

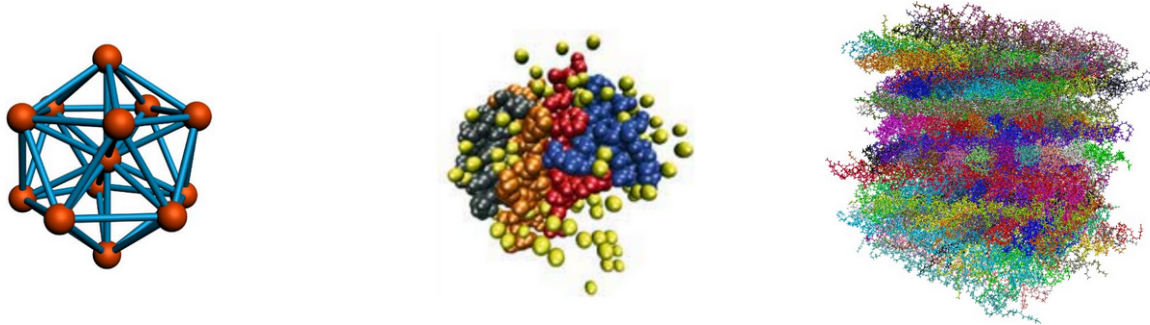
Molecular Dynamics Simulations

Outline:

- introduction
- physical and numerical basis of MD simulations
 - Newton's equations of motion (EOM)
 - velocity-Verlet time-integration scheme
- two-particle Interactions:
 - Lennard-Jones potential,
 - truncation
 - modified potential
- nondimensionalization of physical quantities
- improving performance:
 - linked-cell method
- basic statistical ensembles used in MD:
 - microcanonical
 - canonical
 - ergodic theorem
- thermalization:
 - velocity scaling, Berendsen method (~~Nose-Hoover thermostat~~)
- fluctuations and averages:
 - time correlations
 - ergodic theorem
 - pair-distribution function, radial density function
- Simulations of phase transitions in MD

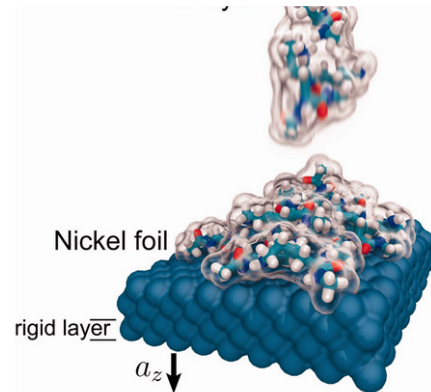
General information about MD

- Molecular Dynamics in its basics version is a computational method that allows for simulation of dynamics of group of interacting objects (particles, molecules, atoms, composite objects)

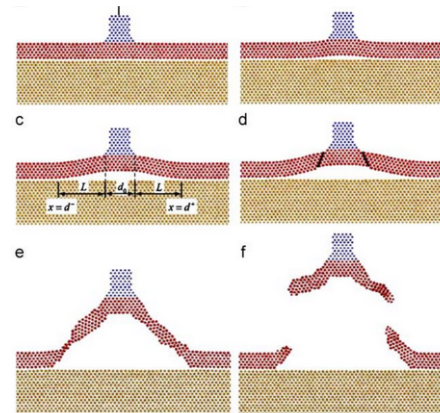


- description of dynamics in **classical version** of MD is based on time-integration of Newton's/Hamilton's equations of motions, the real forces are conservative and are derived from interparticle-interaction potentials as gradients, imposed external (single-particle) forces (electric, magnetic, gravitational, etc.) and some fictitious forces that helps us to manipulate the system in question as a whole (temperature) or enables the manipulation of its particular components (single particles, molecules, other objects)

J. Comput. Chem. 33, 2412–2439 (2012)



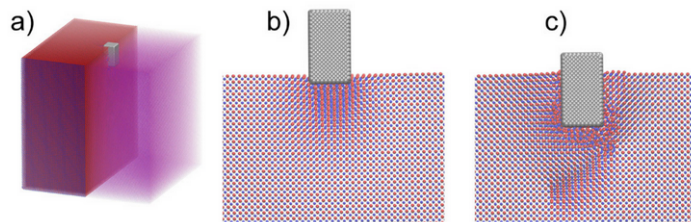
Int. J. Integr. Eng. 14 (2022) 137-159



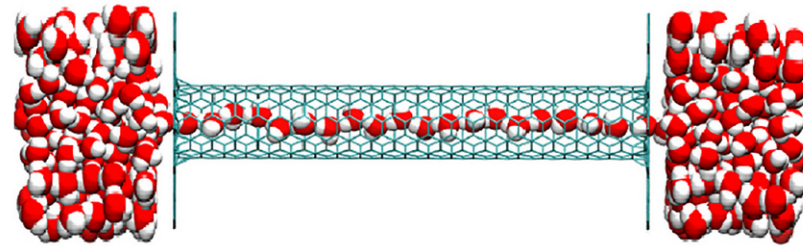
- more advanced modern versions of MD are combined with fast quantum mechanical methods (e.g. DFT) and allow for simulations of quantum dynamics of objects like atoms (nuclei+electrons), their impacts, chemical reactions

- generally the basic MD method is used in simulations in microscale, that is, on the length scale which ranges between a few angstrom to few hundreds nanometers for :
 - mass, heat and charge transport,
 - properties of gases, liquids and solids
 - atomic clusters, nanoparticles, biomolecules, nanostructures, composite materials, interfaces
 - material properties

J. Phys.: Conf. Series 438 (2013) 012021

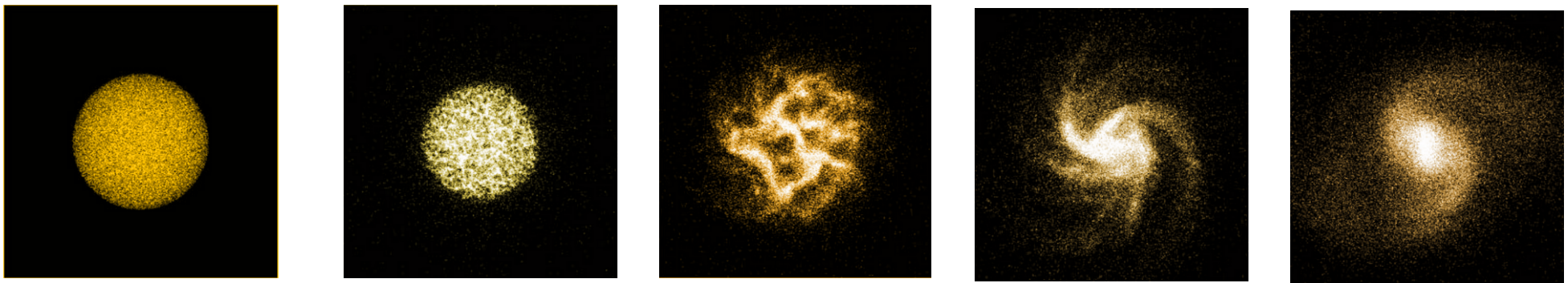


Comp. Mat. Sci. 185, (2020) 109978



- sometimes simple rescaling of interparticle interactions range brings MD method to macroscale, e.g. simulations of **evolution of galaxies**

„Numerical simulations in molecular dynamics” – Griebel, Knappek, Zumbush

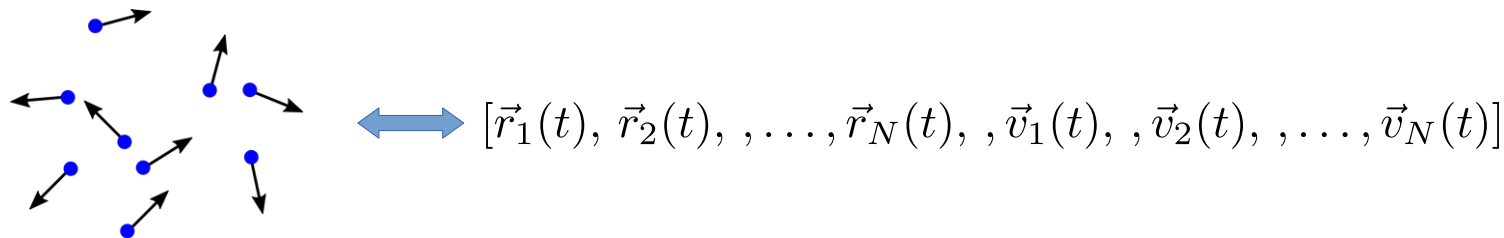


TIME

- most MD simulations contain few hundred to several thousands of objects, but advanced modelling of e.g. material properties requires millions of particles
- high-order time-integrators for ODE are required, particularly the total simulation time may require of making millions of time steps, time step shall be carefully chosen due to possible occurrence of strong and soft interactions as well as different time scales of simulated processes (spatial motions of substrates vs. kinetics of chemical reactions)

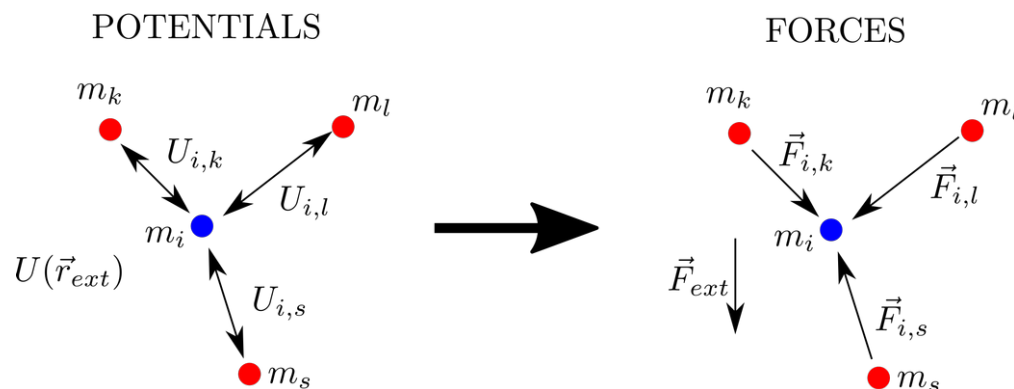
Physical and numerical basics of MD simulations

- our primary goal is to simulate the dynamics of system containing **N interacting particles**
- considering d-dimensional problem, we in fact are focusing on searching for 2dN element trajectory in phase space i.e. the position-velocity vector



- it is the minimal amount of data that defines uniquely the present state of the system, other information (**energy, temperature, forces**) are simply functions of this vector's elements
- MD algorithm begins with determination of interactions in the system, these are described within language of potential energy, from potential one derive the forces exerted on each particle

potential = potential energy



- MD uses model potentials which are derived and parametrized so as to give good approximation of real interactions (**but not exact ones**)
- model potentials always describes interactions in simplified way i.e. **very weak interaction are neglected due to their insignificance for dynamics**, other are parametrized with reduced number of required variables

Equations of Motion

- total potential (energy) is the sum of all single- and many-particle interactions (two-, three-body etc.)

$$U_{tot} = U_{tot}(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N)$$

$$U_{tot} = \underbrace{\sum_i U^{(1)}(\vec{r}_i)}_{\text{single}} + \underbrace{\sum_i \sum_{j>i} U^{(2)}(\vec{r}_i, \vec{r}_j)}_{\text{two-body}} + \underbrace{\sum_i \sum_{j>i} \sum_{k>j>i} U^{(3)}(\vec{r}_i, \vec{r}_j, \vec{r}_k) + \dots}_{\text{higher order interactions}}$$

single-particle interactions: external conservative forces like gravitational, electric, magnetic (+ sometimes non-conservative frictional or drag forces)

two-particle and higher order interactions: conservative forces mostly of electrostatic nature

- conservative nature of potential allows to calculate the force exerted on i-th particle as its gradient, knowing the force one immediately can use the second Newton law to find the acceleration

$$\vec{F}_i = -\nabla_{\vec{r}_i} U_{tot} \quad \nabla_{\vec{r}_i} = \left[\frac{\partial}{\partial x_i}, \frac{\partial}{\partial y_i}, \frac{\partial}{\partial z_i} \right]$$

- Newton's second law

$$\vec{F}_i = -\nabla_{\vec{r}_i} \left[U_i^{(1)}(\vec{r}_i) + \sum_{j>i} U^{(2)}(\vec{r}_i, \vec{r}_j) + \dots \right] \longleftrightarrow \vec{a}_i = \frac{\vec{F}_i}{m_i}$$

Velocity-Verlet time integrator

- nonequilibrium forces drive the dynamics of particles, propagation of system of particles in time requires the use of good, high order time-integrators and the most popular is the **velocity-Verlet scheme** (second order method for both: position and velocity)

$$1) \quad \vec{v}_i \left(t + \frac{\Delta t}{2} \right) = \vec{v}_i(t) + \frac{\Delta t}{2} \vec{a}_i(t)$$

$$2) \quad \vec{r}_i(t + \Delta t) = \vec{r}_i(t) + \vec{v}_i \left(t + \frac{\Delta t}{2} \right) \Delta t + O(\Delta t^3)$$

$$3) \quad \text{compute new acceleration at new position: } \vec{a}_i(t + \Delta t)$$

$$4) \quad \vec{v}_i(t + \Delta t) = \vec{v}_i \left(t + \frac{\Delta t}{2} \right) + \frac{\Delta t}{2} \vec{a}_i(t + \Delta t) + O(\Delta t^3)$$

Other integrators are also possible, especially symplectic ones are recommended, but they are used occasionally rather as part of professional/commercial packages.

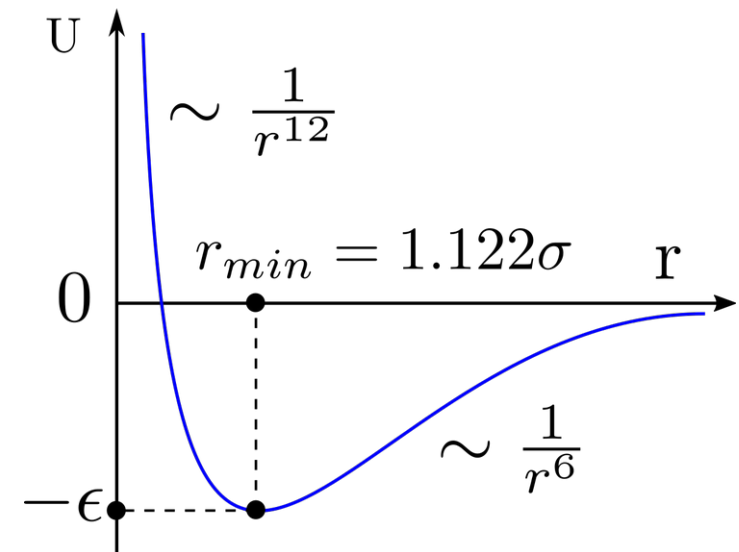
Example pair-wise potential: Lennard-Jones potential

The simplest two-particle potential describing interaction of two atoms with closed electron shell is the **Lennard-Jones potential**

$$r \equiv r_{ij} \implies U_{LJ}(r) = 4\epsilon \left(\frac{\sigma^{12}}{r^{12}} - \frac{\sigma^6}{r^6} \right)$$

ϵ and σ are parameters depending on type of atoms,
 ϵ - defines the strength of interaction,
 σ - sets the length scale

- the first term in bracket describes strong repulsion of two atoms when these are close to each other and their electron densities are being deformed
- second term is the electron density polarization effect responsible for mutual attraction of atoms at larger distance (van der Waals interaction)
- LJ potential belongs to class of short range potentials, for $r \sim 3-4 \sigma$ the interaction (force) its influence on dynamics becomes insignificant as oppose to long range Coulomb potential



	$\epsilon/k_B [K]$	$\sigma [nm]$
<i>He</i>	10.2	0.228
<i>Ne</i>	47.0	0.272
<i>Ar</i>	119.0	0.341
<i>Kr</i>	164.0	0.383

Other types of pairwise potential can be found in book chap. 2.3

"Multiscale modeling of complex molecular structure and dynamics with MBN Explorer", Springer,

Truncation of Lennard-Jones potential

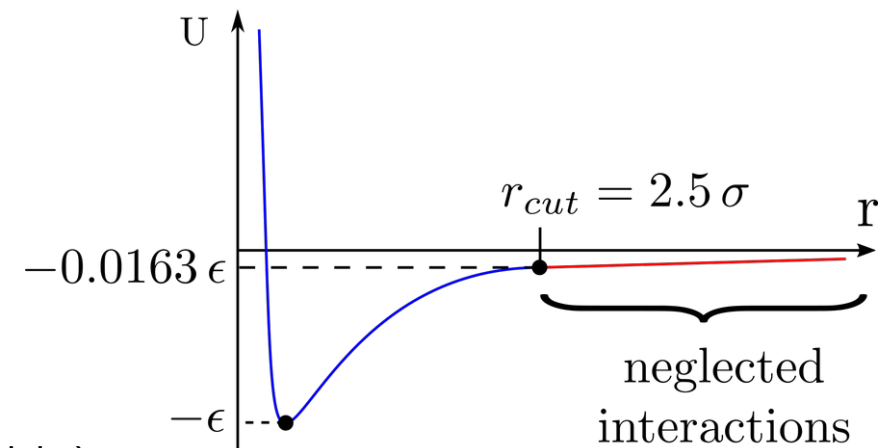
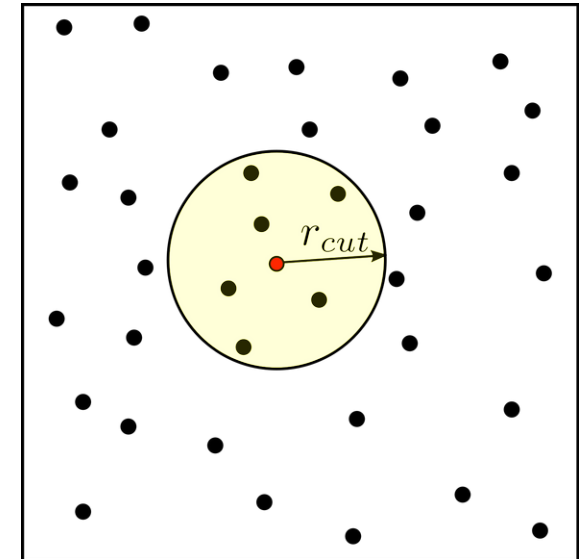
- the short range nature of LJ potential suggests that the given i -th particle strongly interacts only with particles localized in its immediate area while the interactions with more distant ones can be neglected,

usually the cut-off range is assumed at value

$$r_{cut} = 2.5\sigma$$

$$U_{LJ}(r_{cut}) = -0.0163 \epsilon$$

- the use of cut-off potential increases performance of MD method (part of computations are skipped), but it perturbs the results for the following reasons
 - cut-off makes the potential non-differentiable at r_{cut} point (differentiation of step function gives the Dirac delta)
 - even the vanishing forces from distant atoms can not change the dynamics of particle, when neglecting their contributions to total potential, this potential becomes higher than it should be



- the potential discontinuity issue is resolved by improving the tail of LJ potential so as to make it smoothly vanishing exactly at r_{cut}
- example modifications for truncated LJ potential
 - example I** (polynomial scaling)

$$\tilde{U}_{LJ}(r) = S(r) \cdot 4\epsilon \left(\frac{\sigma^{12}}{r^{12}} - \frac{\sigma^6}{r^6} \right)$$

$$S(r) = \begin{cases} 1 & \Longleftrightarrow r \leq r_{cut} \\ 1 - \frac{(r-r_l)^2(3r_{cut}-r_l-2r)}{(r_{cut}-r_l)^3} & \Longleftrightarrow r_l < r < r_{cut} \\ 0 & \Longleftrightarrow r > r_{cut} \end{cases}$$

$$r_l = r_{cut} - \Delta\sigma, \quad \Delta\sigma \approx 0.2 - 0.4$$

- example II** (subtraction of derivative)

$$\tilde{U}_{LJ} = U_{LJ}(r) - U_{LJ}(r_{cut}) - \left. \frac{\partial U_{LJ}(r)}{\partial r} \right|_{r=r_{cut}} \cdot (r - r_{cut})$$

- the amendment to total energy resulting from neglecting distant atoms interactions is taken into account by calculating the **Ewald sum**, this method exploits the Fast Fourier Transform in calculations of interaction with the replicas of the system periodically repeated in space – due to limited time designated for the lectures the details of this method are left untouched.

Nondimensionalization of equations

- making numerical calculations using SI units is not very comfortable due to very low scale (e.g. atoms) or large scale (e.g. galaxies) of the numbers one operates on
- to get rid of this inconvenience the physical quantities must be nondimensionalized before, then one may operate on dimensionless numbers

$$z' [-] = \frac{z [SI]}{z_{ref} [SI]} \implies z [SI] = z' [-] \cdot z_{ref} [SI]$$

$z [SI]$ – any physical quantity defined in SI unit

$z_{ref} [SI]$ – reference quantity of the same type in SI unit

$z' [-]$ – dimensionless quantity used in MD

- length and energy (potential)**

$$U_{LJ}(r) = 4\epsilon \left(\frac{\sigma^{12}}{r^{12}} - \frac{\sigma^6}{r^6} \right)$$

structure of LJ potential suggests the following choice would be a good

$$r_{ref} = \sigma$$

$$E_{ref} = \epsilon$$

$$U'_{LJ}(r) = \frac{U_{LJ}}{E_{ref}} = 4 \frac{\epsilon}{\epsilon} \left(\frac{\sigma^{12}}{\sigma^{12} r'^{12}} - \frac{\sigma^6}{\sigma^6 r'^6} \right) = 4 \left(\frac{1}{r'^{12}} - \frac{1}{r'^6} \right)$$

- **temperature, Boltzmann constant k_b**

We know the reference energy so it can help to find the temperature reference, both are related through thermal energy

$$E = k_b T \quad \Longrightarrow \quad E_{ref} = k_b T_{ref} \quad \Longrightarrow \quad T_{ref} = \frac{E_{ref}}{k_b} = \frac{\epsilon}{k_b}$$

Remark: T_{ref} depends inversely on Boltzmann constant k_b , hence it eliminates it from all dimensionalized equations

$$k_{b,ref} = k_b \quad \Longrightarrow \quad k'_b = \frac{k_b}{k_{b,ref}} = 1$$

- **mass**

If considered system contains atoms of only one type, its mass could be reference mass

$$m_{ref} = m \quad \Longrightarrow \quad m' = 1$$

- **velocity**

Here we can use the equipartition theorem to link velocity with other known reference quantities

$$E_{kin} = \frac{3}{2} k_b T \quad \Longrightarrow \quad \frac{mv^2}{2} = \frac{3}{2} k_b T \quad \Longrightarrow \quad \frac{m'v'^2}{2} [m_{ref} v_{ref}^2] = \frac{3}{2} k'_b T' [k_{b,ref} T_{ref}]$$

$$v_{ref} = \sqrt{\frac{k_{b,ref} T_{ref}}{m_{ref}}} = \sqrt{\frac{E_{ref}}{m_{ref}}}$$

- **time**

We have the reference length and velocity, from these we get reference unit of time

$$v_{ref} = \frac{r_{ref}}{t_{ref}} = \sqrt{\frac{m_{ref}\sigma_{ref}^2}{E_{ref}}}$$

- **force**

Here we can use Newton EOM

$$F = m \frac{d^2 r}{dt^2} \quad \Longrightarrow \quad F_{ref} = m_{ref} \frac{r_{ref}}{t_{ref}^2} = \frac{E_{ref}}{r_{ref}}$$

- **pressure**

From definition of pressure we get

$$p = \frac{F}{\Delta x \Delta y} \quad \Longrightarrow \quad p_{ref} = \frac{F_{ref}}{r_{ref}^2} = \frac{E_{ref}}{r_{ref}^3}$$

- Example unitless quantities for monoatomic system - summary

quantity	reference unit	unitless quantity
length	$r_{ref} = \sigma$	r'
energy	$E_{ref} = \epsilon$	E', U'
mass	$m_{ref} = m$	$m' = 1$
Boltzmann constant	k_b	$k'_b = 1$
time	$\sqrt{\frac{m_{ref} r_{ref}^2}{E_{ref}}}$	t'
velocity	$\sqrt{\frac{E_{ref}}{m_{ref}}}$	E', U'
force	$\frac{E_{ref}}{r_{ref}}$	F'
pressure	$\frac{E_{ref}}{r_{ref}^3}$	p'

Remarks:

- all quantities are expressed by three basics units: length, energy and mass
- for a mixture of atoms of different types these could be chosen differently

Optimization - Partitioning the system into cells

- the first step in optimizing the MD algorithm was to cut-off the interaction at some point and neglect interactions with remote atoms
- however, one must keep in mind that ending optimization at this stage still would require checking interparticle distance for all N^2 pairs
- such **brute-force searching** of interacting atomic pairs decreases performance of MD algorithm

- more efficient searching method requires division of the system on smaller subsystems (cells) of size related to r_{cut} and checking interaction only in close neighbourhood of the chosen cell, besides

(i) the information about the cell positions – (cell data structure)

we also need an information about

(ii) all particles contained in each cell (cell's particles data structure)

- partitioning of system into cells in 2D system**
(with minimal neighbourhood)

- minimal widths of cell's edges:

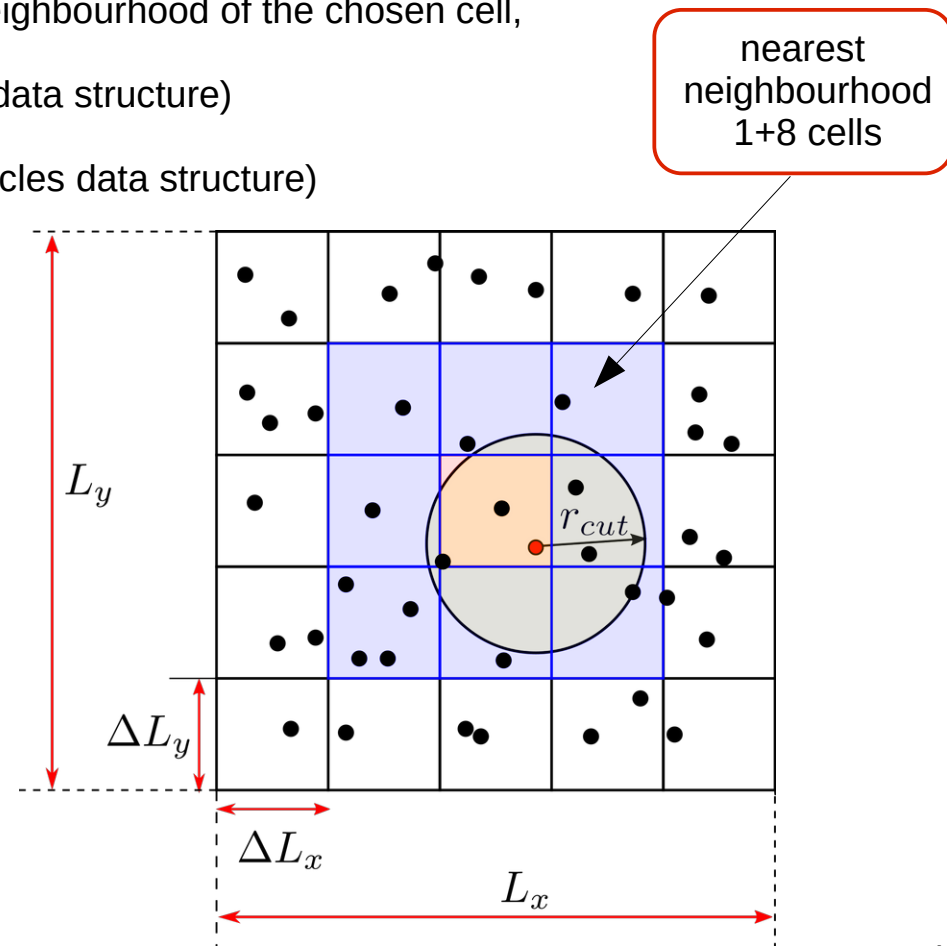
$$\Delta L_x, \Delta L_y \geq r_{cut}$$

- number of cells:

$$m_x = \text{floor} \left(\frac{L_x}{r_{cut}} \right), \quad m_y = \text{floor} \left(\frac{L_y}{r_{cut}} \right)$$

- target cell sizes

$$\Delta L_x = \frac{L_x}{m_x}, \quad \Delta L_y = \frac{L_y}{m_y}$$



Boundary Conditions (BC)

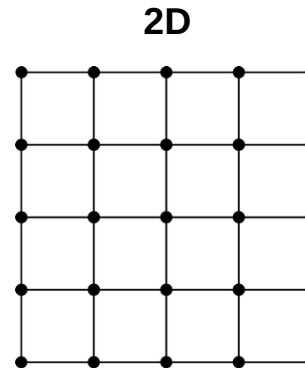
- the simplest MD simulations are conducted for atoms confined in square (2D)/cubic (3D) box
- for the atoms settled at the nodes of a square/cubic lattice we may calculate the number of atoms residing at the boundary edges/walls

What is the problem with boundary conditions?

Reflected BC – atoms reflect from boundaries and move toward the interior of system,

- the depleted particle density layers are formed near the boundary
- reflected particles do not randomize their velocities, so they introduce long-time correlations to the inner part of system
- artificial long-time correlations destroy anticipated correspondence between simulations and theoretical description derived for infinite systems

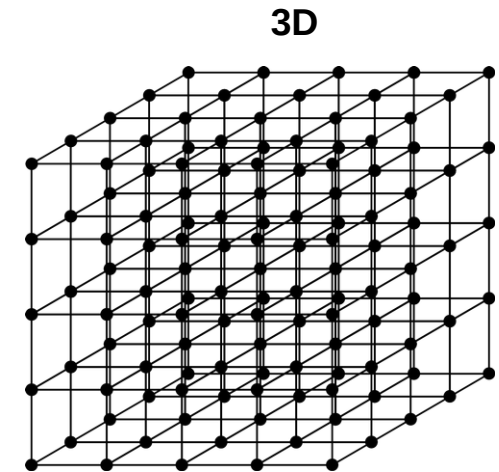
Periodic BC – are introduced to rid of the artificial reflections/fluctuations and hence to mimic the infinite systems



$$N = n^2$$

$$\frac{N_{\text{boundary}}}{N} = \frac{4}{n} - \frac{4}{n^2}$$

n	N	$\frac{N_{\text{boundary}}}{N}$
10	100	0.36
100	10^4	0.0396



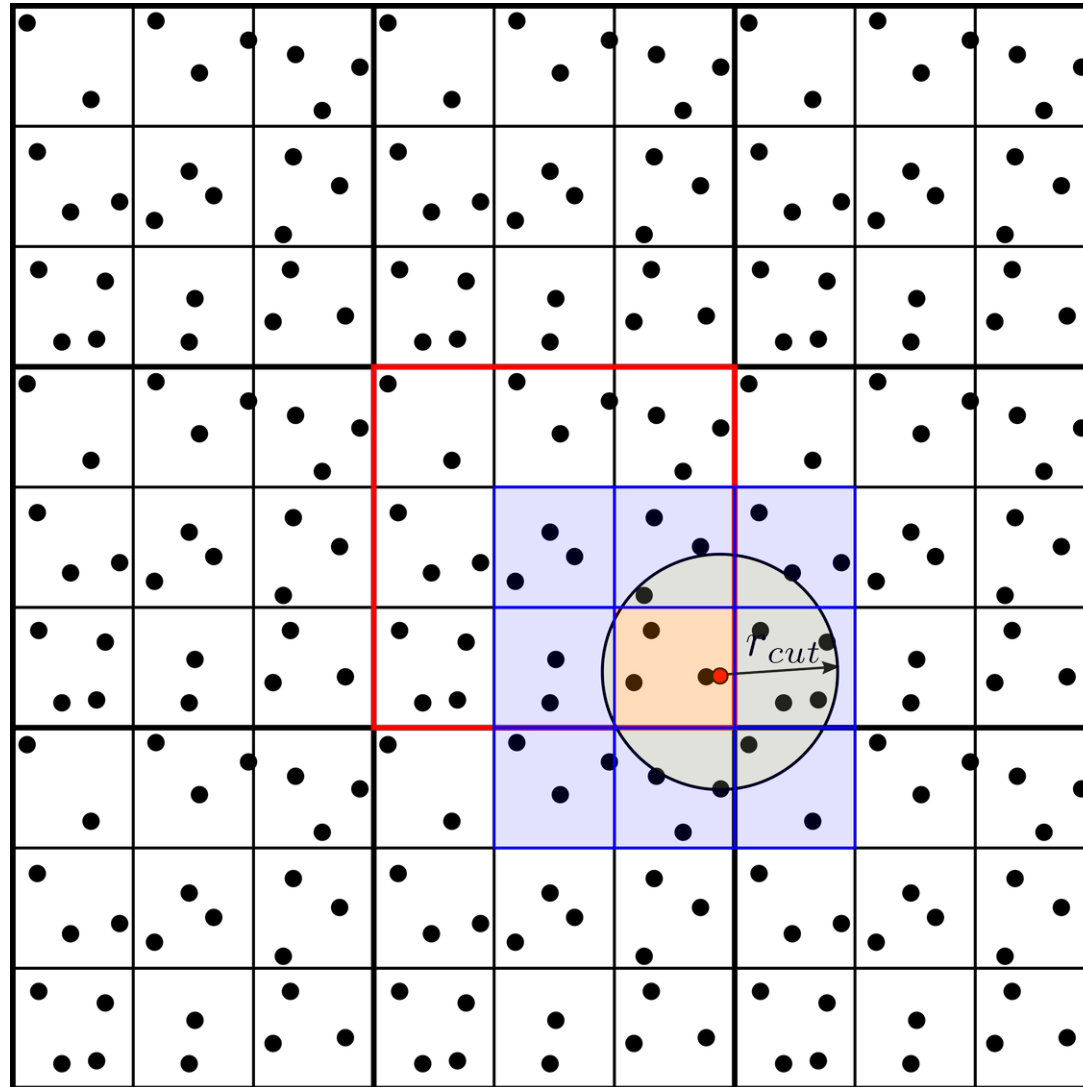
$$N = n^3$$

$$\frac{N_{\text{boundary}}}{N} = \frac{6}{n} - \frac{12}{n^2} + \frac{8}{n^3}$$

n	N	$\frac{N_{\text{boundary}}}{N}$
10	1000	0.488
100	10^6	0.0588

Periodic Boundary Conditions (PBC)

- these conditions mimics the infinite system by filling the space with copies of simulated system
- particles at positions close to any boundary now may interact with copies of particles from the opposite side of the system
- now each particle is surrounded with interacting neighbours, in more or less homogeneously manner (assuming it is a gas phase)



- algorithm for finding the nearest neighbour cells**

Let's consider 2D system with cells structure stored in 2D array (e.g. C/C++)

m_x, m_y – number of cells in x/y direction

```

set the indices of central cell:
for cell_x1=0 to  $m_x-1$  do
  for cell_y1=0 to  $m_y-1$  do

    determine the indices of the nearest cells:
    for i=-1 to 1 do
      for j=-1 to 1 do
        cell_x2= ( cell_x1+i+ $m_x$ ) mod  $m_x$ 
        cell_y2= ( cell_y1+j+ $m_y$ ) mod  $m_y$ 

      end do
    end do

  end do
end do

```

Optimization – Linked-Cell Method

- let's assume N particles are enumerated along with the indices in 1D array $i = 0, 1, 2, \dots, N - 1$
- two arrays are needed for storage of data (both initialized with some negative value e.g. -1):

$$header[0 : m_x - 1][0 : m_y - 1] = \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & -1 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

$$link[0 : N - 1] = \begin{array}{|c|c|c|c|c|c|} \hline -1 & -1 & -1 & \dots & -1 & -1 \\ \hline \end{array}$$

- now let's assume that in cell (k_x, k_y) there are 3 particles with following indices: 3, 27, 43

Iterating over all particles indices the following actions are taken

$$i = 3 : \quad link[3] = header[kx][ky] = -1 \quad - \text{there is no particles with lower indices in cell}$$

$$header[kx][ky] = 3$$

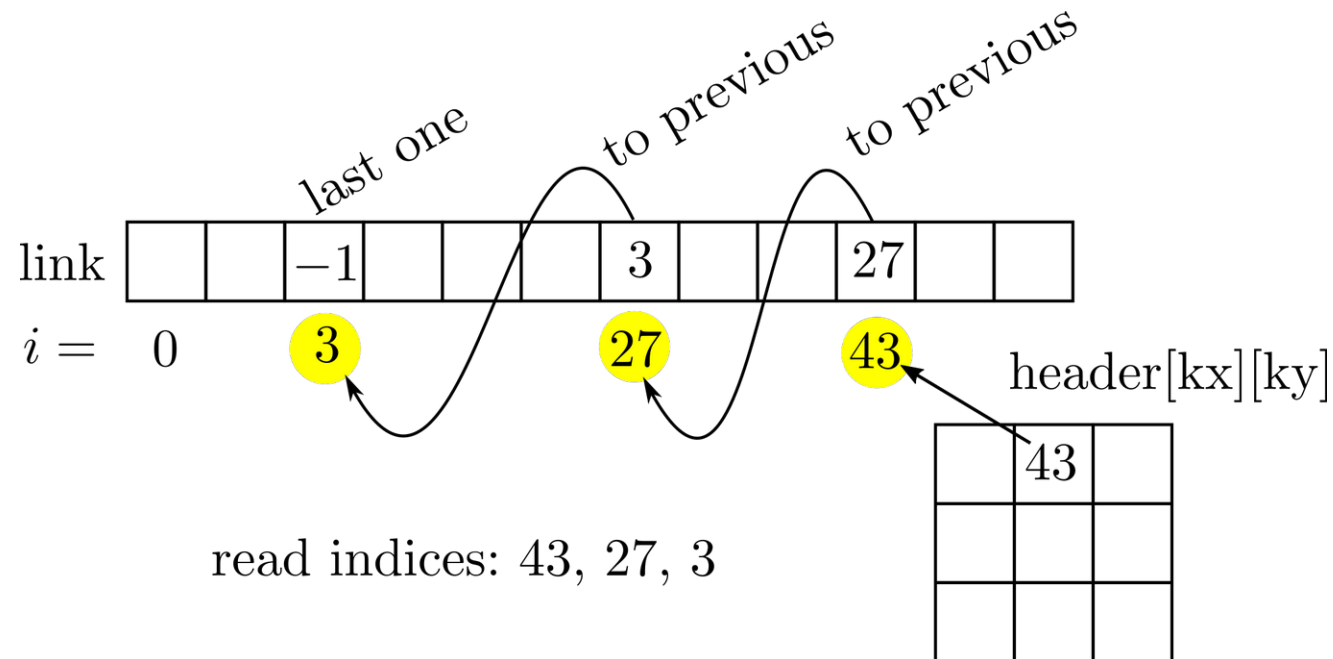
$$i = 27 : \quad link[27] = header[kx][ky] = 3 \quad - \text{link to previous particle}$$

$$header[kx][ky] = 27$$

$$i = 43 : \quad link[43] = header[kx][ky] = 27 \quad - \text{link to previous particle}$$

$$header[kx][ky] = 43 \quad - \text{highest particle's index in cell}$$

- scheme of example linked cell method



Remarks:

- method is simple and fast $\sim O(N)$, only one iteration over particles is needed to ascribe them to proper cells
- method uses minimal data structure $N + m_x \cdot m_y$ integers for:
link array and 2D **header** array
- extension to 3D is straightforward (just add the third index to header)

- linked-cell algorithm for ascribing particles to cells in 2D

```
for i=0 to N-1 do
```

$$k_x = \text{floor} \left(\frac{\text{particle}[i].x}{L_x} \right)$$

$$k_y = \text{floor} \left(\frac{\text{particle}[i].y}{L_y} \right)$$

```
    particles on the right/top boundary would be orphaned
```

```
    if (k_x == m_x) k_x = m_x - 1;
```

```
    if (k_y == m_y) k_y = m_y - 1;
```

```
    link[i] = header[k_x][k_y]
```

```
    header[k_x][k_y] = i
```

```
end do
```

- linked cell algorithm for calculations of two-particle interactions (double counting eliminated)

```

for i=0 to N-1 do
    particle[i].fx = particle[i].fy = particle[i].u = 0
end do

```

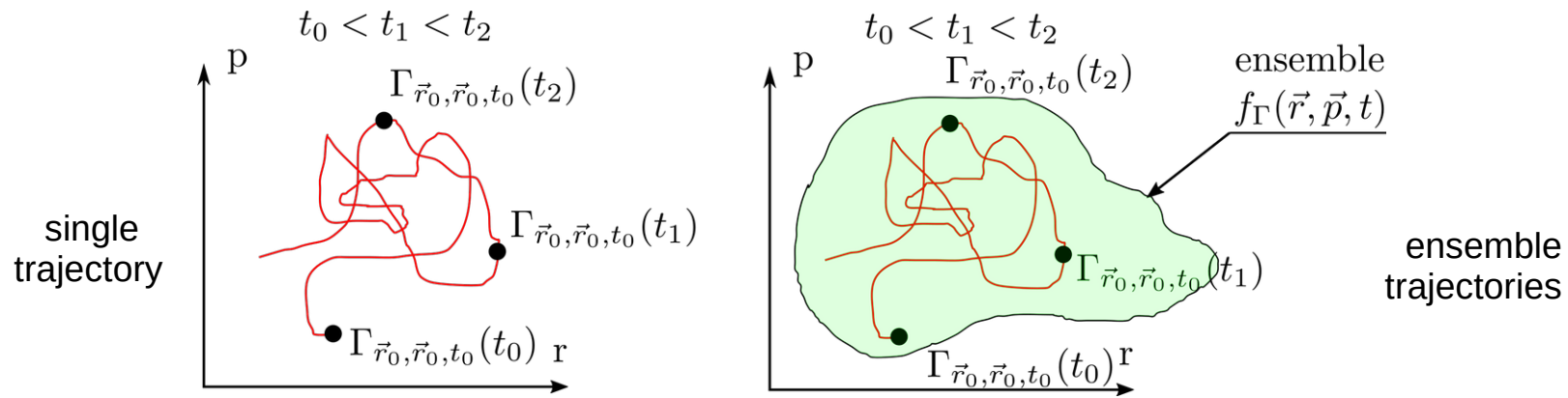
```

for i=0 to N-1 do
    kx ← particle[i].kx; ky ← particle[i].ky; xi ← particle[i].x; yi ← particle[i].y
    for jx0=kx - 1 to kx + 1 do
        for jy0=ky - 1 to ky + 1 do
            jx = (jx0 + mx) mod mx //indices of cell containing j-th particle
            jy = (jy0 + my) mod my
            j = header[jx][jy]
            while j > -1 and j > i do //eliminate double calculations
                xj = particle[j].x; yj = particle[j].y;
                if(jx0 < 0) xj ← xj - Lx
                if(jx0 ≥ mx) xj ← xj + Lx
                if(jy0 < 0) yj ← yj - Ly
                if(jy0 ≥ my) yj ← xj + Ly
                compute_force_u(xi, yi, xj, yj, uij, fxij, fyij)
                particle[i].fx ← particle[i].fx + fxij; particle[j].fx ← particle[j].fx - fxij
                particle[i].fy ← particle[i].fy + fyij; particle[j].fy ← particle[j].fy - fyij
                particle[i].u ← particle[i].u + uij; particle[j].u ← particle[j].u + uij
                j=link[j] //read index of next particle
            end do
        end do
    end do
end do

```

Statistical ensembles and averages

- classical (phenomenological) thermodynamics uses finite macroscopic quantities such as: temperature, heat, pressure etc. to describe properties of a thermodynamical system
- because macroscopic system is built of atoms/particles to be held at equilibrium, one can imagine of many similar systems that have the same macroscopic properties (P,V,T,N) but differ in positions and velocities of its constituents – **the set of such systems of the same macroscopic properties are called ensembles**
- the atoms in a system continuously change their positions and velocities, in other words the system moves along some trajectory in phase-space



- the ensemble fills a fraction of phase-space but nonhomogenously, the time-dependent probability of occupation of given point in phase-space is described by **probability density function $f(\mathbf{q}, \mathbf{p}, t)$**
- $f(\mathbf{q}, \mathbf{p}, t)$ is normalized and the knowledge of this function allows calculating any dynamical physical quantity $A(\mathbf{r}, \mathbf{p})$

$$\int d^{3N}r d^{3N}p f_{ens}(\vec{r}, \vec{p}, t) = 1 \quad (\vec{r}, \vec{p}) = (\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N, \vec{p}_1, \vec{p}_2, \dots, \vec{p}_N)$$

$$\langle A \rangle_{ens} \equiv \frac{\int d^{3N}r d^{3N}p A(\vec{r}, \vec{p}) f_{ens}(\vec{r}, \vec{p}, t)}{\int d^{3N}r d^{3N}p f_{ens}(\vec{r}, \vec{p}, t)}$$

the issue is: (i) f_{ens} is not known in general, and, (ii) it depends on so many variable that can not be used in practise

- **ergodic theorem** states that the averages calculated over all ensemble states are equivalent for the time averages for only one state

$$\langle A \rangle_{ens} = \lim_{t \rightarrow \infty} \langle A \rangle_t = \frac{1}{t} \int_{t_0}^{t_0+t} A(\Gamma_{\vec{r}_0, \vec{p}_0, t_0}(t)) dt' \quad \text{- integration along the trajectory}$$

$\Gamma_{\vec{r}_0, \vec{p}_0, t_0}(t)$ - trajectory in phase-space

and that changes things as now one is able to determine microscopic quantities (P,T,E) just by calculating the mean over the trajectory followed by the system propagated in MD

$$\Delta t = \frac{t_{max}}{n} \quad t_i = t_0 + i \cdot \Delta t, \quad i = 0, 1, 2, \dots, n$$

the superscript m denotes the moment of distribution:

$$\langle A^m \rangle_t \approx \overline{A^m} = \frac{1}{n} \sum_{i=1}^n A^m(\Gamma_{\vec{r}_0, \vec{p}_0, t_0}(t_i)), \quad m = 1, 2$$

$$var(A) = \langle A^2 \rangle_t - \langle A \rangle_t^2 \quad \text{- variance estimates the fluctuations}$$

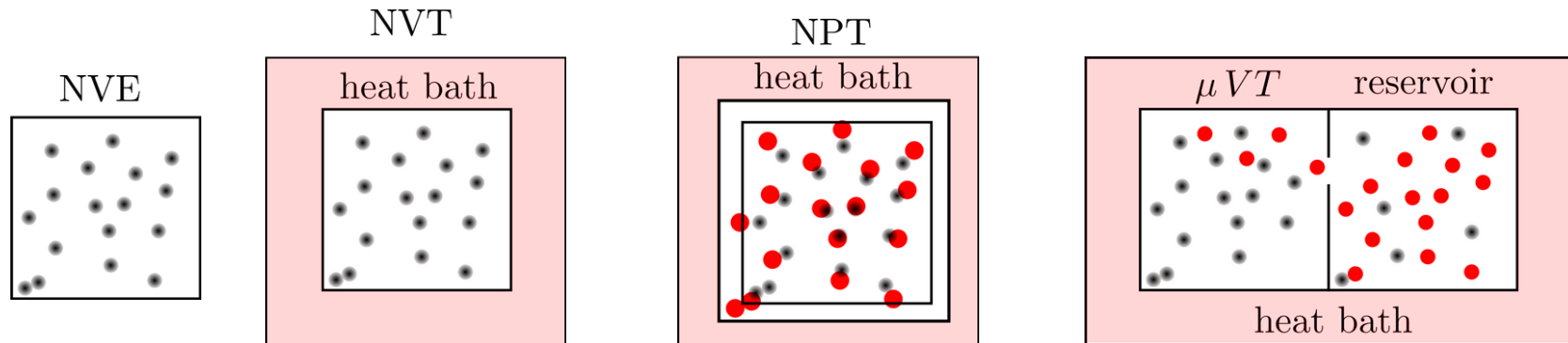
Reamarks:

- assuming the system is in equilibrium, the time-averaged value wouldn't depend on chosen time t_0
- obviously the time of gathering the essential data is limited, but it would depends on amplitude of fluctuations measured by variance

- types of ensembles

- microcanonical: $N, V, E = \text{const}$
- canonical: $N, V, T = \text{const}$
- isobaric-isothermal: $N, P, T = \text{const}$
- grand canonical: $\mu, V, T = \text{const}$

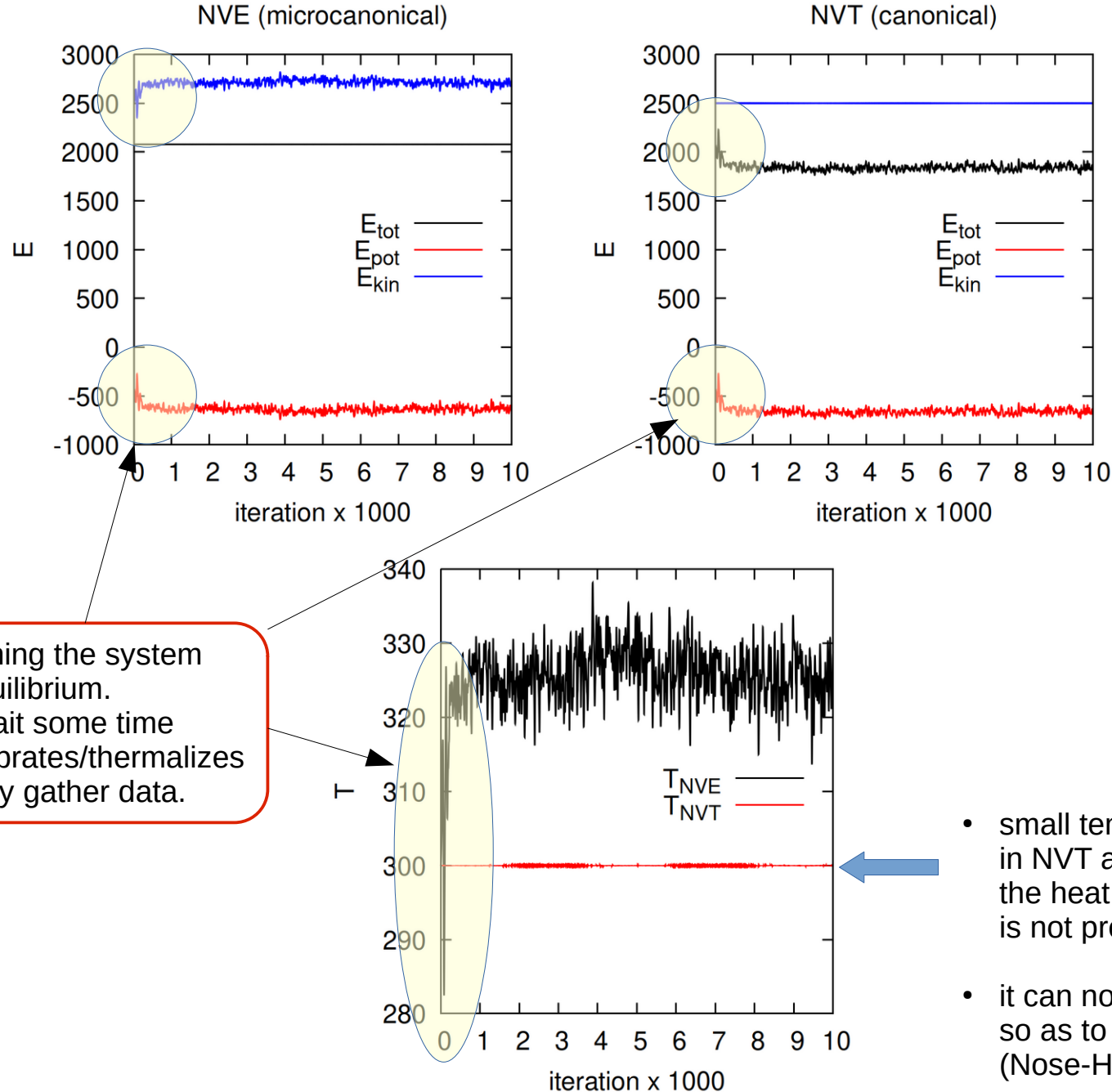
N – number of particles in system
 V – volume of system
 E – total energy of system
 P – pressure
 T – temperature
 μ – chemical potential



- NVE ensemble has only theoretical significance because the fully isolated system can not be realized experimentally
- NVT ensemble is kept at constant temperature so it is in thermal equilibrium with thermostat (heat-bath), notice that the total energy is changing in time
- NPT ensemble is in equilibrium with thermostat and additionally the volume the system may change so as to keep the pressure constant as well
- μVT ensemble keeps constant temperature but the pressure, energy and the number of particles in the system may change

In laboratory project we will focus on NVE and NVT systems as these two are the easiest to simulate within MD.

- example MD simulations data for NVE & NVT systems (1000 argon atoms in 2D)



- small temperature fluctuations in NVT are allowed because the heat bath-system coupling is not perfect
- it can not be extremely strong so as to avoid numerical instability (Nose-Hoover thermostat)

Thermalization

- the basic MD algorithm simulates NVE system, i.e. energy is preserved
- for NVT it must be modified in such a way to keep temperature constant i.e. to keep the system in thermodynamic equilibrium with thermostat (heat bath)
- constant temperature can be obtained with the following methods:
(i) scaling velocity, (ii) Berendsen thermostat, (iii) Nose-Hoover thermostat
- (i) and (ii) are not time-reversed while but are simple

Equipartition theorem states that the instantaneous temperature linearly depends on kinetic energy

$$\mathcal{T} = \frac{2}{Dk_bN} \sum_{i=1}^N \frac{m_i(v_i(t))^2}{2} \quad D=2,3 - \text{number of dimensions}$$

therefore one may relate current temperature with the desired one T_D

$$T_D = \alpha^2 \mathcal{T}$$

$$\alpha = \sqrt{\frac{T_D}{\mathcal{T}}} = \begin{cases} > 1 & \Longleftrightarrow T_D > \mathcal{T} \\ < 1 & \Longleftrightarrow T_D < \mathcal{T} \end{cases}$$

factor α is used in scaling the particles' velocities for the next time step

$$\vec{v}_i \leftarrow \alpha \vec{v}_i, \quad i = 1, 2, \dots, N$$

The issue is that such rescaling produces velocity distribution which differs from expected Maxwell-Boltzmann.

Another way of velocity rescaling is **Berendsen thermostat**

$$\alpha = \sqrt{1 + \frac{\Delta t}{\tau_{lag}} \left(\frac{T_D}{\mathcal{T}} - 1 \right)}$$

- the factor τ_{lag} makes the velocity changes are time-lagged
- its large value means the system approaches the desired temperature after many time steps
- for $\tau_{lag} = \Delta t$ the simple velocity rescaling in single time step is restored

Nose-Hoover thermostat

- in this method the fictitious friction variable is introduced, it's EOM is add to particles' EOM and all EOM must be solved at the same time
- it takes or gives the energy to the system, depending on instantaneous temperature

$$\frac{d^2 \vec{r}_i}{dt^2} = \frac{\vec{F}_i}{m_i} - \xi \vec{v}_i, \quad i = 1, 2, \dots, N$$

$$\frac{1}{2} Q \frac{d\xi}{dt} = \frac{1}{2} \sum_{i=1}^N \frac{m_i \vec{v}_i^2}{2} - \frac{1}{2} (3N + 1) k_b T_D$$

$$Q = \underbrace{10^{-2}, 10^{-1}}_{\text{strong coupling}}, 1, \dots, \underbrace{10^3, \dots}_{\text{weak coupling}}$$

$Q=10^{20}$ - turns coupling off completely
(we can simulate NVE and NVT with the same code)

Nose-Hoover thermostat + velocity-Verlet method

single time step

initialization: $\xi^{n=0} = 0$

for $i=0$ to $N-1$ do

$$\vec{v}_i^{n+\frac{1}{2}} \leftarrow \vec{v}_i^n + \frac{\Delta t}{2} \left[\frac{\vec{F}_i^n}{m_i} - \xi^n \vec{v}_i^n \right]$$

$$\vec{r}_i^{n+1} \leftarrow \vec{r}_i^n + \vec{v}_i^{n+\frac{1}{2}}$$

end do

few Picard iterations for ξ :

$$\text{init: } \vec{v}_i^{n+1} \leftarrow \vec{v}_i^{n+\frac{1}{2}}, \quad i = 0, 1, \dots, N-1$$

for $k=1$ to 5 do

predictor corrector for $\xi(D=2, 3)$:

$$\xi^{n+1} \leftarrow \xi^n + \frac{\Delta t}{Q} \left(\sum_{i=0}^{N-1} \frac{m_i (\vec{v}_i^{n+1})^2}{2} - \frac{D}{2} k_b T_D (N+1) \right)$$

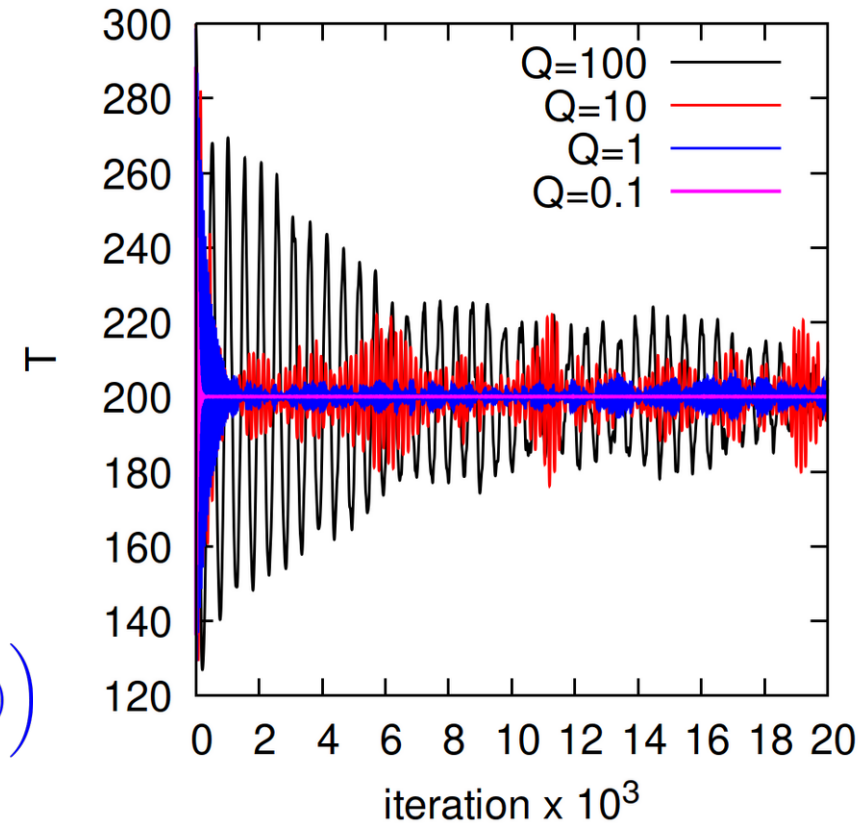
for $i=0$ to $N-1$ do

$$\vec{v}_i^{n+1} \leftarrow \frac{1}{1 + \frac{\Delta t}{2} \xi^{n+1}} \left(\vec{v}_i^{n+\frac{1}{2}} + \frac{\Delta t}{2} \vec{a}_i^{n+\frac{1}{2}} \right)$$

end do

end do

- 1000 argon particles
- $T_{\text{init}}=300$ K
- $T_D=200$ K



Averages of physical quantities

N- number of particles $\Delta t = \frac{t_{max}}{n}, \quad t_j = \Delta t \cdot j, \quad j = 1, 2, 3, \dots, n$

- kinetic energy

$$\mathcal{E}_{kin}(t) = \sum_{i=1}^N \frac{m_i (\vec{v}(t))^2}{2} \quad \Longrightarrow \quad E_{kin} = \frac{1}{n} \sum_{j=1}^n \mathcal{E}_{kin}(t_j)$$

- potential energy

$$\mathcal{E}_{pot}(t) = \sum_{i=1}^N U^{(1)}(\vec{r}_i(t)) + \sum_{i=1}^N \sum_{j>i}^N U^{(2)}(\vec{r}_i(t), \vec{r}_j(t)) \quad \Longrightarrow \quad E_{pot} = \frac{1}{n} \sum_{j=1}^n \mathcal{E}_{pot}(t_j)$$

- total energy

$$\mathcal{E}_{tot}(t) = \mathcal{E}_{kin}(t) + \mathcal{E}_{pot}(t) \quad \Longrightarrow \quad E_{tot} = E_{kin} + E_{pot}$$

- temperature

NVT (canonical): $D=2,3$ – number of dimensions

$$\mathcal{T}(t) = \frac{2}{D N k_b} \sum_{i=1}^N \frac{m_i (\vec{v}_i(t))^2}{2} = \frac{2}{D N k_b} \mathcal{E}_{kin}(t)$$

$$T = \frac{2}{D N k_b} \frac{1}{n} \sum_{j=1}^n \mathcal{E}_{kin}(t_j) = \frac{2}{D N k_b} E_{kin}$$

NVE (microcanonical):

$$T = \frac{2}{D N k_b} \left[\frac{1}{n} \sum_{j=1}^n \frac{1}{\mathcal{E}(t_j)} \right]^{-1}$$

The difference between temperatures calculated for both ensembles decreases as $O(1/N)$.

- pressure (3D) according to comon convention, here: V – volume(3d) / area(2d)

instantnaeous:
$$\mathcal{P}(t) = \frac{2}{DV} \sum_{i=1}^N \left(\underbrace{\frac{m_i (\vec{v}_i(t))^2}{2}}_{kinetic} + \underbrace{\frac{1}{2} \vec{r}_i(t) \cdot \vec{F}_i(t)}_{interactions} \right), \quad \begin{array}{l} V - \text{area/volume} \\ D = 2/3- \text{ dimensions} \end{array}$$

averged:

$$PV = Nk_bT + W, \quad (V, T, W - \text{averages})$$

W is the virial function related to interactions in system

$$W = \frac{1}{n} \sum_{j=1}^n \mathcal{W}(t_j)$$

$$\mathcal{W}_j(t) = \frac{1}{D} \sum_{i=1}^N \sum_{j>i} \vec{r}_{ij}(t) \cdot \vec{F}_{ij}(t)$$

- heat capacity (at constant volume)

macroscopic definition:

$$C_v = \left(\frac{\partial E}{\partial T} \right)_{V=const}$$

NVT (canonical):

$$C_v = \frac{1}{k_b T^2} \text{var}(E_{tot})$$

variance of energy:

$$\text{var}(E) = \langle E^2 \rangle_t - \langle E \rangle_t^2$$

NVE (microcanonical): total energy value is preserved, but the kinetic and potential energies fluctuate identically

$$C_v = \frac{DNk_b}{2} \left[1 - \frac{2}{DNk_b^2 T^2} \text{var}(E_\alpha) \right]^{-1}, \quad \alpha = E_{kin}, E_{pot}$$

D – number of dimensions

- pair distribution function → radial distribution function (**RDF**)
- conducting MD simulations we are often interested in static and dynamic properties of system in question
- an important static property is the spatial structure of atomic configuration, and because it directly comes from two-particle interactions one defines the **pair distribution function**

$$g(\vec{r}', \vec{r}'') = \frac{\Omega^2}{N(N-1)} \left\langle \sum_{i=1}^N \sum_{j \neq i}^N \delta(\vec{r}' - \vec{r}_i(t)) \delta(\vec{r}'' - \vec{r}_j(t)) \right\rangle_t$$

Because its value depends on positions of two particles, we may expect it would take large values at short inter-particle distances, let's say on the length of a few σ .

On the other hand, at larger distances correlations must vanish and then both sums are uncorrelated and can be calculated separately

$$\lim_{||\vec{r}' - \vec{r}''|| \rightarrow \infty} = \frac{\Omega^2}{N(N-1)} \left\langle \sum_{i=1}^N \delta(\vec{r}' - \vec{r}_i(t)) \right\rangle_t \left\langle \sum_{j \neq i}^N \delta(\vec{r}'' - \vec{r}_j(t)) \right\rangle_t = \frac{\Omega^2}{N(N-1)} \frac{N}{\Omega} \frac{N-1}{\Omega} = 1$$

This is an asymptotic behaviour of pair distribution function, this flat plateau means that the motions of two remote particles are uncorrelated.

In MD simulations we more interested what is going on at smaller distances.

- when the particle system is **spatially homogeneous** this function can be written in more compact form

first, let's introduce the difference between $\vec{r}'$ and $\vec{r}''$

$$\vec{r}' = \vec{r}'' + \vec{r} \quad \Rightarrow \quad g(\vec{r}'' + \vec{r}, \vec{r}'') = \frac{\Omega^2}{N(N-1)} \left\langle \sum_{i=1}^N \sum_{j \neq i}^N \delta(\vec{r}'' + \vec{r} - \vec{r}_i(t)) \delta(\vec{r}'' - \vec{r}_j(t)) \right\rangle_t$$

next integrate out $\vec{r}''$ then we get

$$\int_{\Omega} g(\vec{r}'' + \vec{r}, \vec{r}'') d\Omega'' = g_2(\vec{r})$$

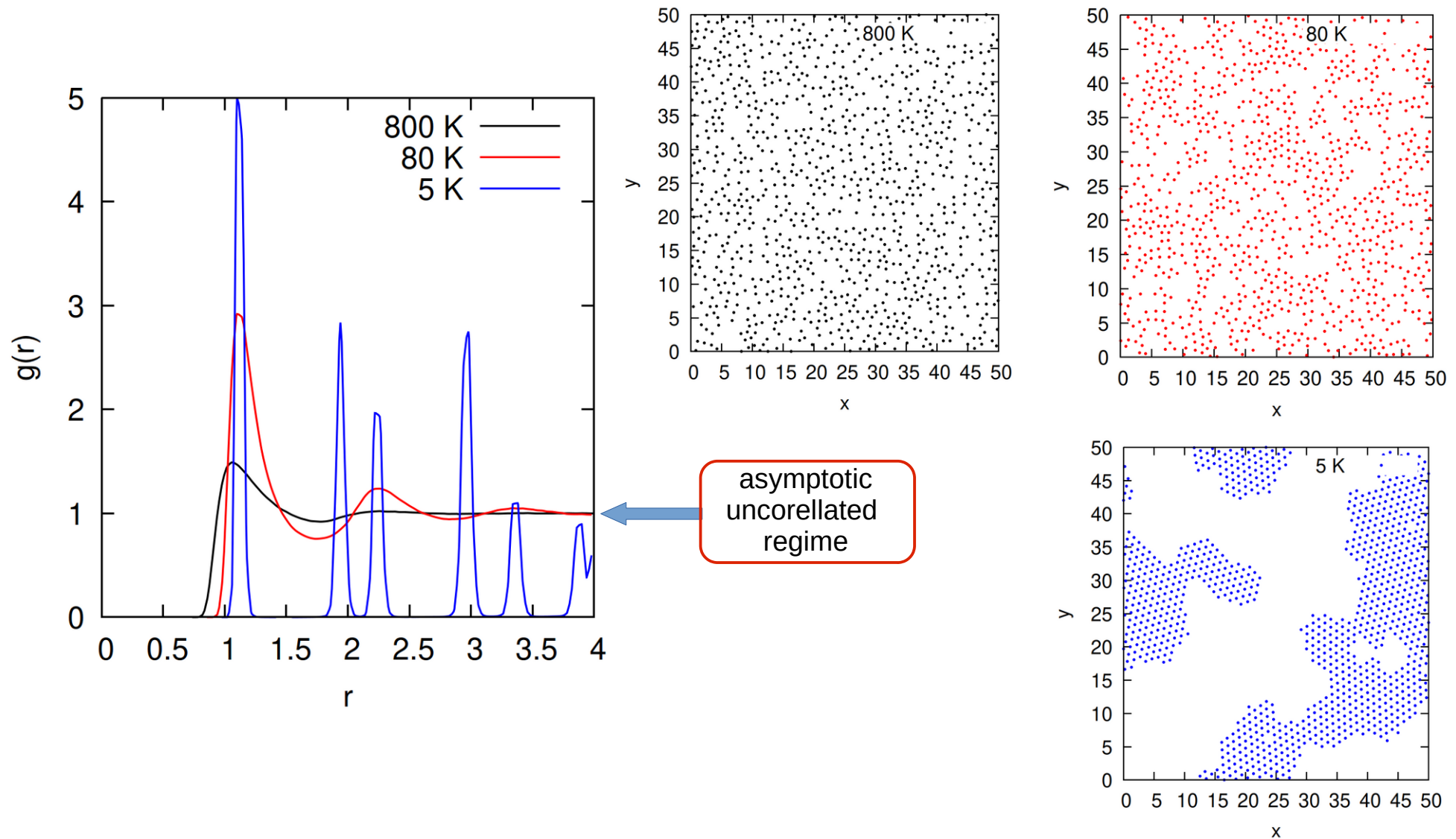
$$g_2(\vec{r}) = \frac{2\Omega}{N(N-1)} \left\langle \sum_{i=1}^N \sum_{j > i}^N \delta(\vec{r} - \vec{r}_{ij}(t)) \right\rangle_t$$

- the factor 2 occurs here due to skipping double summation

Interpretation: $g_2(r)$ shows the probability of finding any particle at distance r from the particle settled at the origin of local coordinate system

- example of RDF for 2D system

- 1000 argon atoms were cooled down from 800K to 5K
- system undergoes continuously (2D) following transformations: **gas** $\rightarrow$ **liquid** $\rightarrow$ **solid**



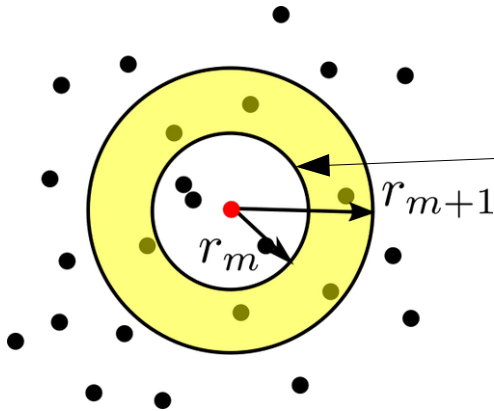
- calculations of RDF histogram**

- in MD simulation we can not work with definition of RDF because it contains Dirac delta function
- instead the essential data are gathered in the course of simulation and the histogram for RDF is built

Ω – the volume of system

$$2D : \quad \Omega = L_x L_y$$

$$\Delta\Omega_m = 4\pi(r_{m+1}^2 - r_m^2)$$



$$3D : \quad \Omega = L_x L_y L_z$$

$$\Delta\Omega_m = \frac{4}{3}\pi(r_{m+1}^3 - r_m^3)$$

initialization :

M_{bin} – number of bins

$r_{max} = 2 \div 4\sigma$ – range of histogram

$\Delta r = \frac{r_{max}}{M_{bin}}$ – width of bins

$RDF[0:M_{bin}-1]=0$ – array for RDF

time iteration loop:

for $t=0$ to t_{max} by Δt do

other calculations: forces , Verlet etc.

gather contributions for RDF:

for $i=0$ to $N-1$ do

for $j=i+1$ to $N-1$ do

$$\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$$

$$m = \text{floor}\left(\frac{r_{ij}}{\Delta r}\right)$$

if ($m < M$)

$$RDF[m] += \frac{2\Omega}{N(N-1)} \frac{1}{\Delta\Omega_m}$$

end do

end do

end do

normalize RDF data for time averaging

for $m=0$ to $M_{bin}-1$ do

$RDF[m] /= n$

end do

Dynamic properties – time-correlation function

- the time correlation between two quantities A and B means that for measured value of A(t') we observed dependence of value at later times B(t'') ($t'' > t'$)
- in other words, non-vanishing time correlation means that system has something like „memory”
- analysis of these types of correlations, other are the static correlations, gives us additional informations about the dynamical properties of a system
- in equilibrium the time-correlation always depends on the difference

$$t = t'' - t'$$

only, but not on their individual values, we can choose arbitrarily the **time interval T** for calculation of correlations

General form of **time-correlation function** is following

$$C_{AB}(t) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{t_0}^{t_0+T} A(t') B(t+t') dt' = \langle A(t') B(t+t') \rangle_{time}$$

we can measure correlation between any two quantities, but if we set A=B then we get an **autocorrelation function**

$$C(t) = C_{AA}(t) = \langle A(t') A(t+t') \rangle_{time}$$

- in practice slightly modified expression is used

$$\delta A(t) = A(t) - \langle A \rangle_{time}$$

$$C(t) = \langle \Delta A(t + t') \Delta A(t') \rangle_{time}$$

then we get for $t=0$

$$C(0) = \langle \delta A \rangle_{time} = var(A)$$

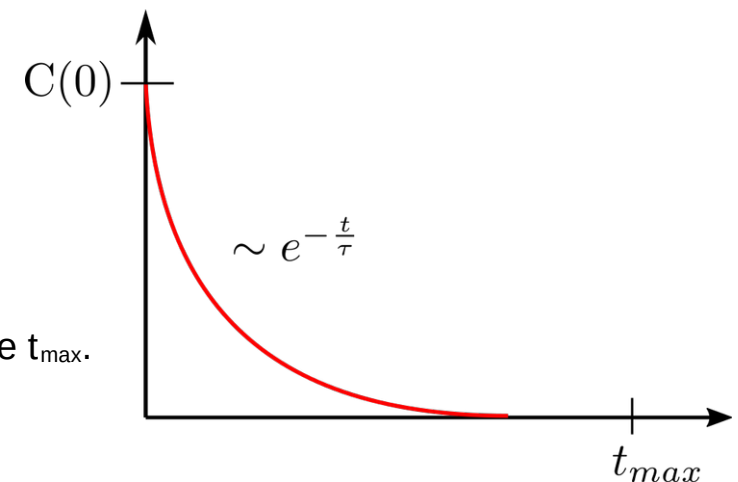
while for $t \rightarrow \infty$ we don't expect correlations

$$\lim_{t \rightarrow \infty} C(t) = 0$$

Thus we can calculate the characteristic time after which system forgets about any perturbation that came into existence and returns to equilibrium - it is a **relaxation time**

$$\tau_{rel} = \frac{1}{C(0)} \int_0^{\infty} C(t) dt$$

in practice the upper limit in integral is substituted by some finite long time t_{max} .



- calculation of time-correlation function

The best way is to make MD simulation and gather needed data in file, then we have all required data to be used in calculation of time-correlation.

mean value: $\langle A \rangle \approx \frac{1}{n} \sum_{i=1}^n A_i$

$$t = \Delta t \cdot k, \quad k = 0, 1, 2, \dots$$

$$C(t) = C(\Delta t k) = C_k = \frac{1}{n-k} \sum_{i=0}^{n-k-1} (A_{i+k} - \langle A \rangle) (A_i - \langle A \rangle)$$

- example: **diffusion coefficient** from velocity autocorrelation

$$C_v(t) = \sum_{i=1}^N \langle \vec{v}_i(t+t') \vec{v}_i(t') \rangle$$

$$D = \frac{1}{3} \int_0^{\infty} dt C_v(t)$$

integration by parts leads to **Einstein relation**

$$D = \frac{1}{6t} \langle (\vec{r}(t+t') - \vec{r}(t'))^2 \rangle$$

Phase transition in MD

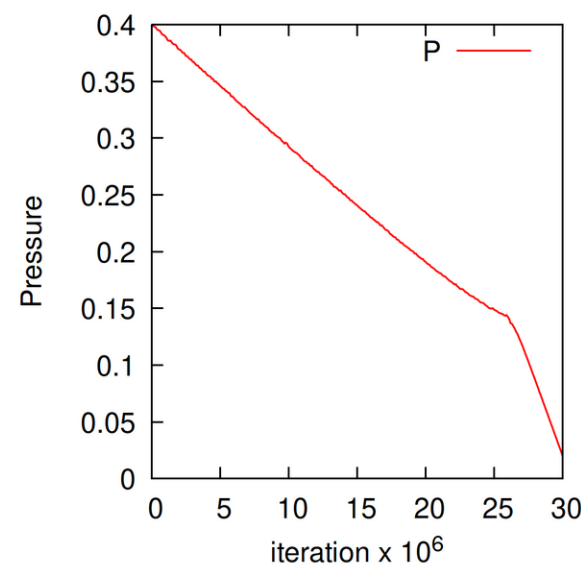
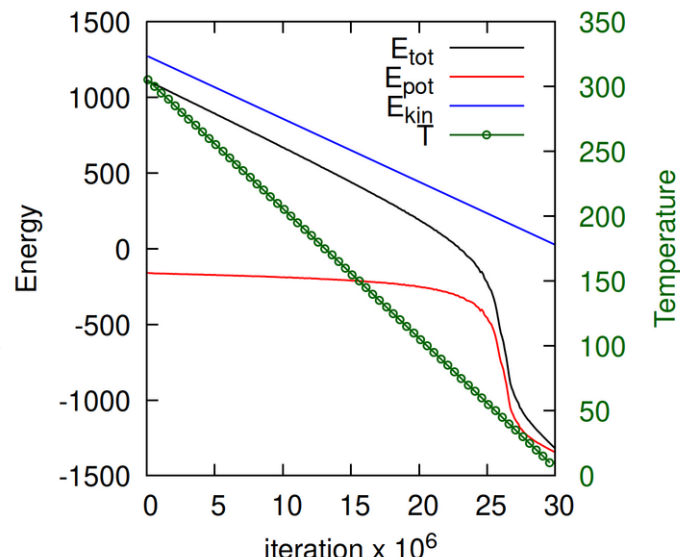
- phase transition is the transformation between two states of some medium that differ much in their properties

gas $\rightarrow$ liquid or liquid $\rightarrow$ solid
- gas-liquid transition is observed for boiling/condensation processes so its occurrence relates with abrupt change in density of water
- for liquid-solid transition there are no changes in density (besides the water), so other parameters must change which relate to internal structure of medium, this for example is the specific heat
- in thermodynamics we distinct **1-st** and **2-nd** phase transitions:
 - 1-st order** relates to discontinuous first-order derivative of free energy with respect to some thermodynamic variable (temperature, pressure etc),
 system absorbs or gives back the energy (latent heat) but doesn't change temperature,
 example: boiling/condensation, sublimation, crystalization/solidification of water
 - 2-nd order** relates to discontinues in second order derivative of free energy, the first order is continuous so it is also called **continuous phase transition** then susceptibility becomes divergent, correlation length increases to infinity,
 examples: ferromagnetic transition in spin chain, superconductivity
- the type of phase transition may depend on dimensionality of problem, for example:
 - liquid-solid in 3D system of argon atoms is 1-st order transition
 - in 2D it becomes continuous phase transition

Simulation of phase transition of argon atoms in 2D

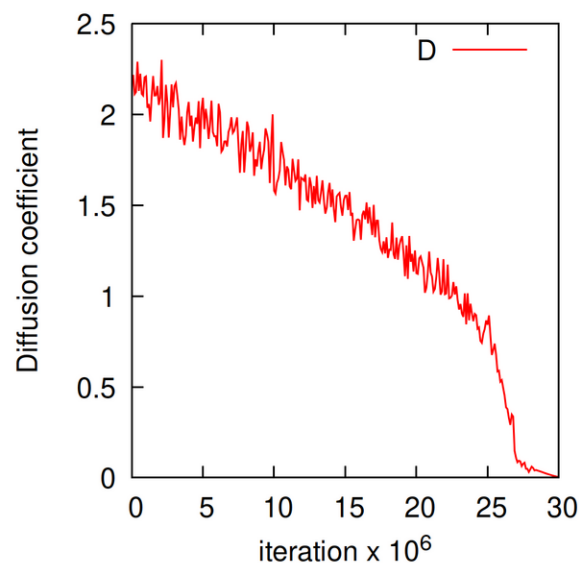
volume: $V=L^2$, $L=50$
 number of particles: $N=500$
 cooling down: $T_{\text{start}} = 305 \text{ K}$, $T_{\text{end}} = 5 \text{ K}$, $\Delta T = 1 \text{ K (step)}$
 iterations: total = 3×10^7 , at given temperature = 10^5

energy changes
continuously
(2-nd order)

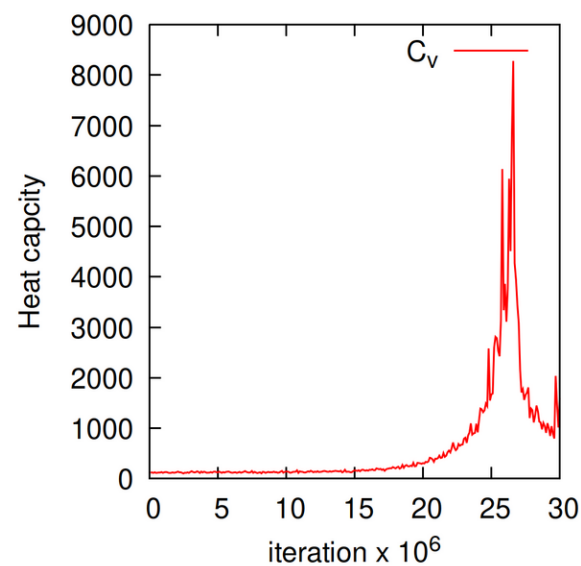


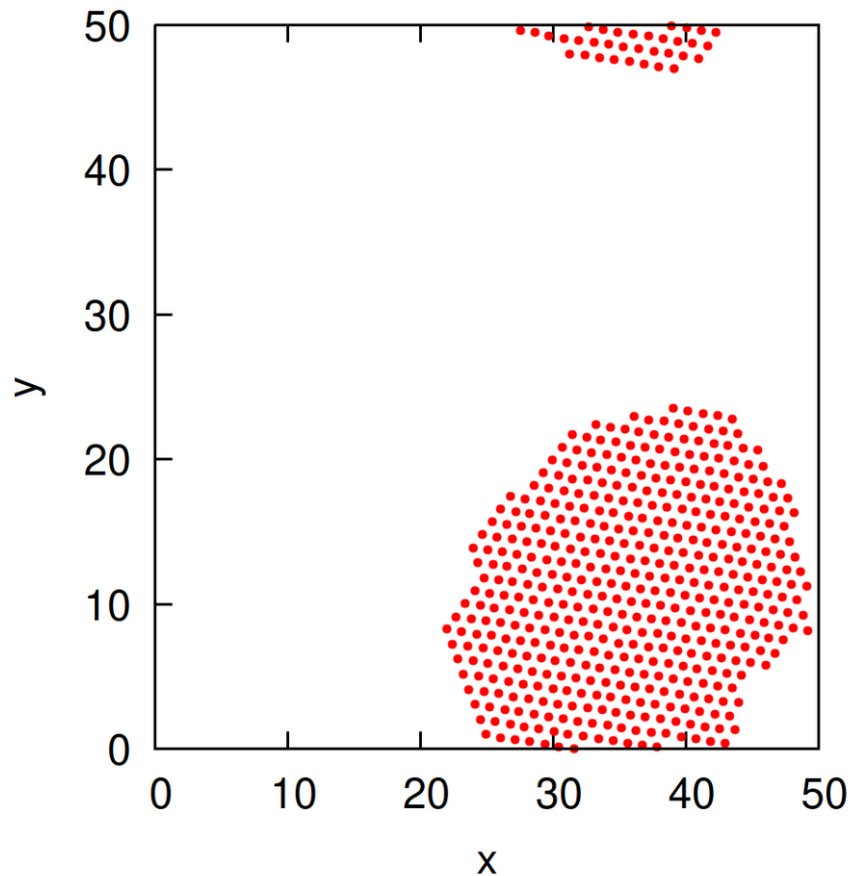
abrupt change
in pressure

large fluctuations
in gas phase,
diffusion in solid
vanishes

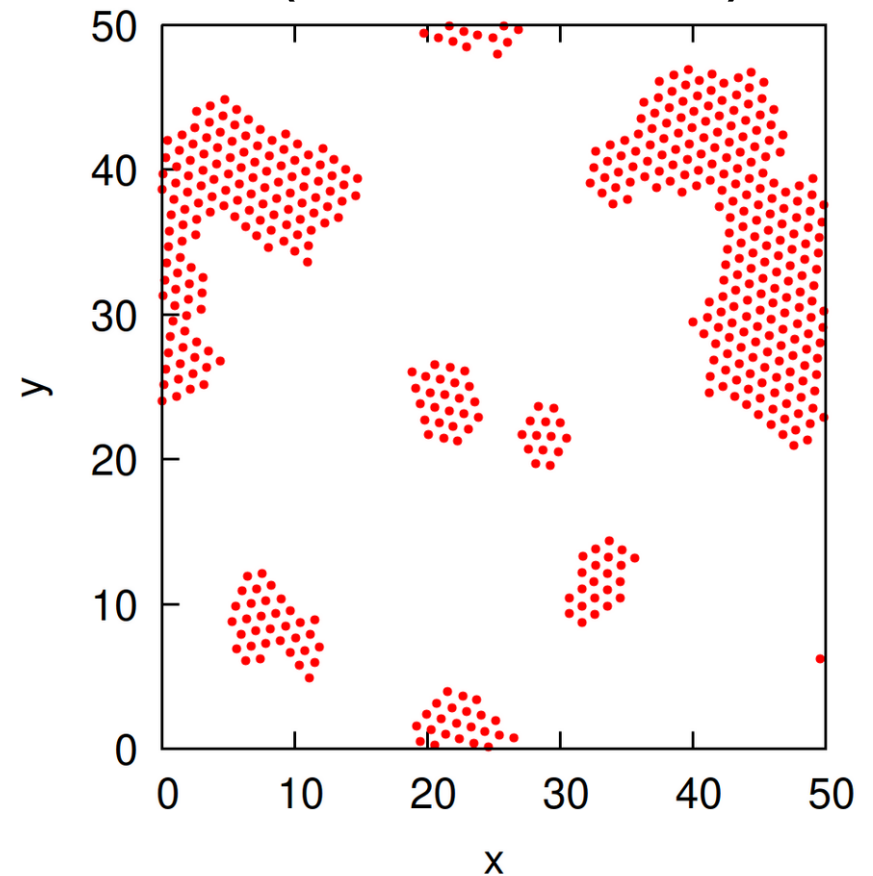


high peak
but of finite width
(not abrupt)



SLOW COOLING – 5 K

- system dissipates energy slowly and had enough time to attach atoms one-by-one to single solid

**FAST COOLING – 5 K
(100 times less iterations)**

- the energy dissipation rate is so fast, that distant bunches of atoms could only form small fragments of solid