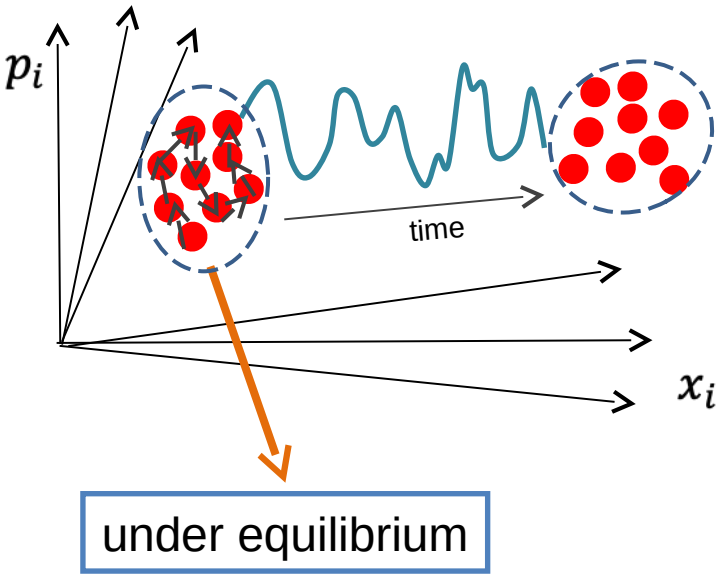
$$-\frac{dV}{d\vec{r_i}} = m_i \frac{d^2\vec{r_i}}{dt^2}$$

Macroscopic properties can be obtained from trajectories via statistical mechanics (statistical thermodynamics)



Statistical ensembles - equilibrium

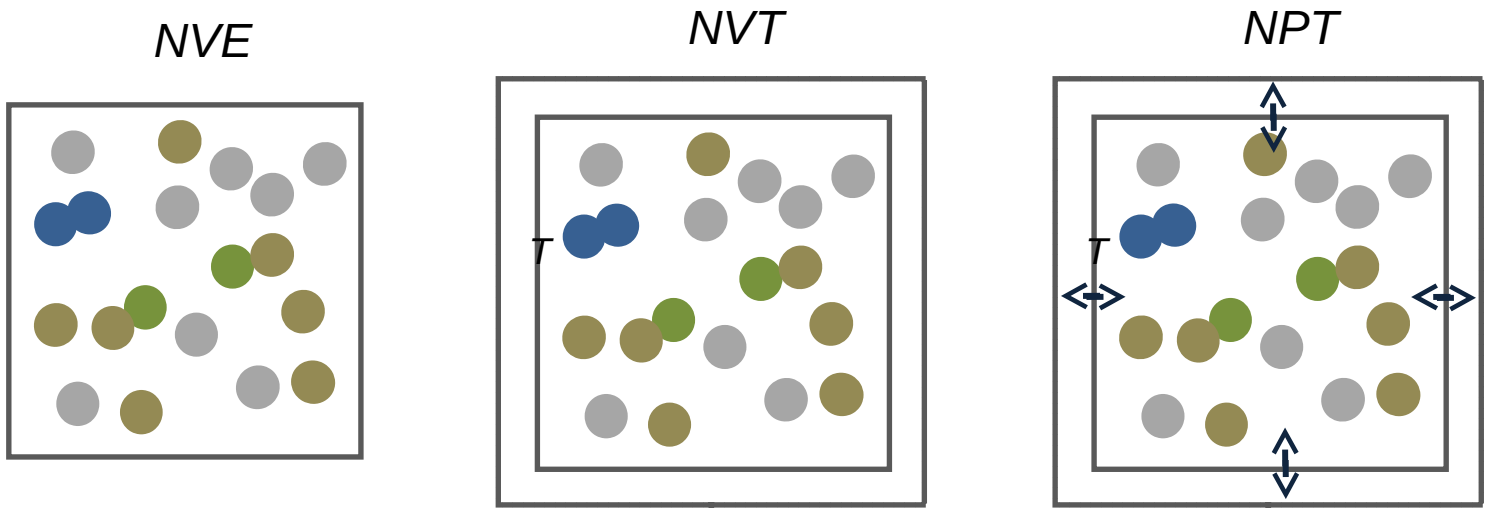
- **Ergodic hypothesis**: all accessible microstates are equiprobable over a long period of time (different formulations, many consequences!)
  - Average over ensemble = average over time
- Macroscopic properties are given by probability distribution within an ensemble



- In equilibrium MD, we “move” within the ensemble, reproduce its probability distribution and hence we can estimate macroscopic properties, one long MD trajectory is in principle sufficient
- Macroscopic properties can be estimated along a MD trajectory
- Equilibrium dynamic properties can also be studied (e.g. self-diffusion)

Statistical ensembles

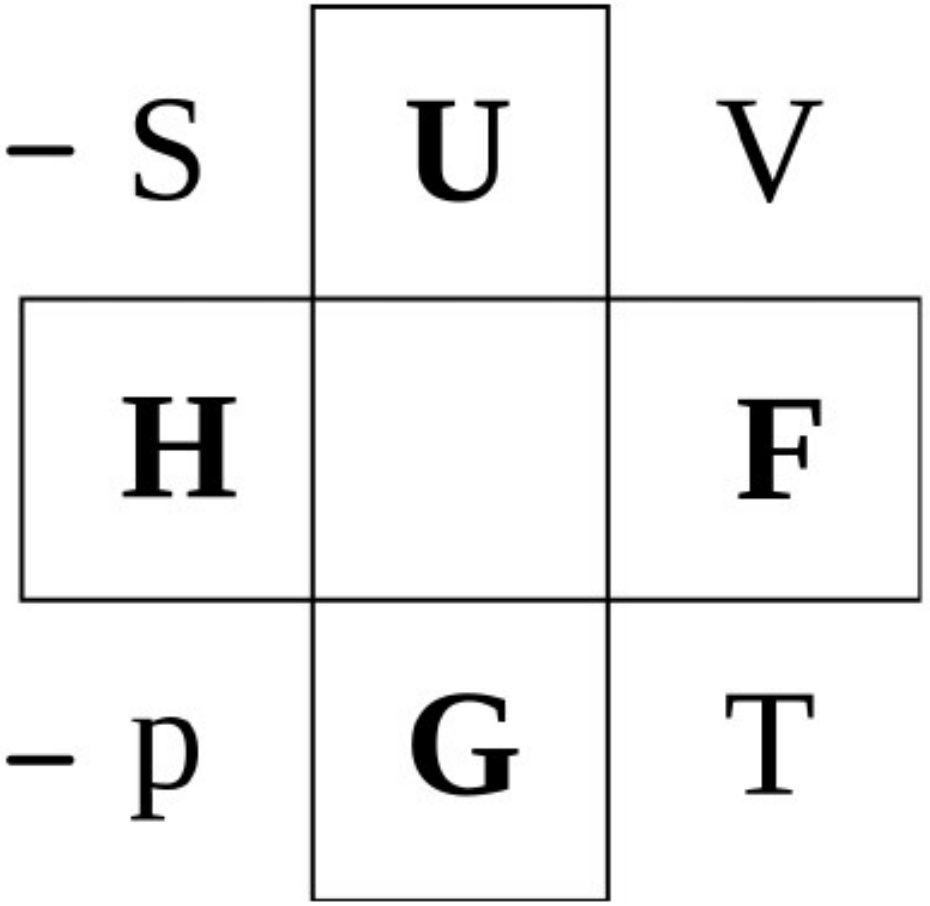
- thermodynamic statistical ensembles describe macroscopic conditions
- *NVE* – microcanonical
- *NVT* – canonical (other names: isothermal, Helmholtz canonical)
- *μVT* – grand-canonical
- *NPE* – isobaric
- *NPT* – isobaric-isothermal (other name: Gibbs canonical)



$$\Omega(N,V,E) = \frac{1}{N!}\sum_X \delta[\mathscr{H}(X) - E] \qquad Z(N,V,T) = \frac{1}{N!}\sum_X e^{-\beta\mathscr{H}(X)} \qquad Q(N,P,T) = \int dV e^{-\beta PV}\frac{1}{N!}\sum_X e^{-\beta\mathscr{H}(X)}$$

$$S(N,V,E) = k_B ln\Omega(N,V,E) \qquad F(N,V,T) = -k_B T ln Z(N,V,T) \qquad G(N,P,T) = -k_B T ln Q(N,P,T)$$

Maxwell relations:
calculating thermodynamic functions

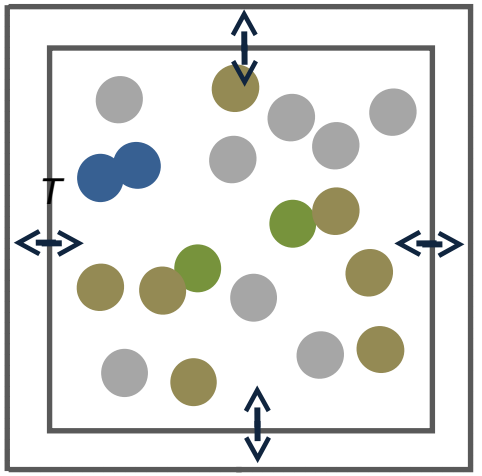


RoB at Wikimedia Commons
CC

Statistical ensembles

Temperature:

$$T = \frac{2}{3k_B}\left\langle \frac{1}{2}mv^2 \right\rangle$$



Pressure:

$$p = \frac{2N}{3V}\left\langle \frac{1}{2}mv^2 \right\rangle$$

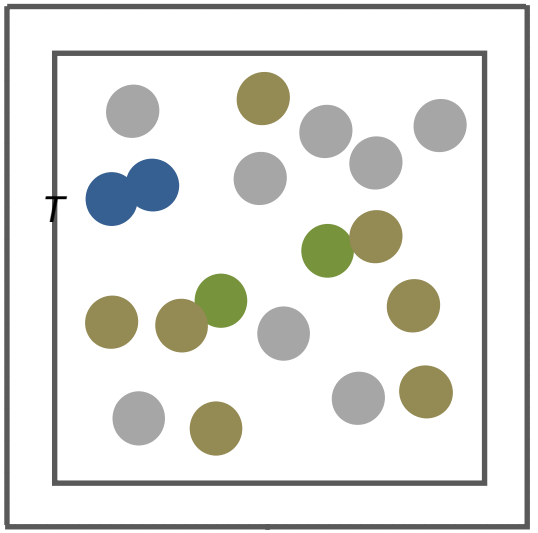
- Using statistical thermodynamics, both temperature and pressure can be related to average kinetic energy so they can be obtained from Newton equations of motion
- But Newton's equations of motion correspond to NVE (actually, even more restrictively: *E=const*)
- Therefore, fluctuations of temperature and pressure must be added by using special algorithms: thermostat and barostat (temperature- and pressure-coupling algorithms

Nosé-Hoover thermostat

- A friction term introduced in the eq. of motion:

$$\frac{d^2 r_i}{dt^2} = \frac{F_i}{m_i} - \lambda v_i$$

λ – friction coefficient



$$\frac{\mathrm{d}^2 \boldsymbol{r}_i}{\mathrm{d}t^2} = \frac{\boldsymbol{F}_i}{m_i} - \frac{p_\xi}{Q}\frac{\mathrm{d}\boldsymbol{r}_i}{\mathrm{d}t}$$

ξ – friction parameter ("heat bath" variable)
p_ξ - friction momentum
Q – mass parameter

$$\frac{\mathrm{d}p_\xi}{\mathrm{d}t} = (T - T_0)$$

$$Q = \frac{\tau_T^2 T_0}{4\pi^2}$$

τ_T - coupling constant (by analogy to Berendsen algorithm)

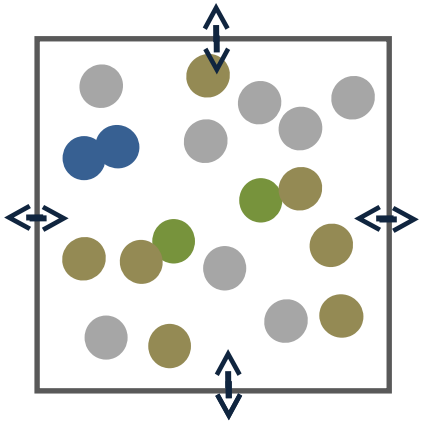
- additional equation of motion solved for the new variable ξ

List of thermostats

- Nosé-Hoover
- Langevin
- Stochastic velocity rescaling (V-rescale)
  - (More stable but only in Gromacs)
- Berendsen (Do not use for production run !!)

Parrinello-Rahman barostat

- similar concept to Nosé-Hoover thermostat
- a friction term introduced in the eq. of motion:



$$\frac{\mathrm{d}^2 \boldsymbol{r}_i}{\mathrm{d}t^2} = \frac{\boldsymbol{F}_i}{m_i} - \boldsymbol{M}\frac{\mathrm{d}\boldsymbol{r}_i}{\mathrm{d}t}$$

$$\boldsymbol{M} = \boldsymbol{b}^{-1}\left[\boldsymbol{b}\frac{\mathrm{d}\boldsymbol{b}'}{\mathrm{d}t} + \frac{\mathrm{d}\boldsymbol{b}}{\mathrm{d}t}\boldsymbol{b}'\right]\boldsymbol{b}'^{-1}$$

τ_P - coupling constant
β – compressibility
P_0- desired pressure
b – box vectors matrix

$$\frac{\mathrm{d}\boldsymbol{b}^2}{\mathrm{d}t^2} = V\boldsymbol{W}^{-1}\boldsymbol{b}'^{-1}(\boldsymbol{P} - \boldsymbol{P}_{ref})$$

$$r_i := \mu \cdot r_i$$
$$V := \mu^3 \cdot V$$

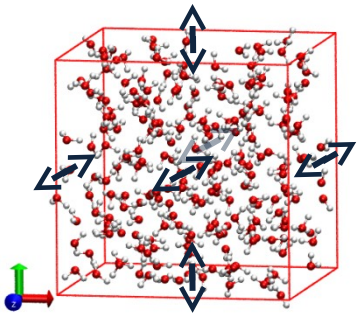
$$\left(\boldsymbol{W}^{-1}\right)_{ij} = \frac{4\pi^2 \beta_{ij}}{3\tau_p^2 L}$$

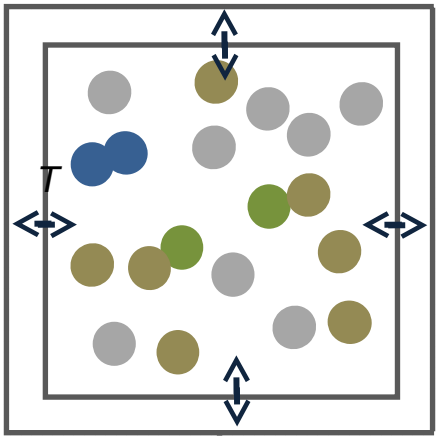
List of barostats

- Stochastic cell rescaling (c-rescale)
  - (More stable but only in Gromacs)
- Parrinello-Rahman
- Berendsen

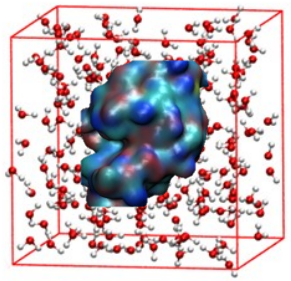
Temperature and pressure in MD - practicalities

very often NPT ensemble is used (thermostat + barostat) (with p=1 atm, T=293 K or T=310 K)

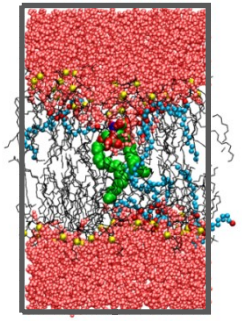




$p_x = p_y = p_z$



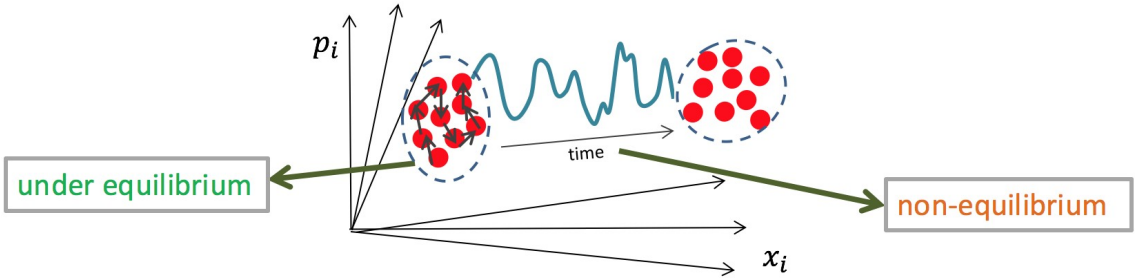
$p_x = p_y \neq p_z$



$p_x = p_y \neq p_z = 0$

- few thermostats can be introduced, e.g., 1) protein 2) water
- surface tension ensemble can be used (NγT)

Simulation protocol

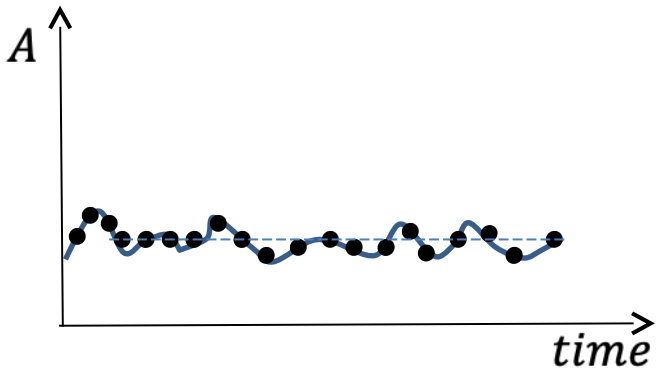


- **A) equilibrium MD**
  - one very long trajectory required if the system is ergodic
  - MD trajectory corresponds to 'jumps' between microstates in a given equilibrium ensemble
  - system properties calculated as averages in time (= average in ensemble)
  - time not important *per se* but time-dependent phenomena can be studied (e.g., self-diffusion)

- **B) non-equilibrium MD:**
  - many trajectories required
  - each trajectory corresponds to one possible evolution path in the phase-space
  - time is important
  - system properties calculated as averages over a set of trajectories (mean values are functions of time)

The average concept

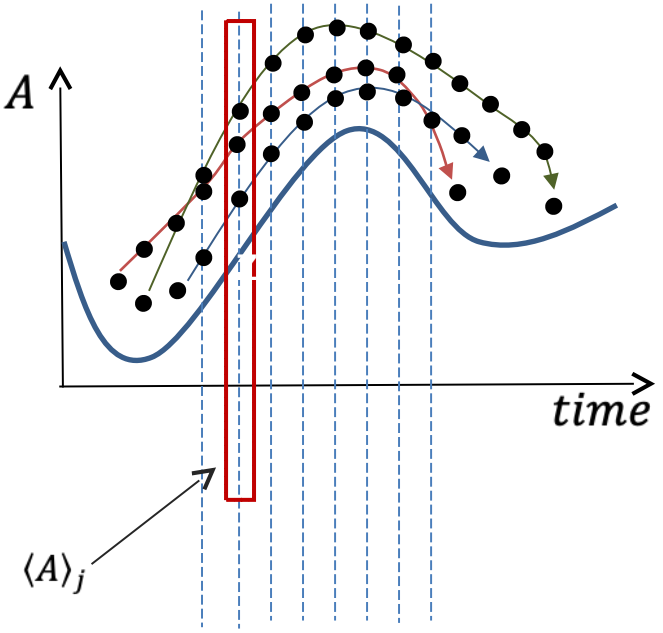
Under equilibrium



$$\langle A \rangle_{time} = \frac{1}{N} \sum_{i=1}^{N} A(t_i)$$

averaging over N points taken along the trajectory each δt
(usually $\delta t > \Delta t$)
the result is time-independent

Out-of-equilibrium

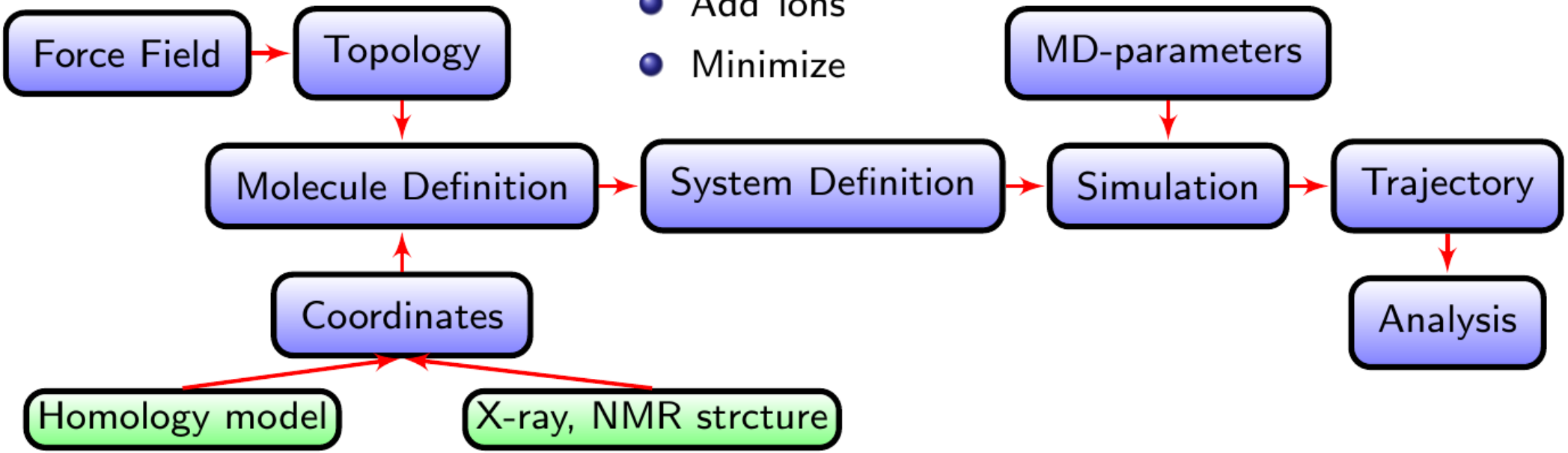


$$\langle A(t) \rangle_{set} = \langle A \rangle_j (t_i)$$

averaging over j trajectories each δt
(usually $\delta t > \Delta t$)
the result is a function of time

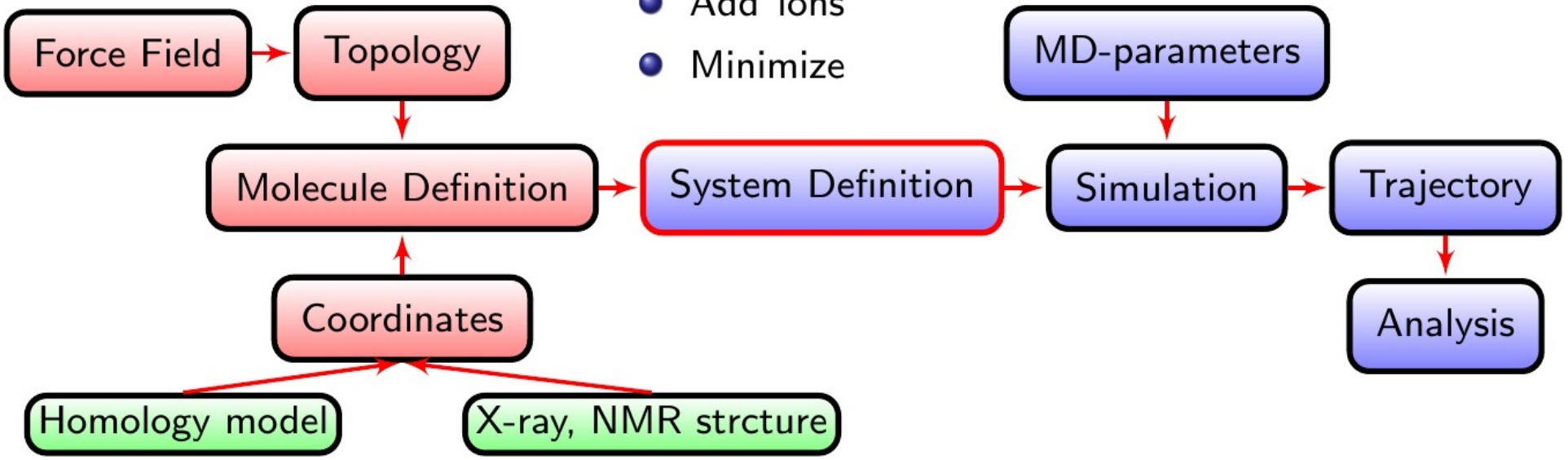
Typical MD workflow

- Define unit cell
- Assemble the system
- Add solvent
- Add ions
- Minimize



Stage: Defining the system

- Define unit cell
- Assemble the system
- Add solvent
- Add ions
- Minimize



Designing /Understanding your system

- Lipids:
  - Composition
  - Mixing time
  - Asymmetric systems
- Other molecules
  - Ionic concentration
  - Experimental buffers
  - Availability of force field
- Proteins
  - Complete 3d structure?
  - Protonation state
  - Ligands (ff?)
- Not too small or too big
  - PBC effect
  - Computational cost
- Initial molecular arrangement

Force field in biosystems

$$-\frac{dV_{total}}{d\vec{r}_i} = m_i \frac{d^2\vec{r}_i}{dt^2}$$

$$U = \sum_{bonds} \frac{k_i^b}{2}\left(\mathbf{r_i} - \mathbf{r_i^{eq}}\right)^2 + \sum_{angles} \frac{k_i^\alpha}{2}\left(\theta_i - \theta_i^{eq}\right)^2 +$$

$$+ \sum_{torsions} \sum_{n} k_{\phi,n}\left(1 + \cos\left(n\phi - \phi^{ref}\right)\right) +$$

$$+ \sum_{pairs} \left(\left\{ \frac{C_{ij}^{(12)}}{r_{ij}^{12}} - \frac{C_{ij}^{(6)}}{r_{ij}^{6}} \right\}_{vdw} + \left\{ k_e \frac{q_i q_j}{r_{ij}} \right\}_{coul} \right)$$

Main force fields use in biosystems

- AMBER
- CHARMM
- OPLS
- GROMOS
- Glycan
- MARTINI (coarse grained)
- ProsECCo (ECC)

Choosing the force field

- Preferably choose a force field which is currently widely used
  - Berger was the best for lipids, not anymore.
- At least you will be able to have controls
- Choose a force field that describe your target properties adequately
  - Martini
- The force field is widely used in the community, therefore it must be good.
  WRONG!!!
  - Check always about know issues with the force fields.
  - Check papers benchmarking force fields

THIS STEP WILL DEPEND IN THE USED ENGINE.
(It should be the way around)
Don't try to fix a bulb with a hammer

Building the System

- Manual
  - Full control
  - Too much job
  - Error prone
  - Only for learning and debugging

Martinez-Seara, H. & Rog, T. **Molecular dynamics simulations of lipid bilayers: Simple recipe of how to do it**. Methods Mol. Biol., 2013, 924, 407-429

- Tools like CHARMM-GUI
  - Important changes without warning.
  - No version control (±)
    - Tomorrow might be different and you have no clue
  - Save the download date
  - Be aware of the atom order

THIS STEP WILL DEPEND IN THE USED ENGINE.
There are no standards.

Universal builder

CHARMM-GUI (http://www.charmm-gui.org/)

CHARMM-GUI
Effective Simulation Input Generator and More

CHARMM is a versatile program for atomic-level simulation of many-particle systems, particularly macromolecules of biological interest. - M. Karplus

about us :: input generator :: Q&A :: archive :: charmm docs :: lectures :: movie gallery :: video demo :: citations :: update log :: jobs & events :: giving

Some lectures and job postings are now available. See upload log for update history and giving for donation. Contact info is given below.

Input Generator
- Job Retriever
- PDB Reader
- Glycan Reader & Modeler
- Ligand Reader & Modeler
- Glycolipid Modeler
- LPS Modeler
- Nanomaterial Modeler
- Multicomponent Assembler
- Solution Builder
- Membrane Builder
- Martini Maker
- PACE CG Builder
- Drude Prepper
- Free Energy Calculator
- MAP Utilizer
- DEER Facilitator
- NMR Structure Calculator
- PBEQ Solver
- Implicit Solvent Modeler
- Boundary Potential Utilizer
- GCMC/BD Ion Simulator

Input Generator

One easiest way to support CHARMM-GUI is to cite the CHARMM-GUI main paper as well as the papers of the modules used in users' publications. Please see Citations for details.

Since most modules start with PDB Reader, it is strongly recommended to read the PDB Reader page and to see the PDB Reader demo in Video Demo.

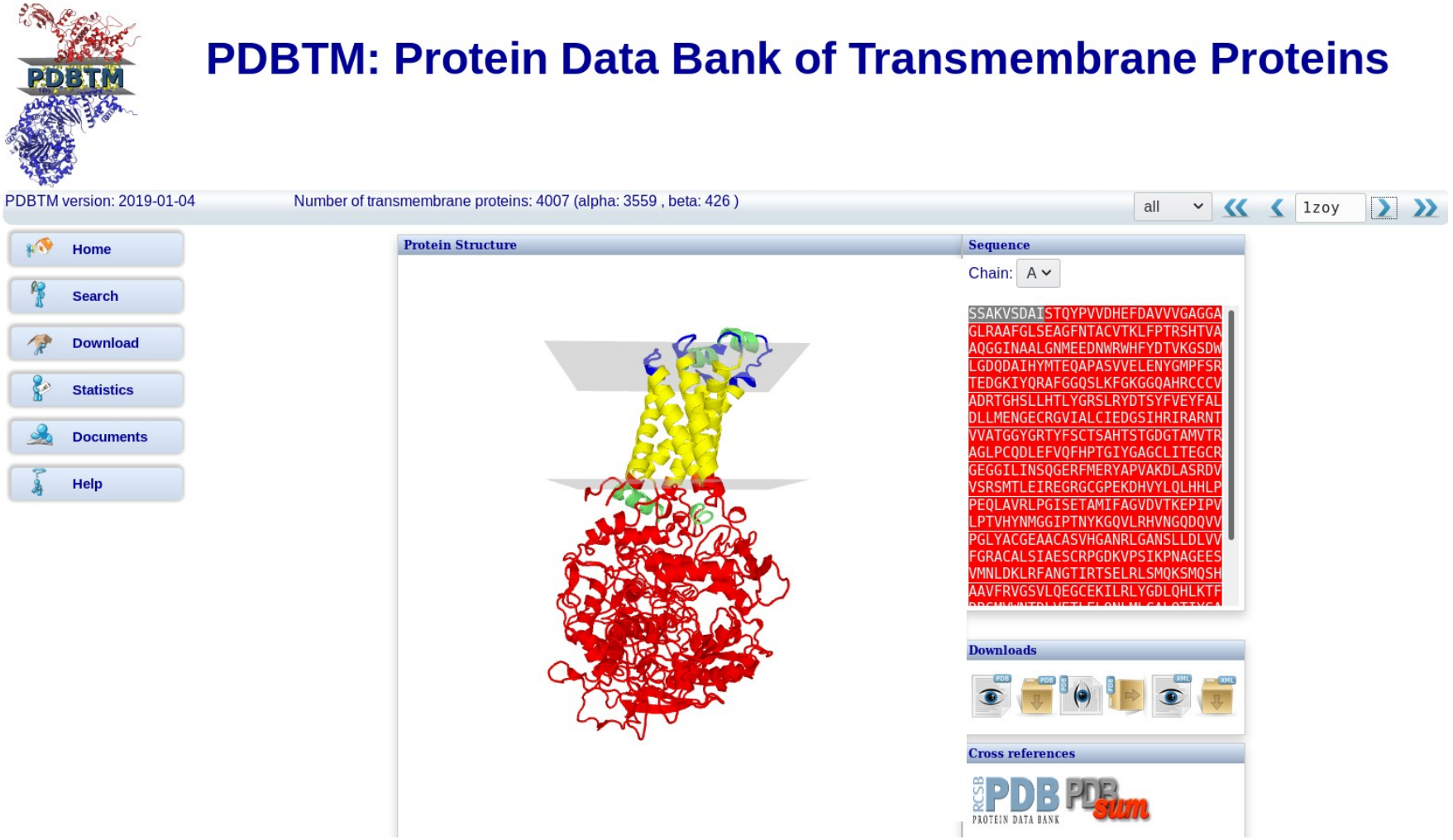
- Job Retriever
  Facilitates recovery of jobs, when the Job ID is known
- PDB Reader
  Read a PDB file (RCSB or CHARMM formats) into CHARMM
- Glycan Reader & Modeler
  Read carbohydrate structures from a PDB file into CHARMM and/or model user-specified N-/O-glycan or glycan-only structure(s)
- Ligand Reader & Modeler
  Generate various ligand structures using the CHARMM force field
- Glycolipid Modeler
  Provide various glycolipid structure and PSF files
- LPS Modeler
  Provide various lipopolysaccharide (LPS) structure and PSF files
- Nanomaterial Modeler
  Generate various nanomaterial systems for molecular dynamic simulation
- Multicomponent Assembler
  Combine PSF/CRD of non-membrane molecules into a heterogeneous system
- Solvator
  Solvate globular protein, or generate various shapes of water box
- Solution Builder (new Quick MD Simulator)
  Setup subsequent steps for molecular dynamics simulations of globular proteins
- Drude Prepper
  Prepare the systems ready for simulations with the Drude polarizable force fields from an identical system equilibrated with the CHARMM36 non-polarizable additive force fields
- Membrane/Bilayer Builder
  Generate a protein/bilayer complex or bilayer-only system for molecular dynamics simulations
- Membrane/Monolayer Builder
  Generate a protein/monolayer complex or monolayer-only system for molecular dynamics simulations
- Membrane/Nanodisc Builder
  Generate a lipid-only or protein-embeded nanodisc system for molecular dynamics simulations

Protein Data Bank [pdb] (https://www.rcsb.org/)

- The best place to find if your target protein has a resolved 3d structure.

Transmembrane protein structural databases

PDBTM: Protein Data Bank of Transmembrane Proteins (http://pdbtm.enzim.hu/)



Transmembrane protein structural databases

OPM database (https://opm.phar.umich.edu)

orientations of proteins in membranes (OPM) database

UNIVERSITY OF MICHIGAN | COLLEGE OF PHARMACY

Search proteins by PDB ID or name

HOME | ABOUT OPM | DOWNLOAD OPM FILES | CONTACT US | PPM SERVER

Protein Classification

Types (3)
Classes (11)
Superfamilies (499)
Families (958)
Species (836)
Localizations (24)
Proteins (4444)

Assembly

Superfamilies (9)
Families (19)
Localizations (8)
Assemblies (207)

Protein Links

PDB Sum, PDB, MPKS, MPDB

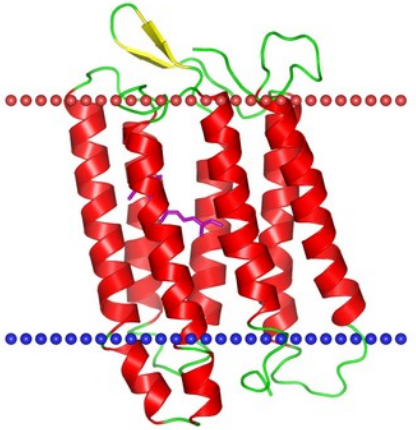
PPM Server

1uaz » Archaerhodopsin-1

- *Type:* Transmembrane (3 classes)
- *Class:* Alpha-helical polytopic (123 superfamilies)
- *Superfamily:* Rhodopsin-like receptors and pumps (7 families) CL0192
- *Family:* Microbial and algal rhodopsins (46 proteins) 3.E.1 (TCDB) PF01036 PDBsum
- *Species:* Halobacterium sp. (3 proteins)
- *Localization:* Archaebacterial membrane (104 proteins)

1uaz >> Archaerhodopsin-1	
Hydrophobic Thickness or Depth	31.8 ± 1.3 Å
Tilt Angle	9 ± 2°
$\Delta G_{transfer}$	-65.3 kcal/mol
Links to 1uaz	PDB Sum, PDB, SCOP, MSD, MMDB, Encompass
Topology	subunit A (N terminus extracellular side)
Resolution	3.40
Primary PDB represention	1uaz
Other PDB entries representing this structure	none
Number of TM Secondary Structures	7
Membranome	none

Comments: Light-driven proton pump. It may interact with bacterioruberin in the claret membrane.



Download File: 1uaz.pdb
PDB Sum, PDB

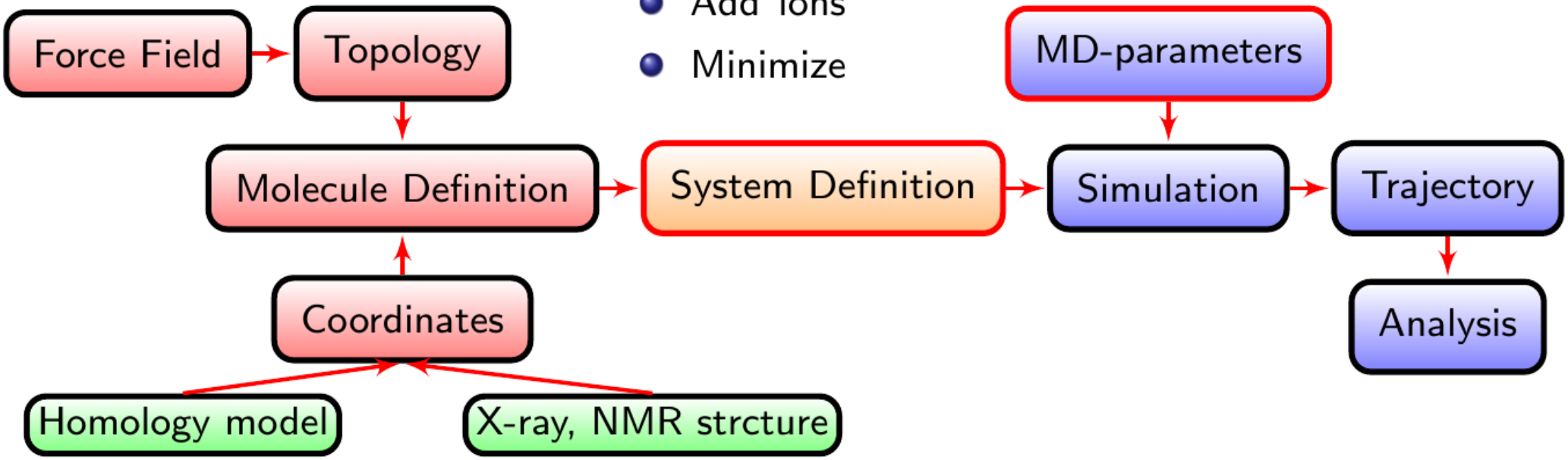
Topology in Archaebacterial membrane

3D view in GLMol or Jmol

Subunits: 1	
A - Tilt: 8 - TM segments: 1(15-38),2(48-68),3(86-104),4(111-133),5(138-160),6(179-197),7(207-230)	view assembly page

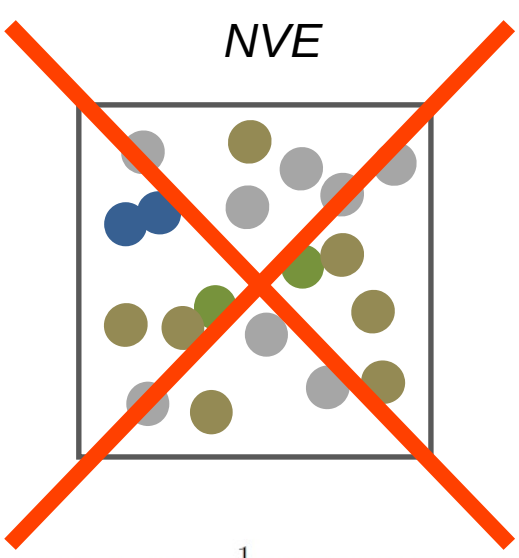
Stage: Defining MD parameters

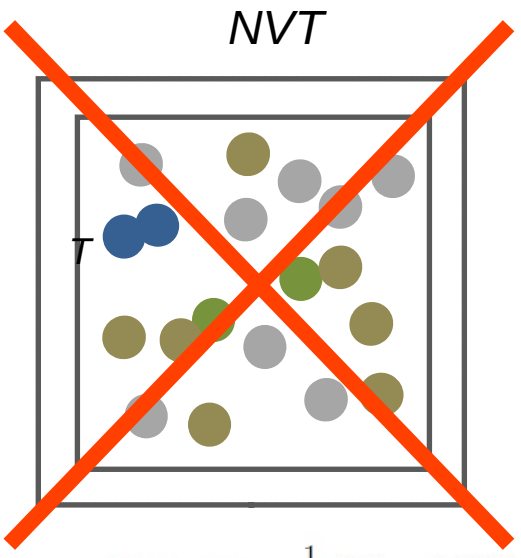
- Define unit cell
- Assemble the system
- Add solvent
- Add ions
- Minimize

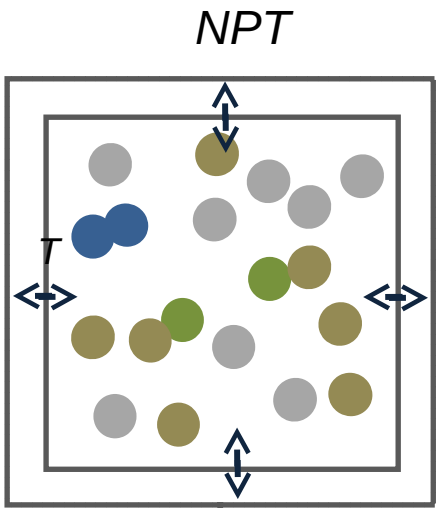


Choose the statistical ensemble for membrane simulation

- thermodynamic statistical ensembles describe macroscopic conditions
- *NVE* – microcanonical
- *NVT* – canonical (other names: isothermal, Helmholtz canonical)
- *μVT* – grand-canonical
- *NPE* – isobaric
- *NPT* – isobaric-isothermal (other name: Gibbs canonical)

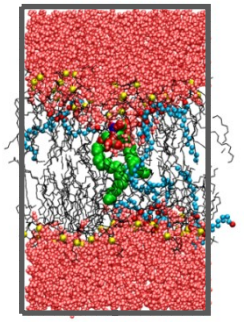






semi-isotropic

$p_x = p_y \neq p_z$



$$\Omega(N,V,E) = \frac{1}{N!}\sum_X \delta[\mathscr{H}(X) - E]$$

$$S(N,V,E) = k_B ln\Omega(N,V,E)$$

$$Z(N,V,T) = \frac{1}{N!}\sum_X e^{-\beta\mathscr{H}(X)}$$

$$F(N,V,T) = -k_B T ln Z(N,V,T)$$

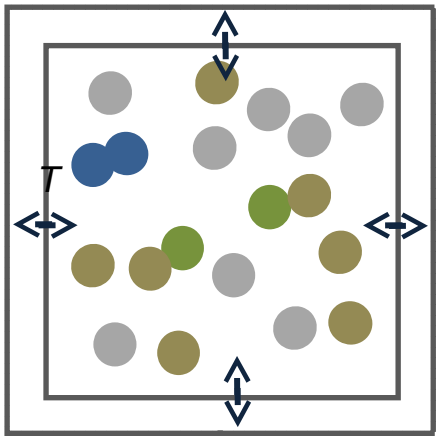
$$Q(N,P,T) = \int dV e^{-\beta PV}\frac{1}{N!}\sum_X e^{-\beta\mathscr{H}(X)}$$

$$G(N,P,T) = -k_B T ln Q(N,P,T)$$

NPT baths

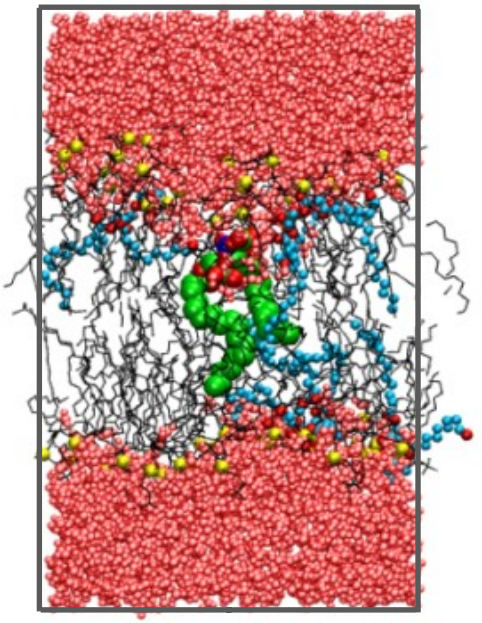
- Thermostats
  - Nosé-Hoover
  - Langevin
  - V-rescale
  - ~~Berendsen~~ (Do not use for production run !!)
- Barostats:
  - Parrinello-Rahman
  - c-rescale
  - ??Berendsen??

Temperature and pressure: Summary



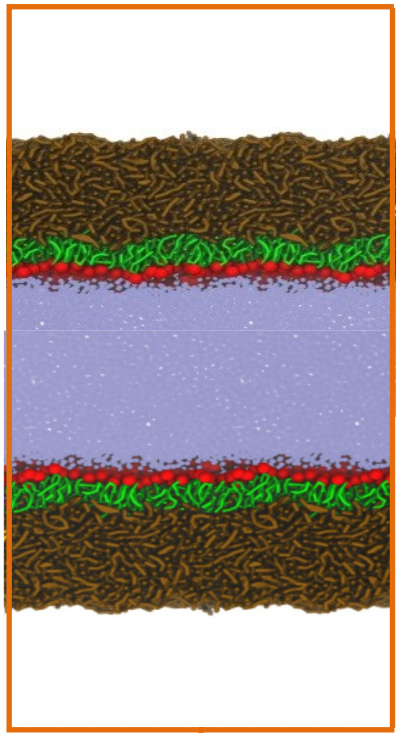
membrane

$p_x = p_y \neq p_z$



Monolayer

$p_x = p_y \neq p_z = 0$



- Use NPT ensemble with proper thermostat and barostat combination
  - e.g., p=1 atm, T=310 K
  - Use semi-isotropic barostat
- Few thermostats can be introduced
  - e.g., 1) membrane 2) protein 3) water
  - Separate groups have to be big enough

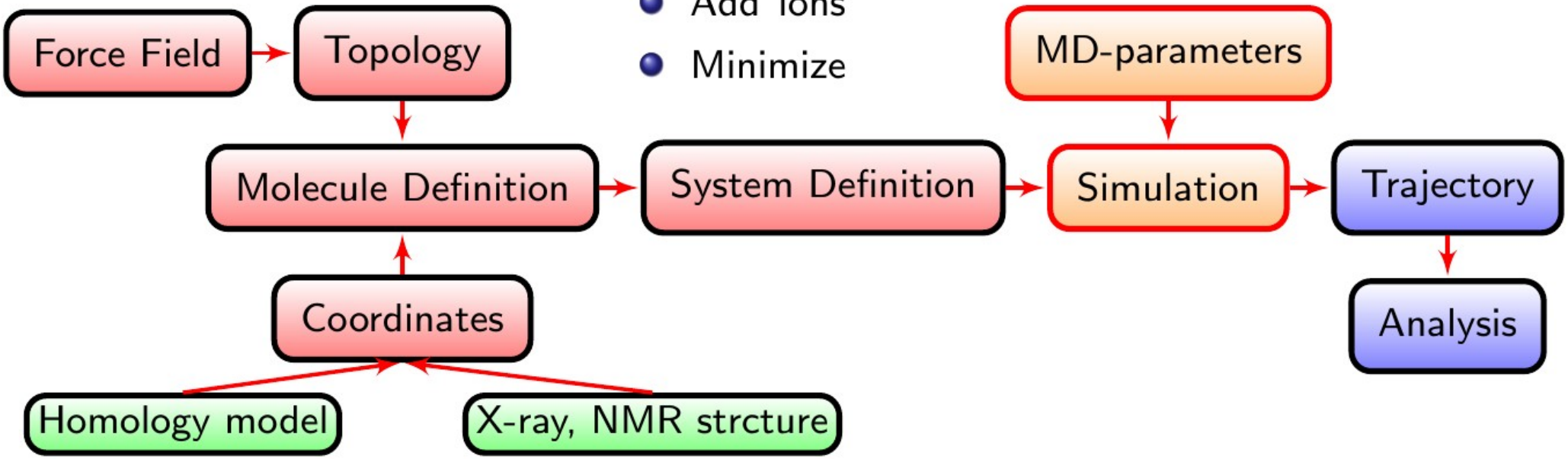
Choosing MD parameters

- Cutt-off schemes
- Time step
- PME

You CANNOT choose them. They are part of the force field

Stage: Simulation

- Define unit cell
- Assemble the system
- Add solvent
- Add ions
- Minimize



Simulation software

- Molecular dynamics (MD) simulation
  - GROMACS (www.gromacs.org)
  - Amber (www.ambermd.org)
  - CHARMM (www.charmm.org)
  - NAMD (www.ks.uiuc.edu/Research/namd)
  - OpenMM (http://openmm.org/)
  - LAMMPS (https://lammps.sandia.gov/)
- **MD software are like religions, better belong to many.**
  - Then you can pick the best of each
  - Communities hardly mix

Open science

Open source is not an option; it is a MUST.

What you cannot see, you cannot comprehend

Software considerations

- Use always the last stable version:
  - At least check if there are bugs affecting you?
- If your results change between versions, it is likely that you have hit a bug. At least is worth researching
- Using black boxes usually leads to one place. Wrong data
  - Close source codes :(
- Expensive codes?
  - Why to learn something you won’t have in the future

Perform the simulations

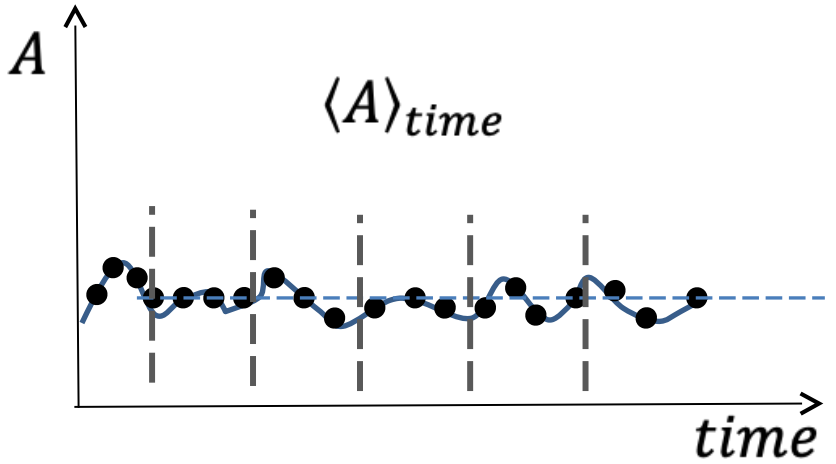
- Equilibration time
- Long enough
- Replicas (Ergodicity)

- What can I do when resources are limited? My answer:
  - Search for the required resources
  - Don’t do it

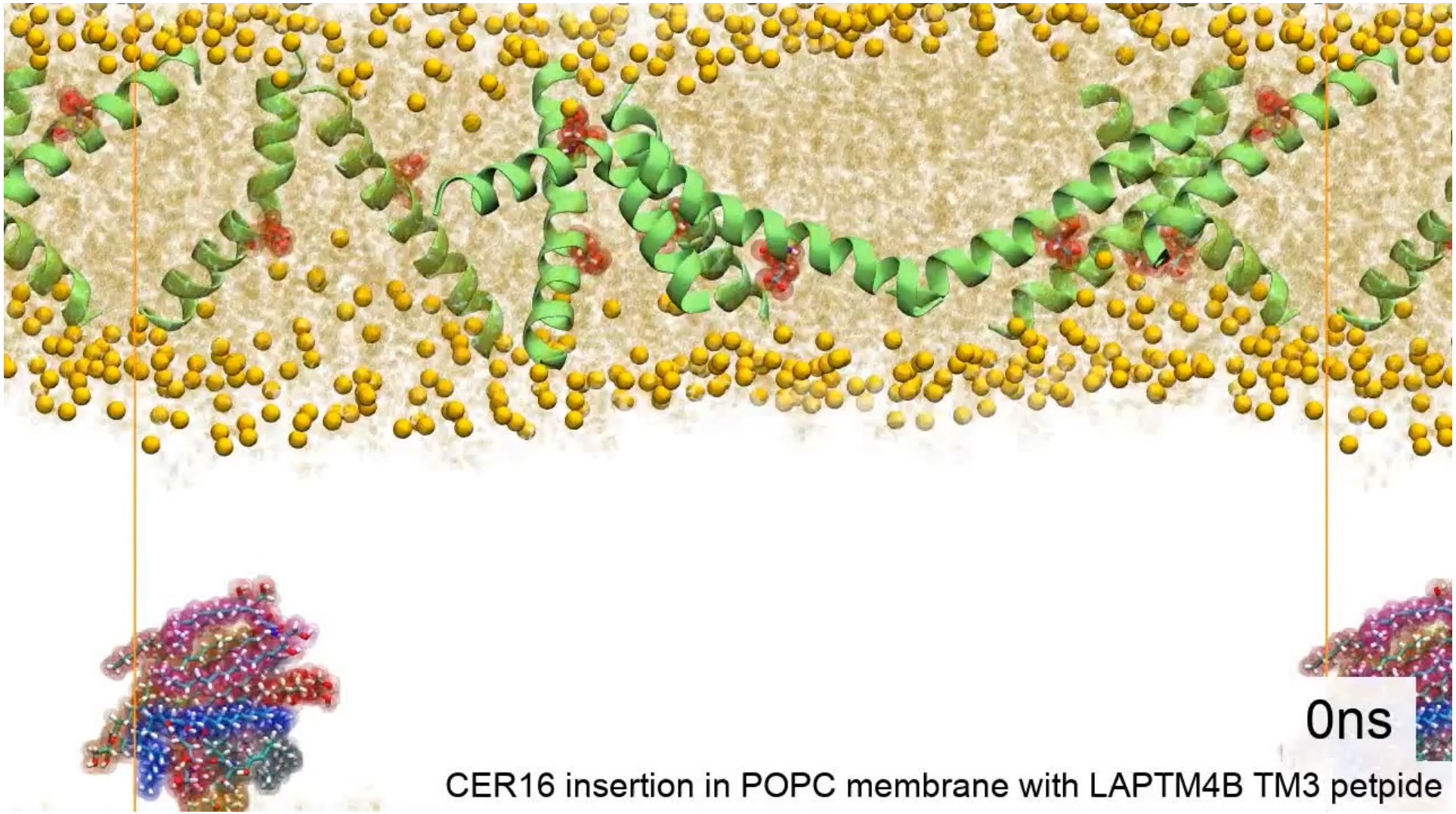
- For people of Finland you should be thankful of having CSC.
- For the others: http://www.hpc-europa.eu/

Have you simulated long enough? Likely not

- You need to sample enough. If not do not bother to make the simulation
- Make replicas to be sure that your simulation is indeed converged
- Protein membrane dynamics is slow
  - One simulation of 1 microsecond not enough

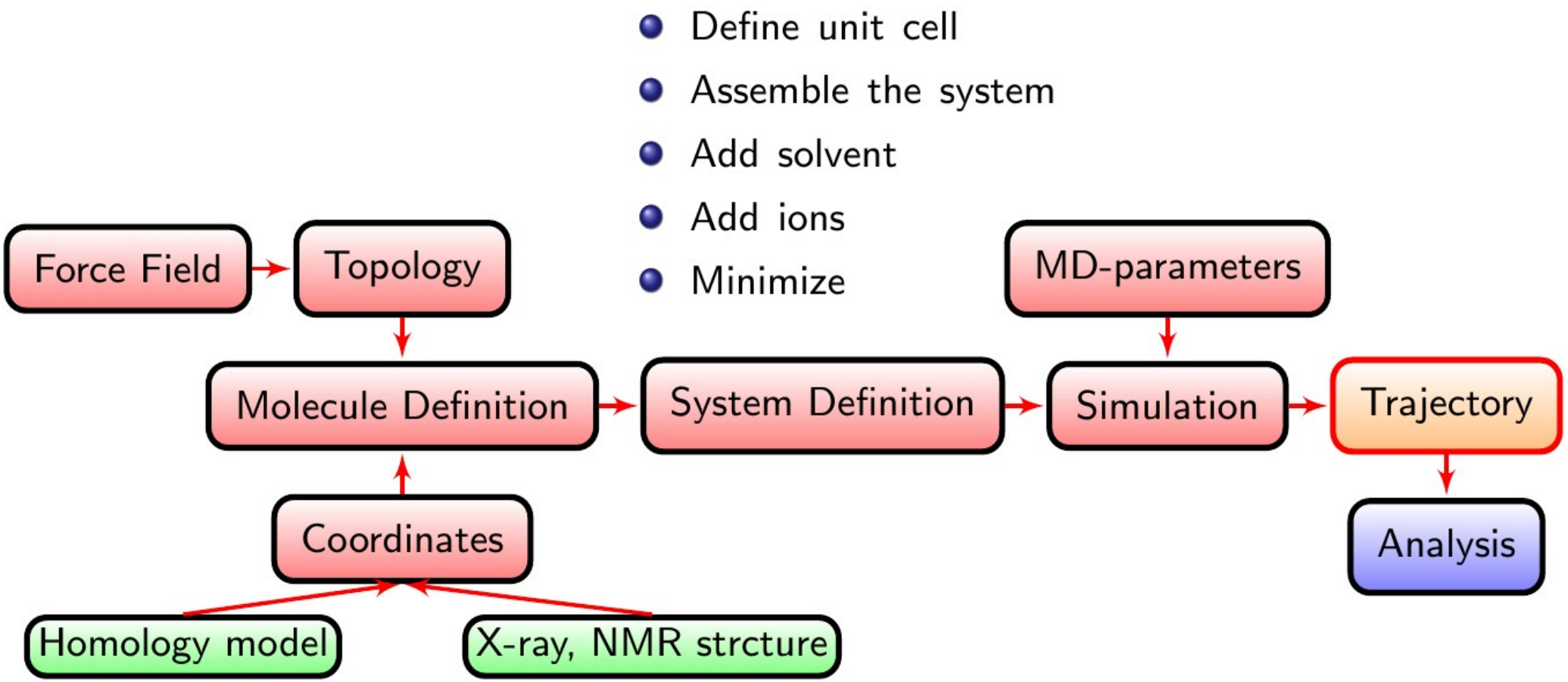


MD simulations



CER16 insertion in POPC membrane with LAPTM4B TM3 petpide

Stage: Validation and Analysis



Validate your results

- Always compare your results with experiments.
  - It is tedious but we do science not science fiction.
- Ask someone else to check your simulation files.
  - We are humans, aka, we make errors.
- There is not a perfect force field
  - That you get something in one force field does not make it true
- Use a couple of force fields.
  - They are usually parametrized independently so if they share the same results, it might be that is some sense in the underlying physics

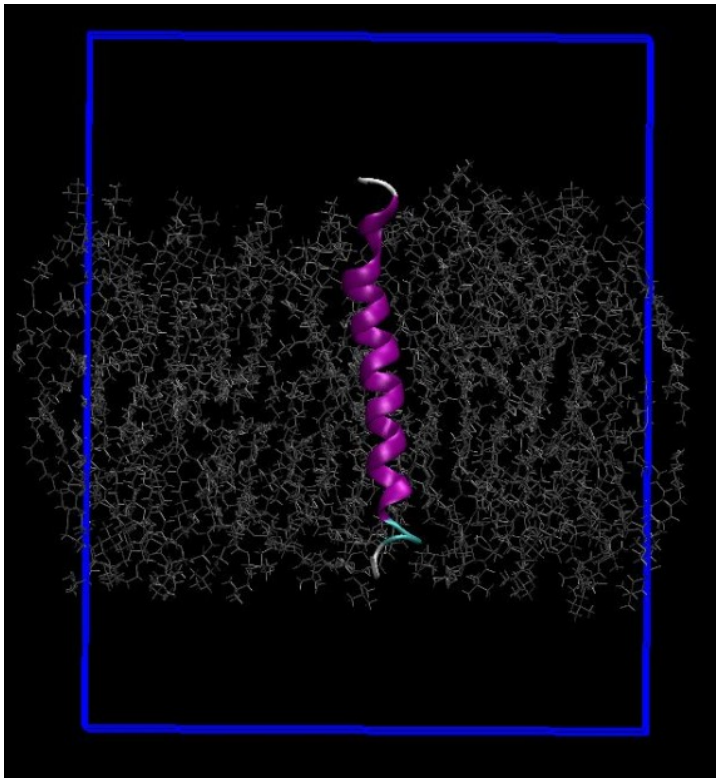
Analysis

Any data you will obtain by analyzing a simulated system, no matter how sophisticated analysis method is, will only be as good as the simulation allows.

The “ in” “ out” principle

Visual analysis of MD simulations

- employs visualization software (e.g., VMD)
- basic properties of the system can be qualitatively analyzed
- useful for simulation monitoring but also as a basis for further detailed numerical analysis
- fast, powerful, often underrated



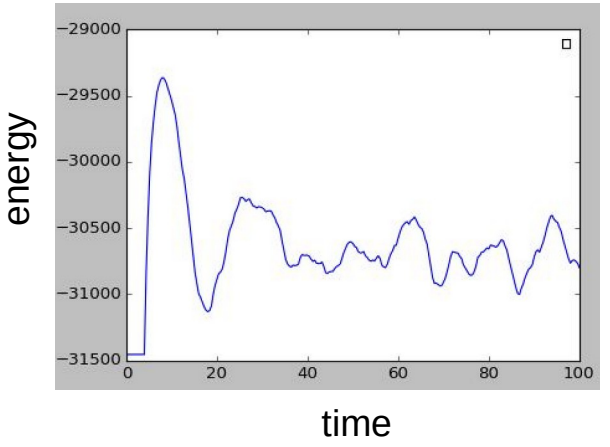
Example: orientation of peptide in lipid membrane

Numerical analysis of MD simulations

MD only provides instantaneous positions, velocities, forces on each atom at each time step. Also box size.

Typically analyzed quantities:

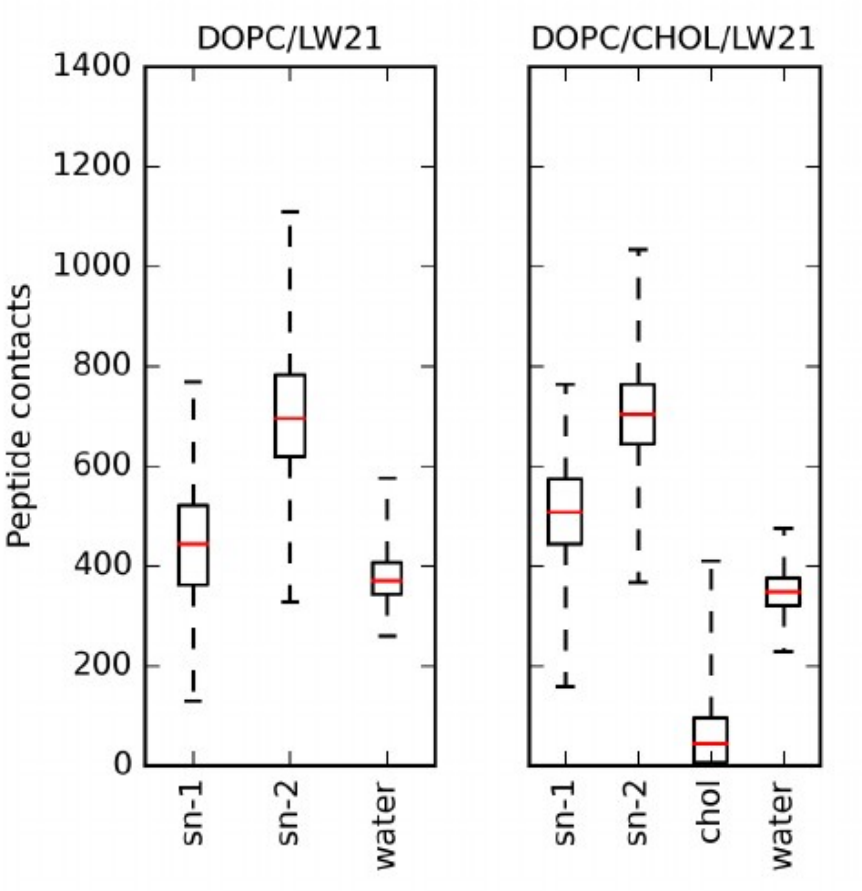
- macroscopic properties: temperature, pressure, volume, energy, polarization, surface tension
- microscopic properties: atomic/molecular ordering, distribution, conformations, hydrogen bonding

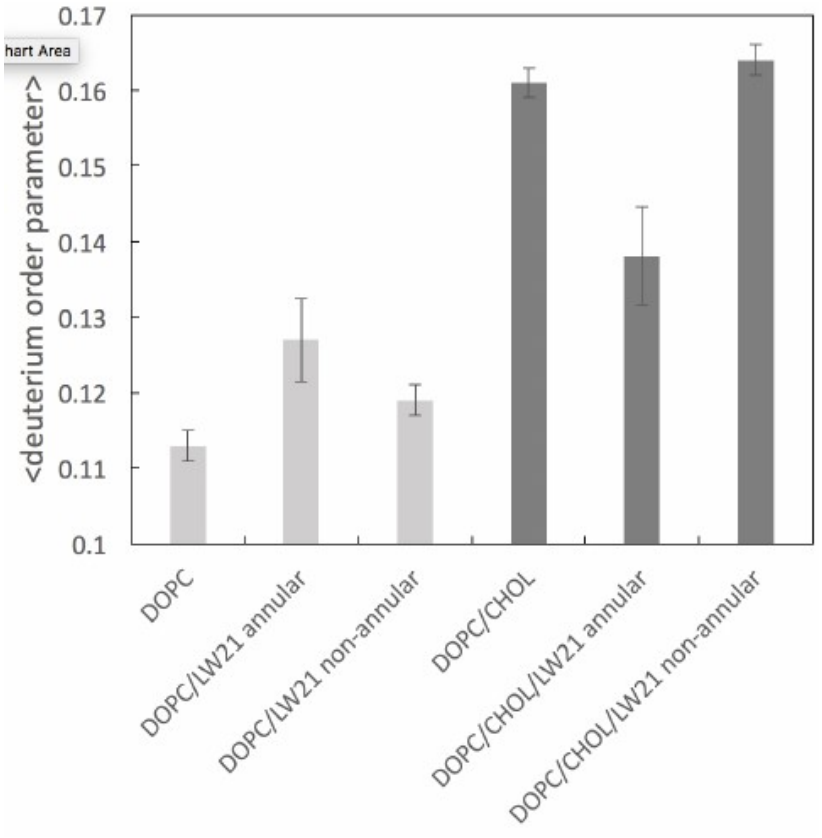


These quantities are either static (e.g., atomic distribution) or dynamics (e.g., diffusion coefficient)

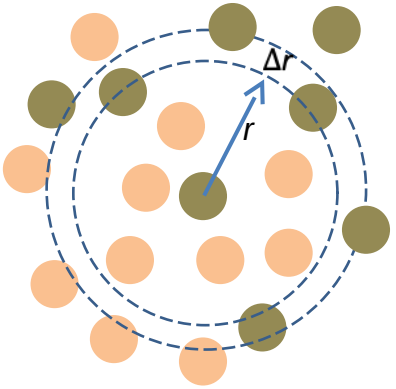
Numerical analysis of MD simulations

Example: contacts of membrane groups with transmembrane helix, and deuterium order parameter of lipid tails





Radial distribution function (RDF, g(r))



$$g_{AB}(r) = \frac{\langle \rho_B(r) \rangle}{\langle \rho_B \rangle_{local}}$$

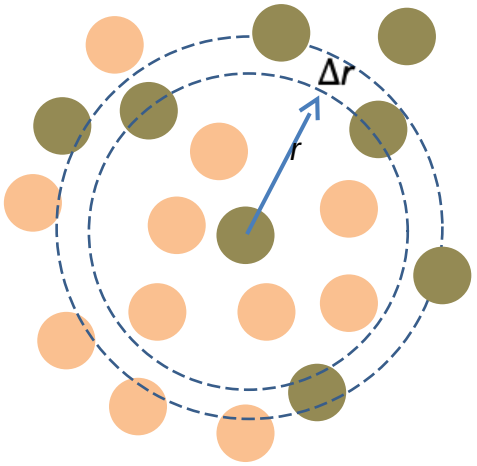
$$= \frac{1}{\langle \rho_B \rangle_{local}} \frac{1}{N_A} \sum_{i \in A}^{N_A} \sum_{j \in B}^{N_B} \frac{\delta(r_{ij} - r)}{4\pi r^2}$$

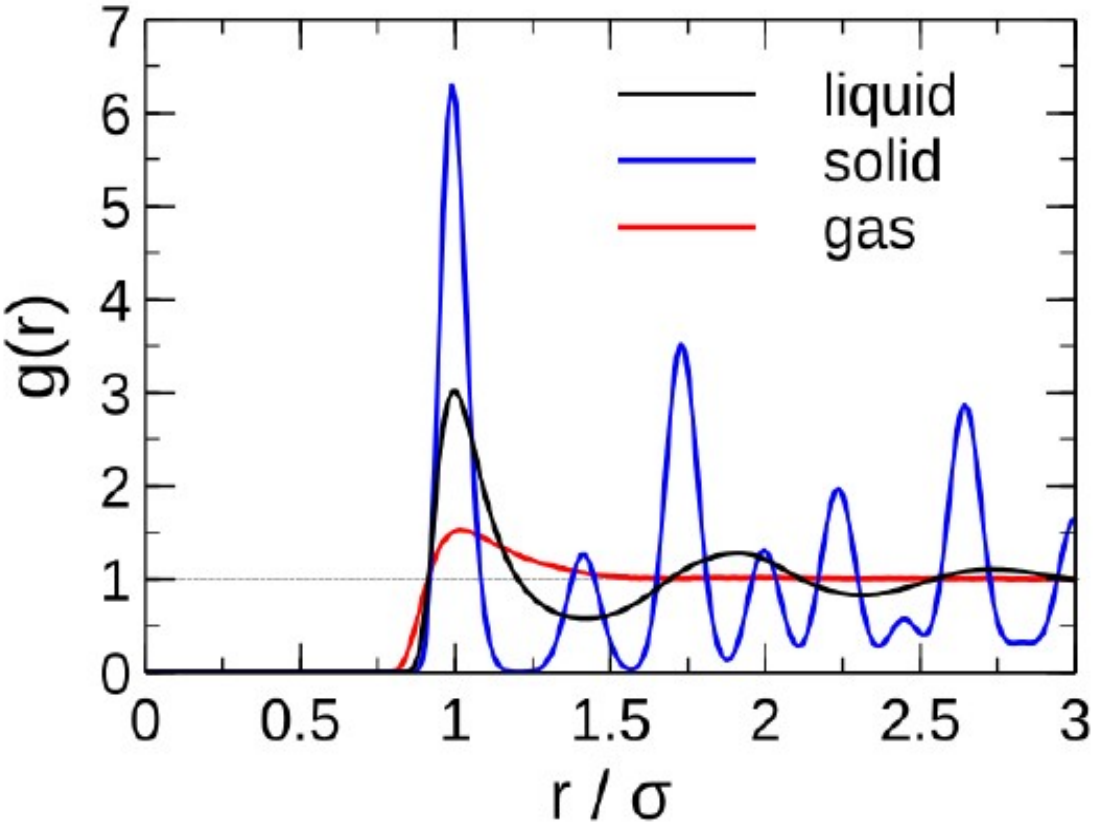
- a measure of the probability of finding a particle at a distance of r away from a given reference particle, relative to that for an ideal gas

- RDF is related (via Fourier transform) with so-called structure factor (S(q)) which can be determined experimentally via X-ray diffraction or neutron diffraction

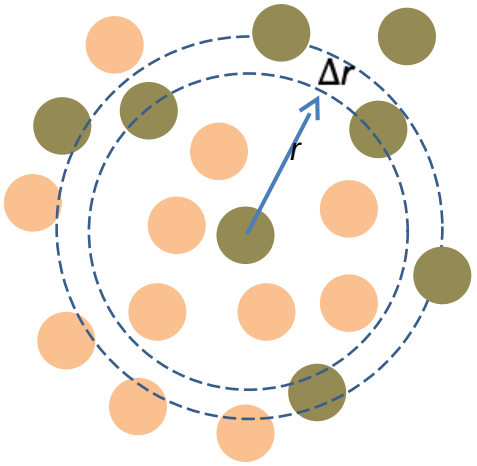
$$S(\mathbf{q}) = 1 + \rho \int_V \mathrm{d}\mathbf{r}\, \mathrm{e}^{-i\mathbf{q}\mathbf{r}} g(\mathbf{r})$$

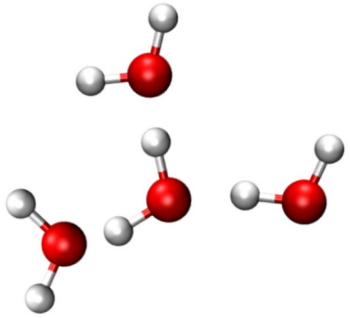
RDF example 1: ideal lattice (crystal)

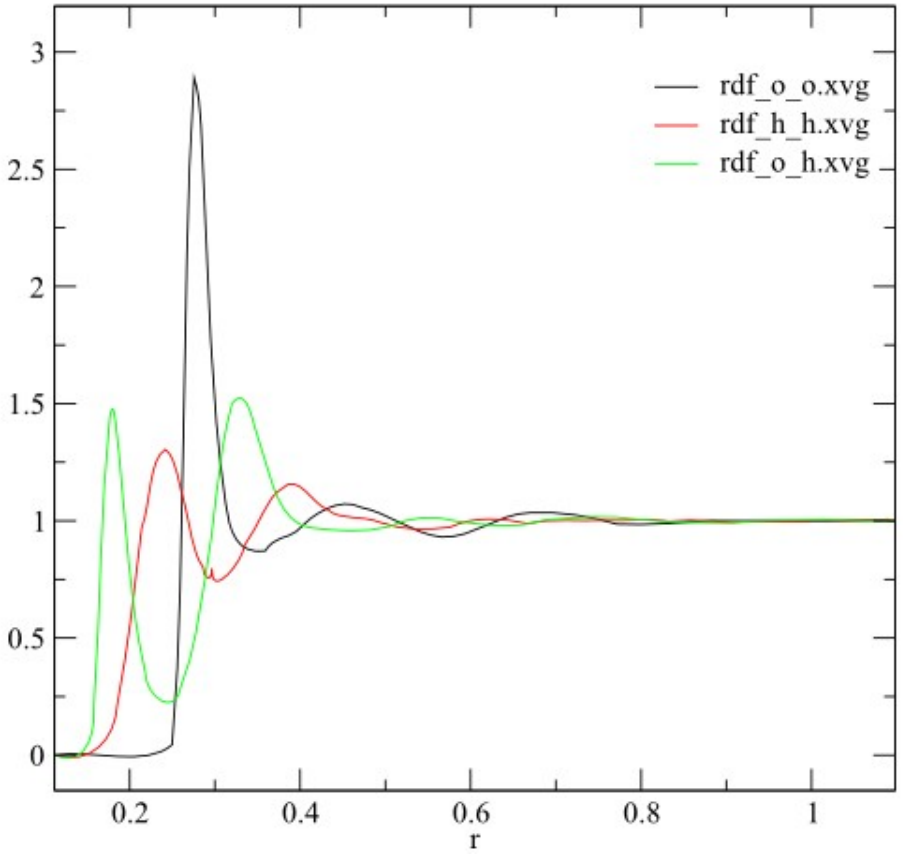




RDF example 2: water

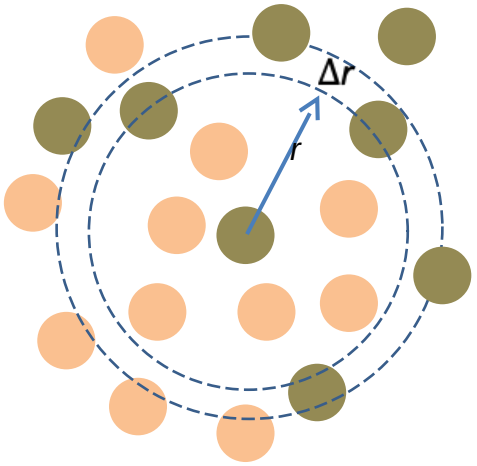




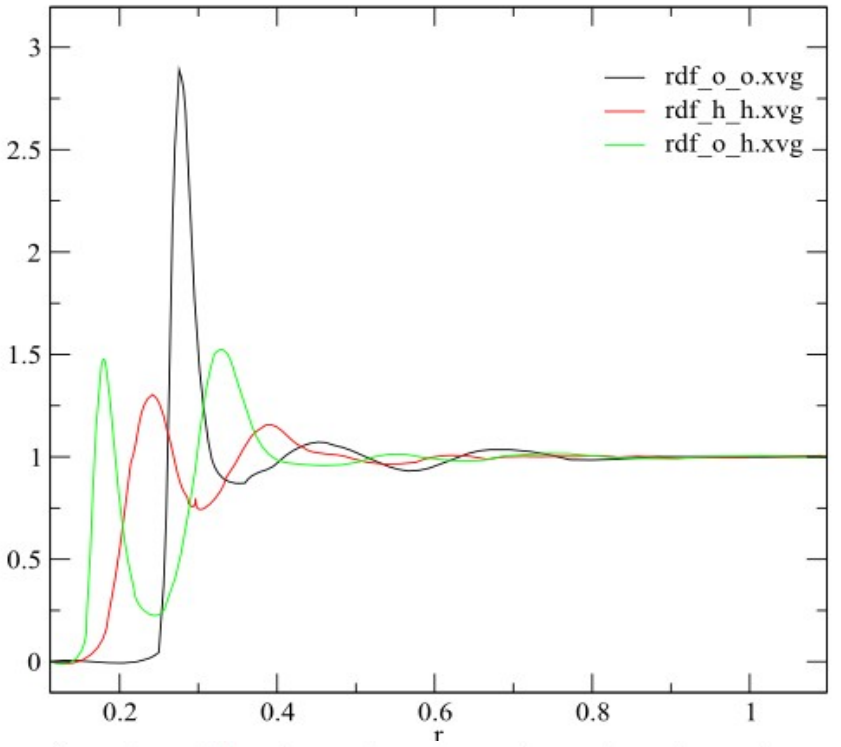


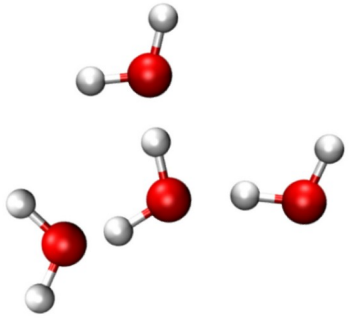
- hydration structure analysis
- comparison with experiment

RDF example 2: water

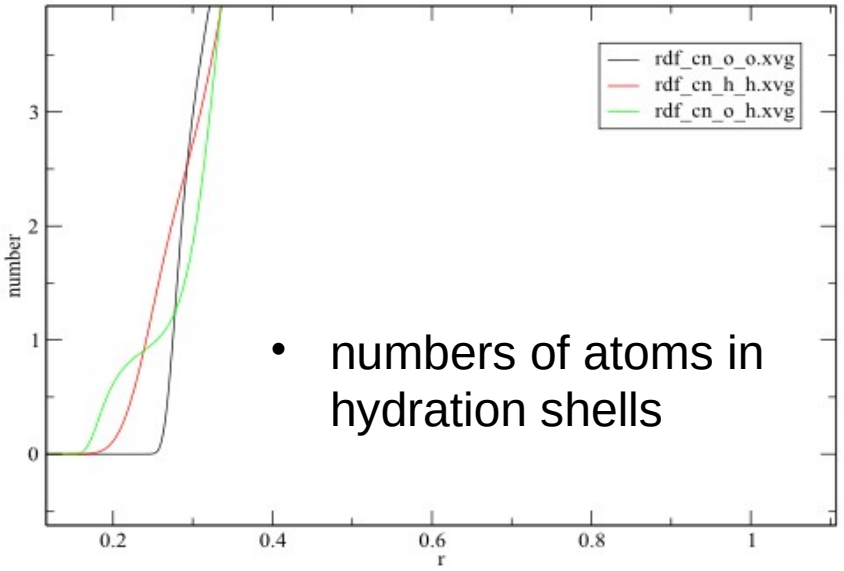


RDF



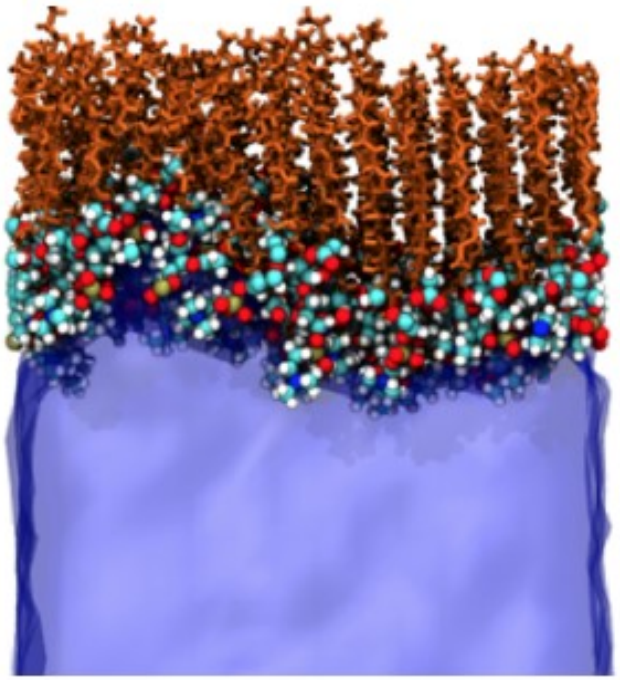


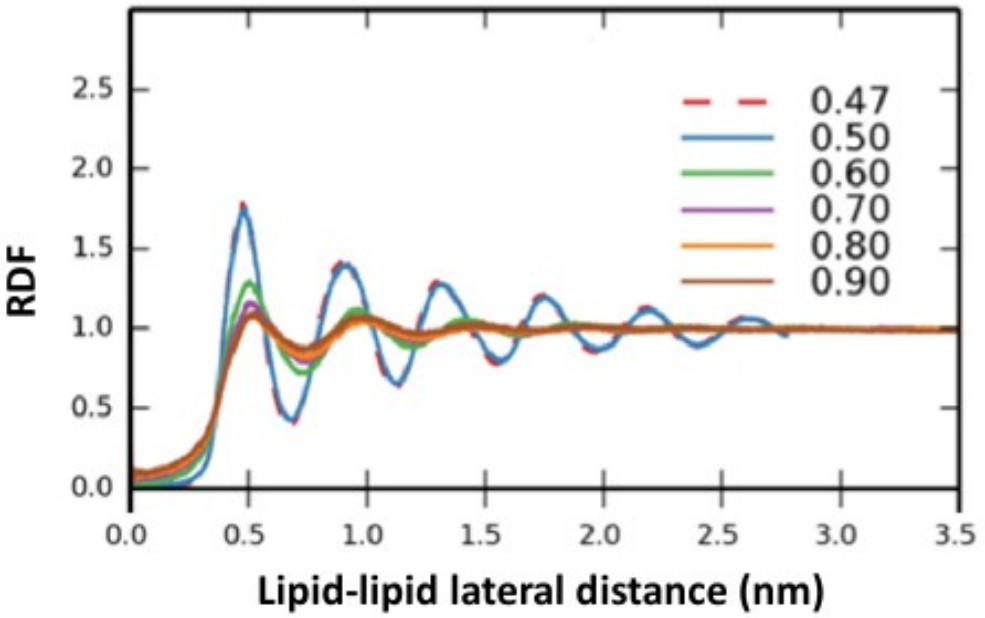
cumulative sum

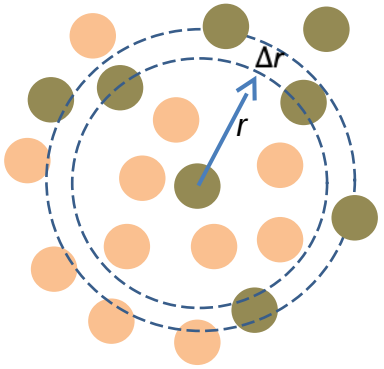


- numbers of atoms in hydration shells

RDF example 3: Lipid Monolayer





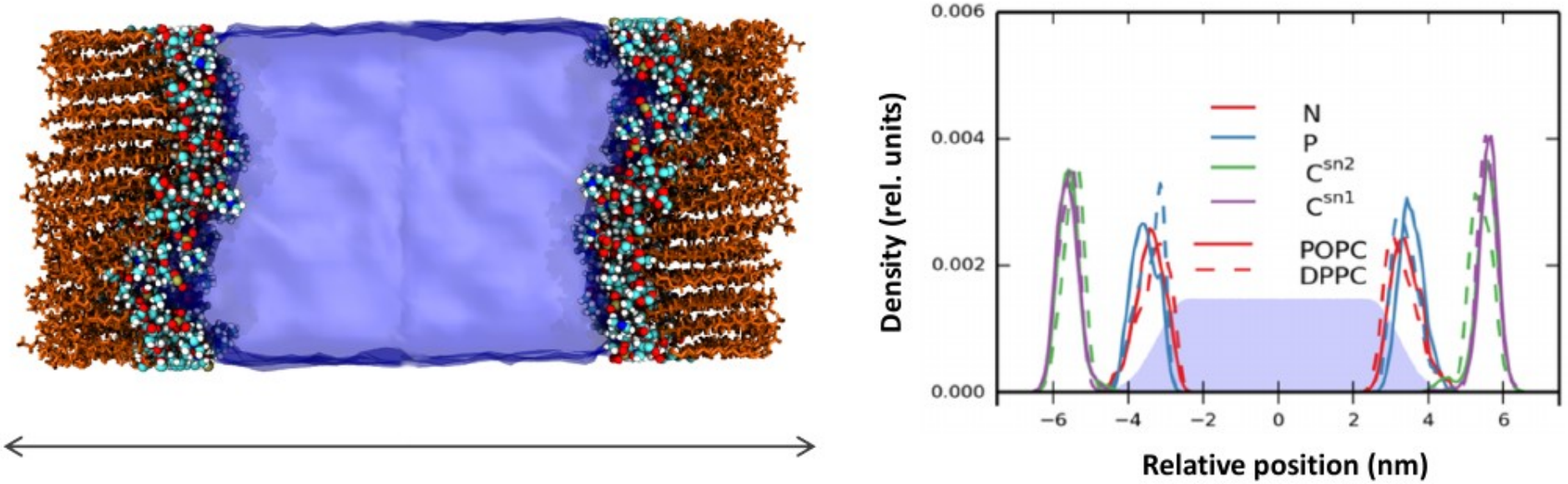


- 2D RDF for studying lateral arrangement of molecules
- phase transition in monolayer can be analyzed

Density profiles

density of selected atoms, groups of atoms, molecules as a function of position

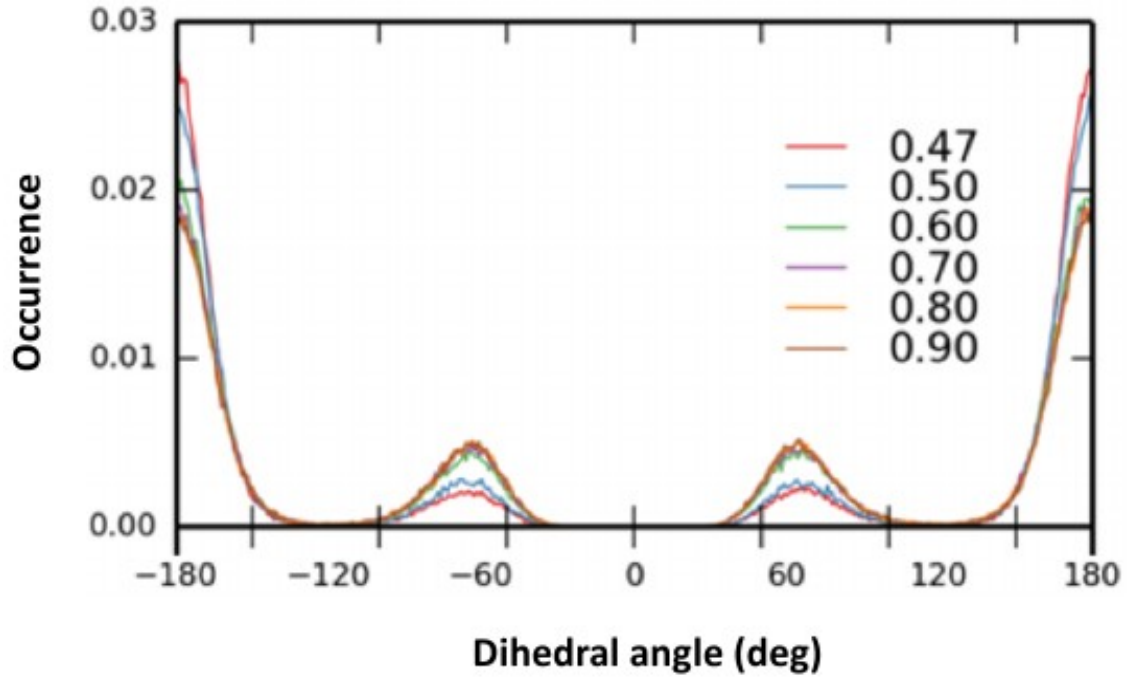
Example 1: lipid Langmuir film
calculated in gromacs using gmx density



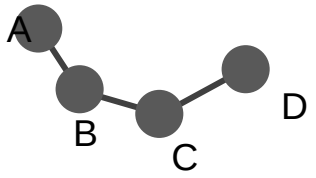
- typical units: kg/m^3, number density (nm^{-3}), electron density, arbitrary unit
- density profile can provide structural information but also for orientation of molecules (e.g., at interfaces)

Changes on the molecular properties due to the environment

Probability distribution of bond, angle, dihedral value
(It does not have to be the same as in the potential provided)



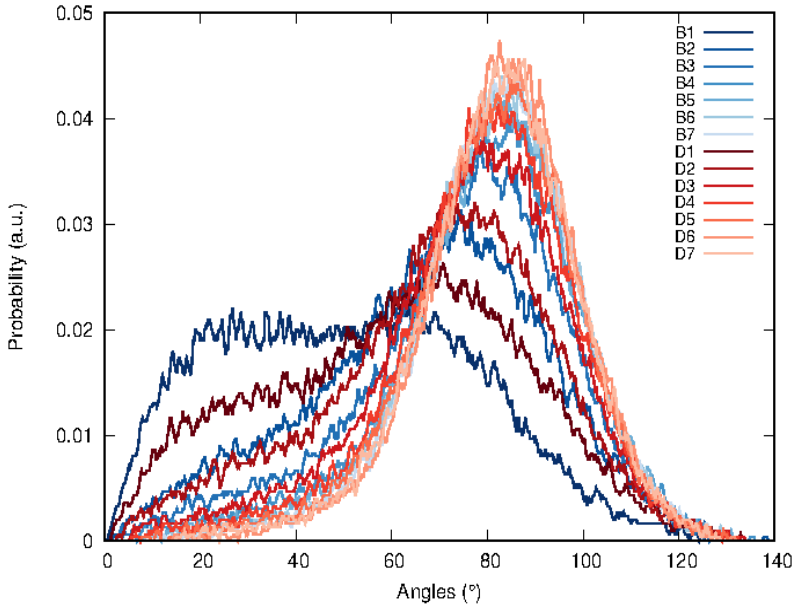
$P(r_{AB})$
$P(\alpha_{ABC})$
$P(\theta_{ABCD})$



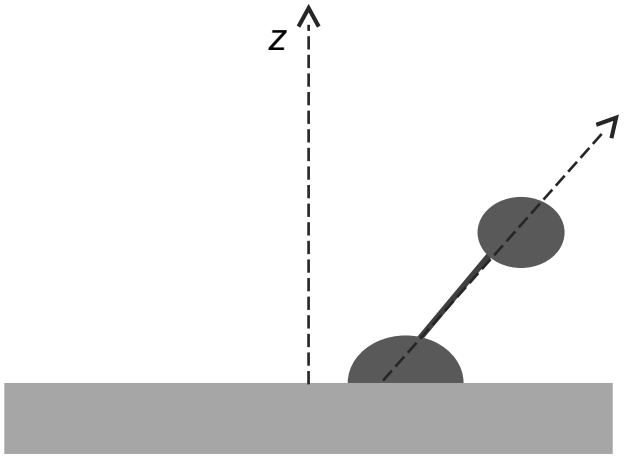
in Gromacs: gmx angle

- for analysis of average molecular conformations
- distances can be used for binding analysis (also time-dependent)
- other, indirect, phenomena can be captured (e.g., phase transitions)

Orientation distributions



$$P(\theta)$$

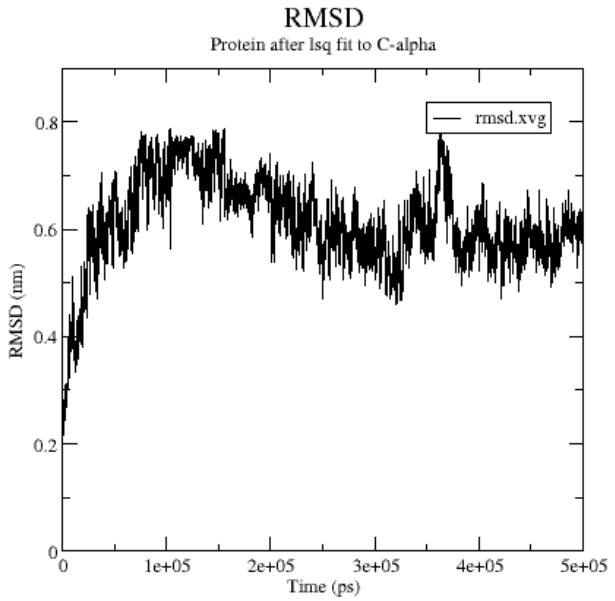


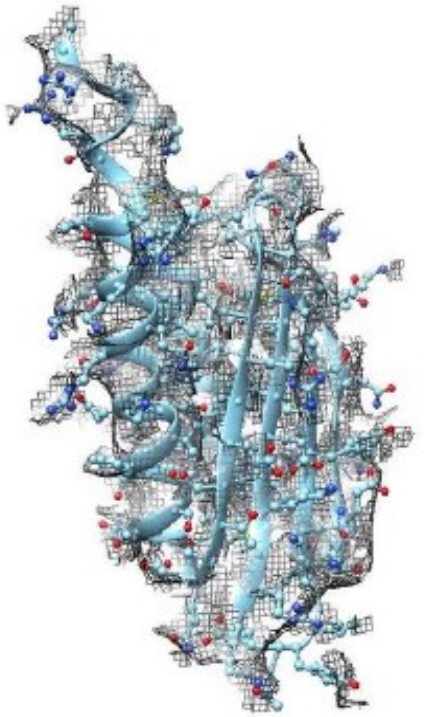
We can follow the orientation distributions or with time
(for example peptide orientation respect to bilayer normal)

Root mean square deviations (RMSD) in structure

$$RMSD(t_1, t_2) = \left[\frac{1}{M} \sum_{i=1}^{N} m_i \|\mathbf{r}_i(t_1) - \mathbf{r}_i(t_2)\|^2 \right]^{\frac{1}{2}}$$

change of structure in time

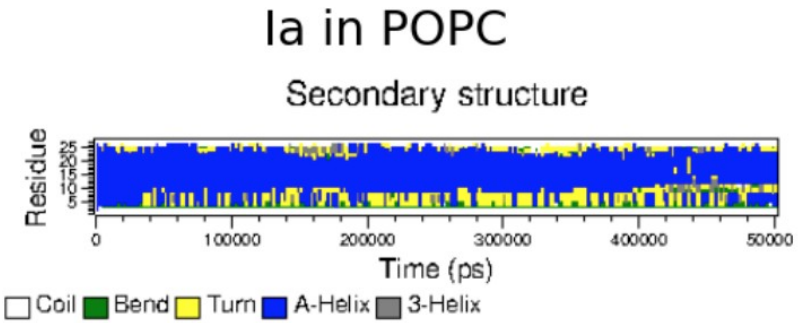




- usually used for proteins, monitoring time evolution of protein structure
- fitting may be done to t=0 or $t=t_i-\Delta t$
- fitting may be done to different structure than RMSD calculation (e.g., fitting to C-alpha, RMSD of whole structure)
- very good for equilibration monitoring!

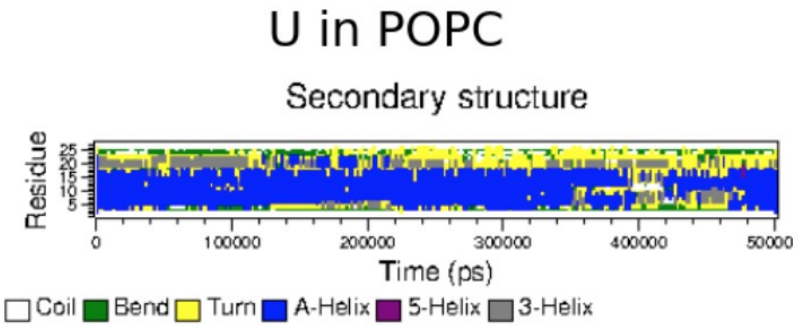
Secondary structure of proteins

time evolution of protein secondary structure



- only for proteins
- secondary structure with individual residues influence in time
- external software (DSSP) needed (free)
- very good for <u>equilibration monitoring</u>!

Riedlova & Cwiklik

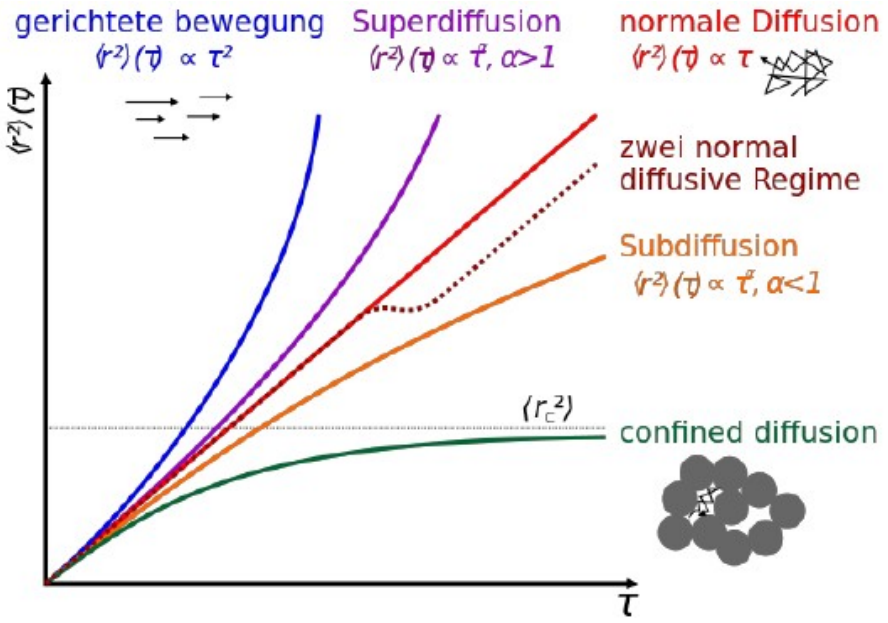


in Gromacs: gmx do_dssp

Molecular diffusion

Mean square displacement, diffusion coefficient

$$\lim_{t\to\infty} \langle \|\mathbf{r}_i(t) - \mathbf{r}_i(0)\|^2 \rangle_{i\in A} = 6D_A t$$



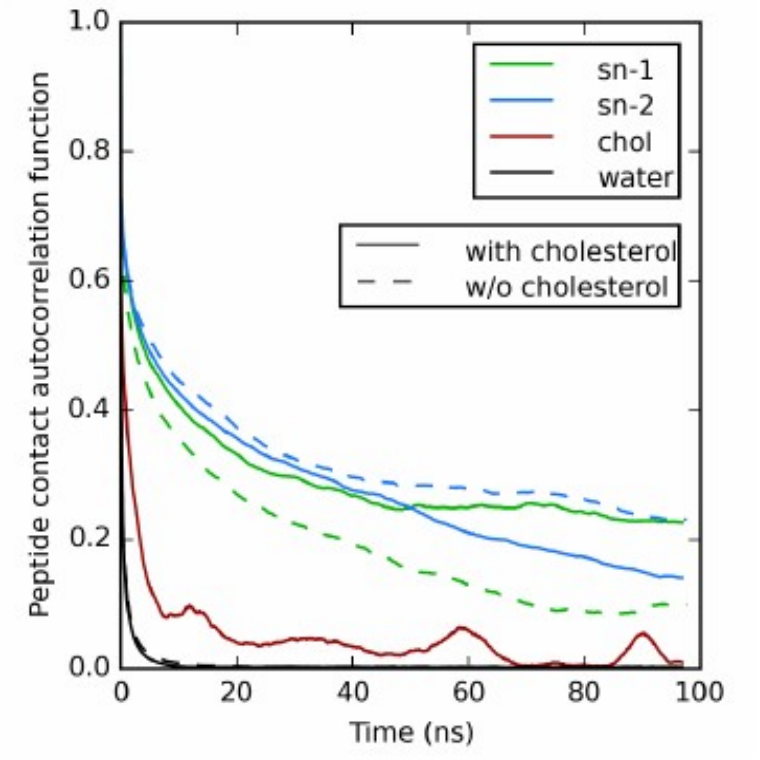
Jkrieger at Wikimedia Commons
CC BY-SA

in Gromacs: gmx msd

- diffusion coefficient calculated based on Fick's law (linearity of MSD assumed!)
- Problems: MSD non-linearity, center of mass movement during in simulation

Time correlation functions

time and memory characteristics
of a process



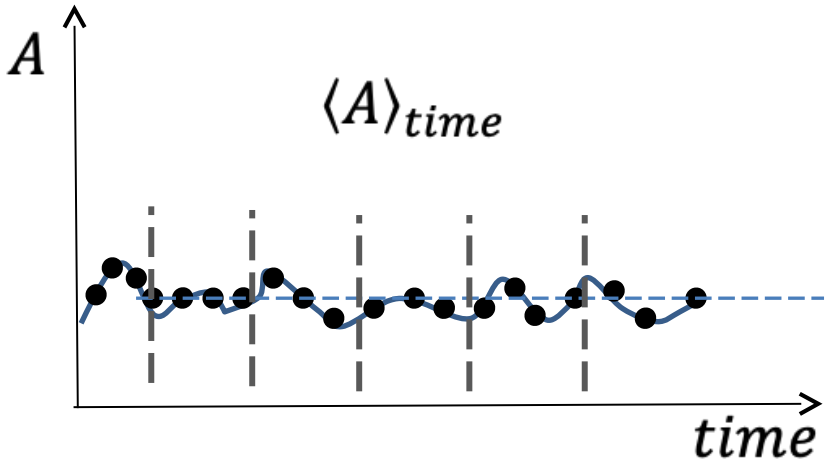
$$C_f(t) = \langle f(\xi) f(\xi + t) \rangle_\xi$$

averaged over time origins ξ

different variants: contact acf, velocity acf (for IR spectra), dipole acf (permittivity), pressure acf (for viscosity)

Error analysis: block averaging

- Statistics over thermodynamically fluctuating data is difficult
- Issues: fluctuations due to limited size of the system, time correlations, trajectory sampling etc.
- Good estimation of the error of a mean: block averaging (see Allen & Tildesley for more details)



in Gromacs:
gmx analyze -ee

The best review for membrane and membrane protein systems

Javanainen, M. & Martinez-Seara, H.

Efficient preparation and analysis of Membrane And Membrane Protein Systems

Biochimica et Biophysica Acta (BBA) - Biomembranes, Elsevier BV, 2016, 1858, 2468-2482