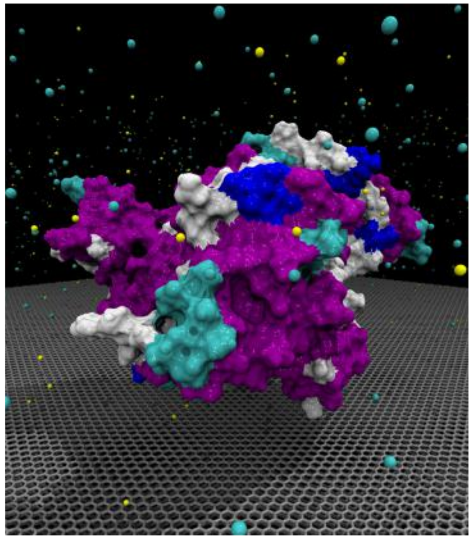
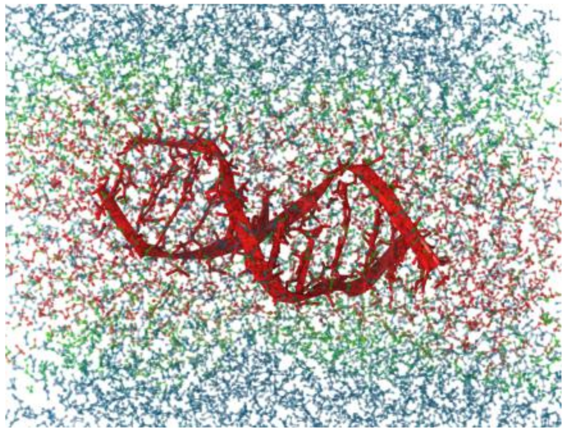
Molecular dynamics: Overview

H. M Urbassek

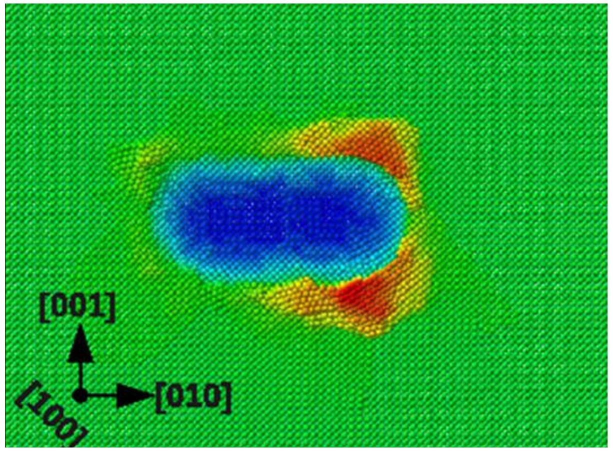
Physics Dept., University of Kaiserslautern, Germany



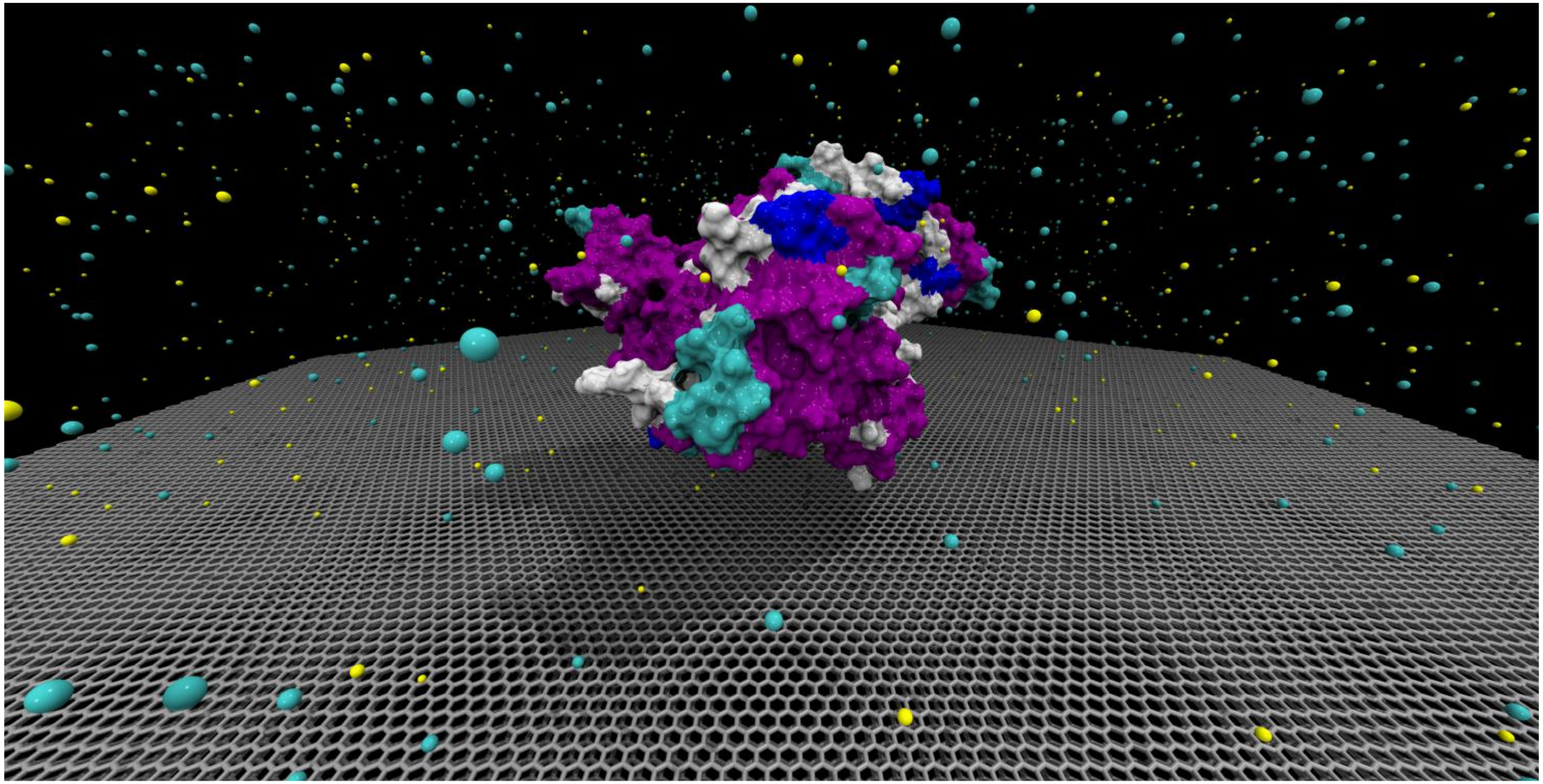
Protein adsorption



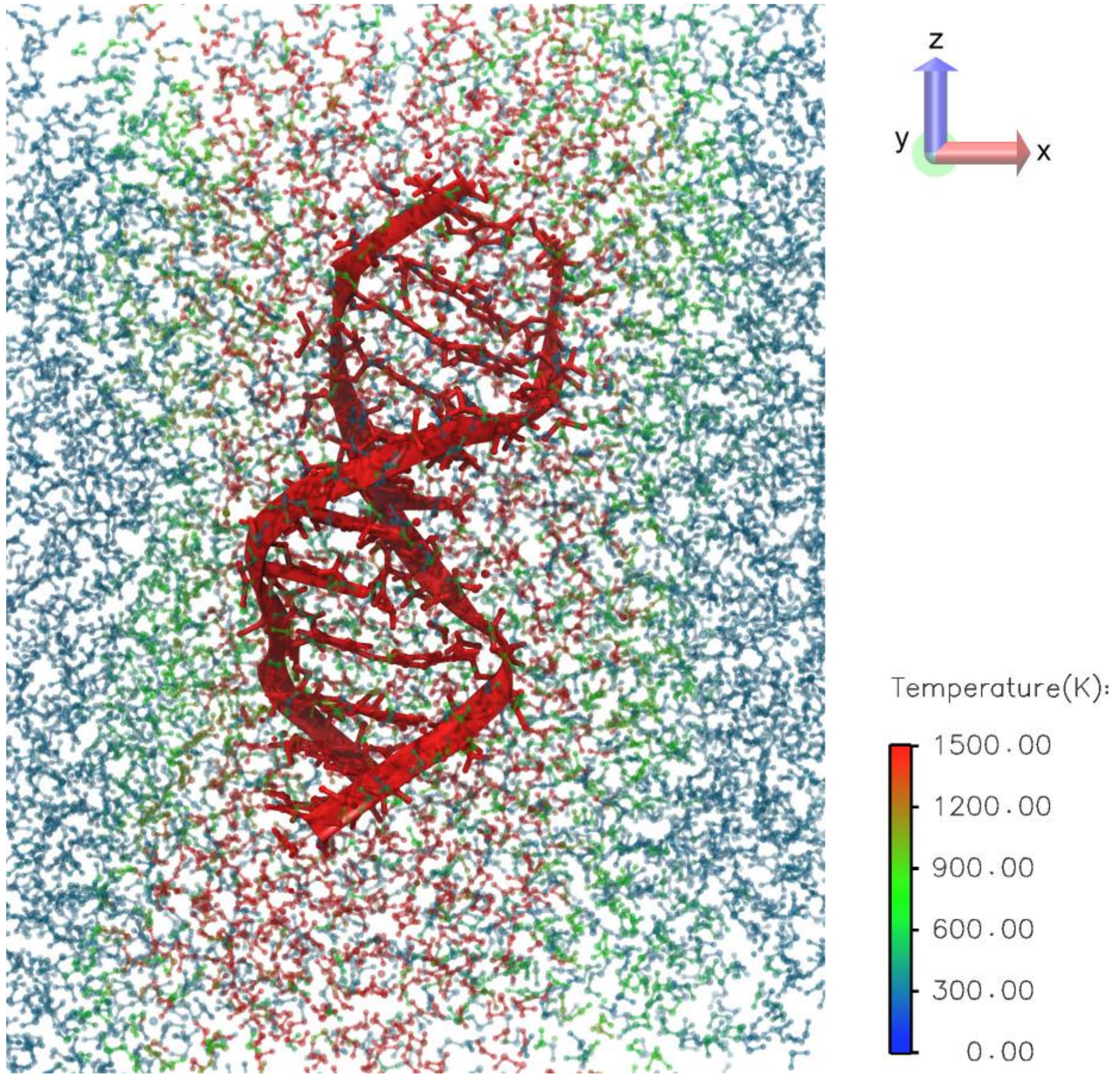
Hadron therapy



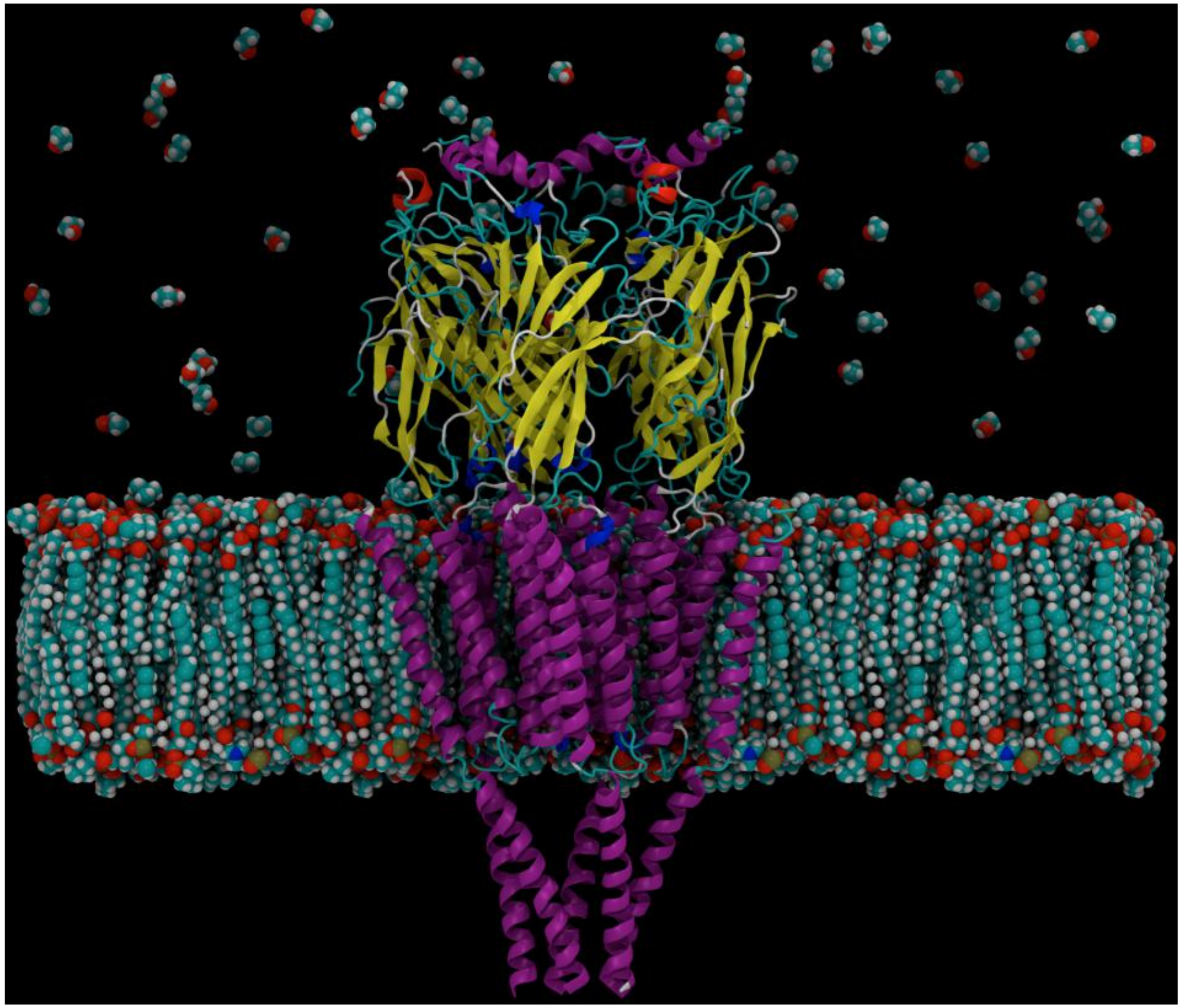
Nanoscratching



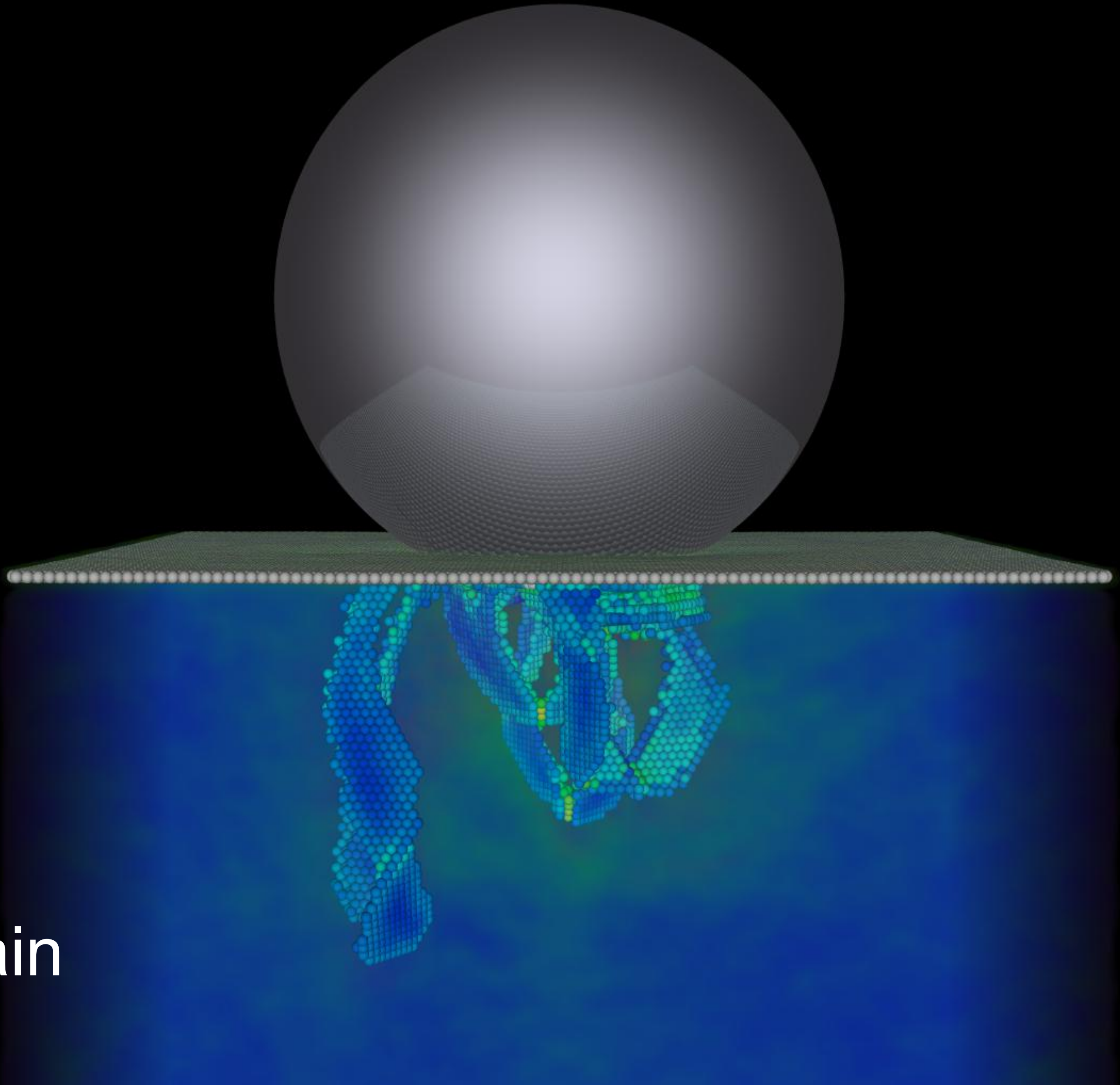
Bovine serum albumin (BSA)
in water (not shown) containing Na and Cl ions
before adsorption on a graphite surface



Bottländer, Mücksch, Urbassek 2015


passage of ethanol through acetylcholine receptor

Nanoscale Effects on Indentation



Gerolf Ziegenhain

Theoretical tools

- **Molecular dynamics**

Solve Newton‵s equations.

Advantages:

- as realistic as possible in comparison to analytical theory or Monte Carlo simulations
  - for many-body simulations
  - for thermal nonequilibrium situations
- easy visualization / animation:
  appeals to imagination

Disadvantages:

- slow
- cannot handle time scales >> 1 μs
- cannot handle space scales >> 100 nm

Isaac Newton (1643 – 1727)
1687: Philosophiae Naturalis Principia Mathematica

Molecular dynamics

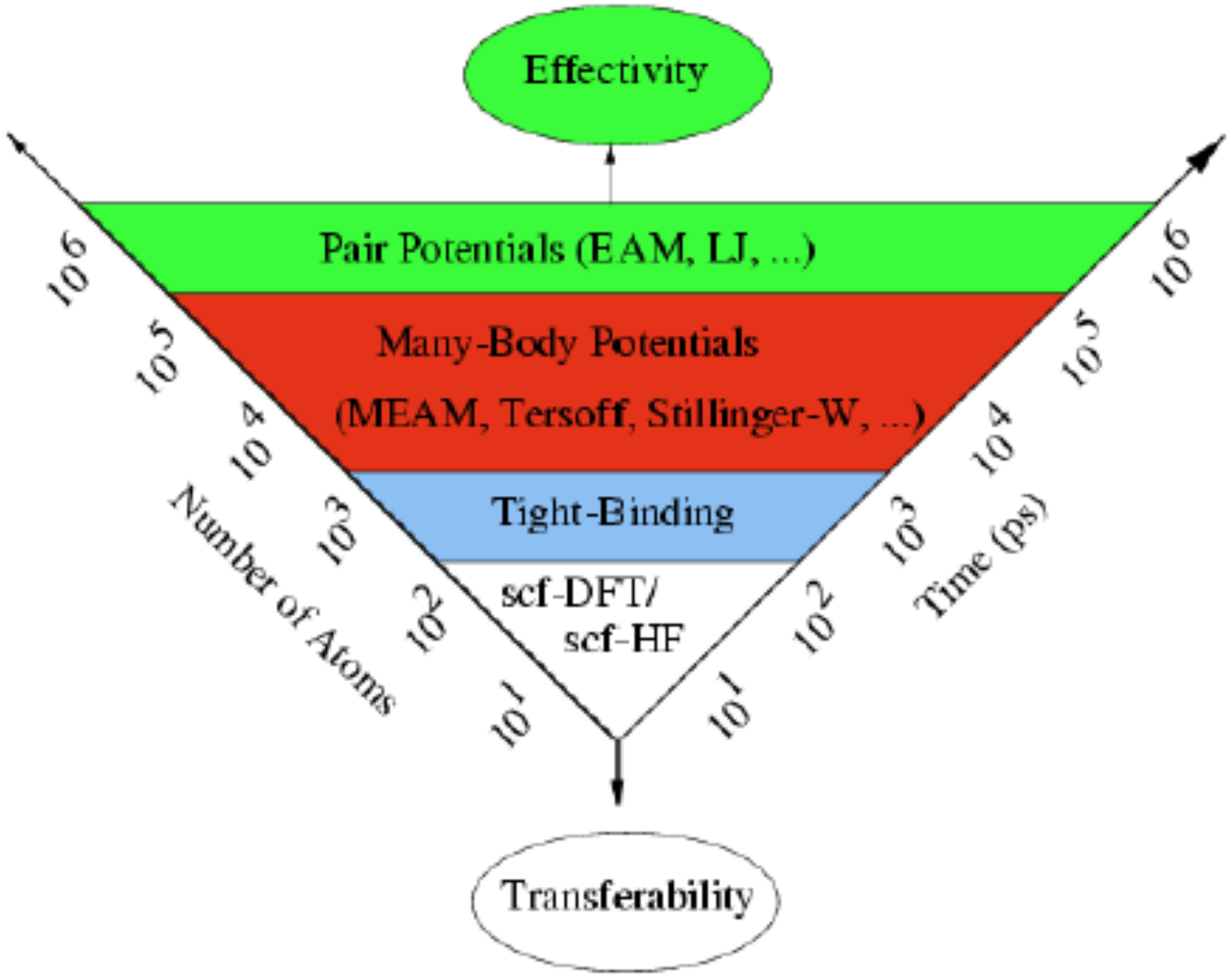
Solve Newton‘s equations

$$m\,\ddot{\mathbf{r}} = \mathbf{F}$$

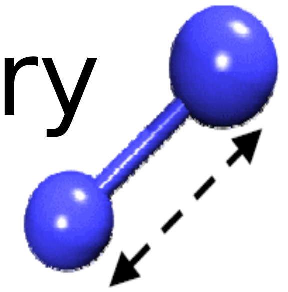
To consider:

- Potentials:
  atomic interaction forces
- Integrator:
  how to solve Newton‘s equations numerically
- Boundary conditions:
  how to understand (infinitely) large systems
  from the calculation of a finite system
- neighbour lists:
  optimize force calculation
- Detectors:
  how to extract physics information
- Initialization:
  how to reach equilibrium

Computational Efficiency of Interatomic Potentials

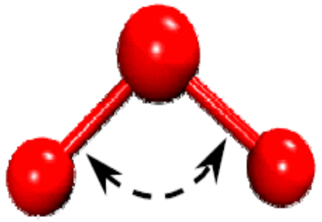


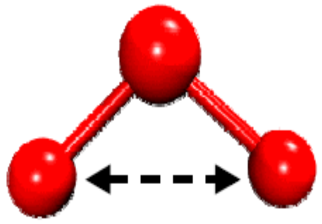
Molecular Dynamics for chemistry

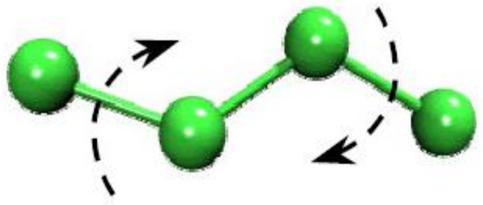


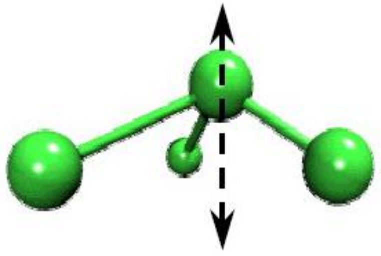
Forces between atoms

in classical simulation, forces derive from interatomic potentials (“force fields“)

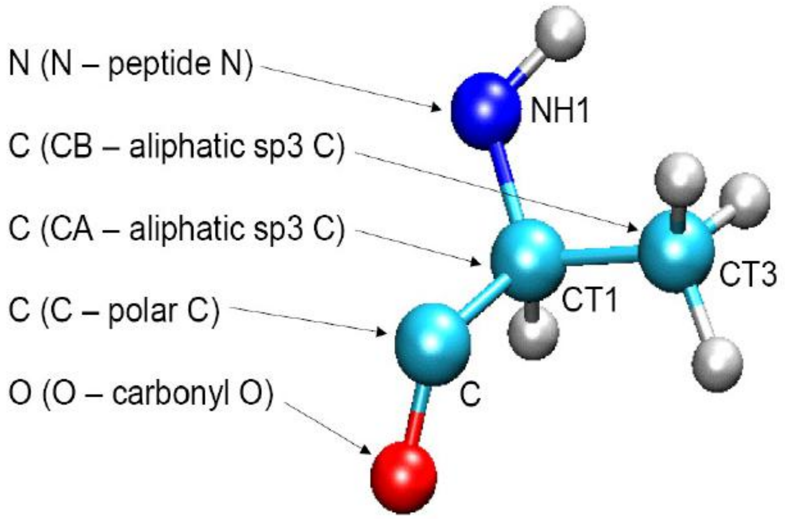






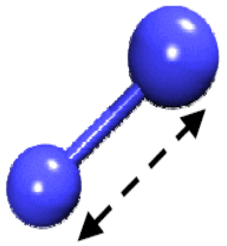


Atoms have different environments



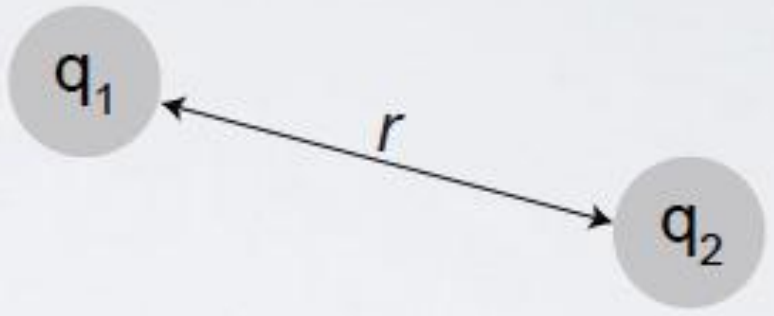
E_{bond} – bond-stretching potential energy

$$E_{bond} = \sum k_b (r - r_0)^2$$



Electrostatic interactions between ions

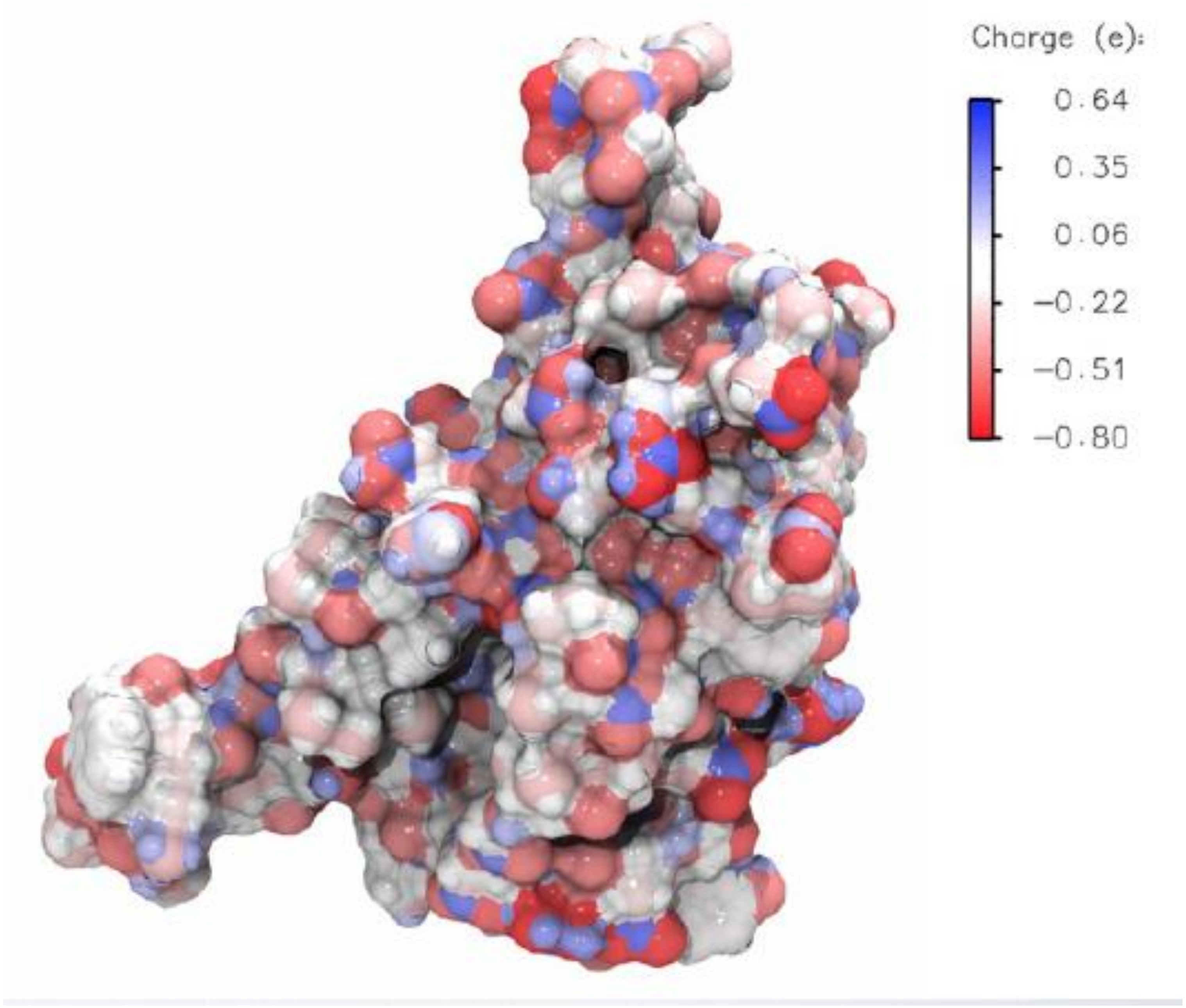
$$V(r) = \frac{q_1 q_2}{4\pi\epsilon_0}\frac{1}{r}$$



Problem: long-ranged

- **AMBER** (Assisted Model Building and Energy Refinement)
- **CHARMM** (Chemistry at HARvard Molecular Mechanics)
- **GROMOS** (Kraftfeld aus GROMACS – GROningen MOlecular Simulation package)
- **OPLS** (Optimized Potential for Liquid Simulations), Gesamtpotential:

$$V_{\mathrm{OPLS}} = \sum_{i<j} \left[\frac{q_i q_j e^2}{r_{ij}} + 4\epsilon_{ij} \left\{ \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right\} \right] f_{ij} + \sum_{\mathrm{Bindungen}} K_r \left(r - r_0 \right)^2$$
$$+ \sum_{\mathrm{Winkel}} K_\Theta \left(\Theta - \Theta_0 \right)^2 + \sum_i \frac{V_1^i}{2} \left[1 + \cos(\phi_i) \right] + \frac{V_2^i}{2} \left[1 - \cos(2\phi_i) \right] + \frac{V_3^i}{2} \left[1 + \cos(3\phi_i) \right]$$

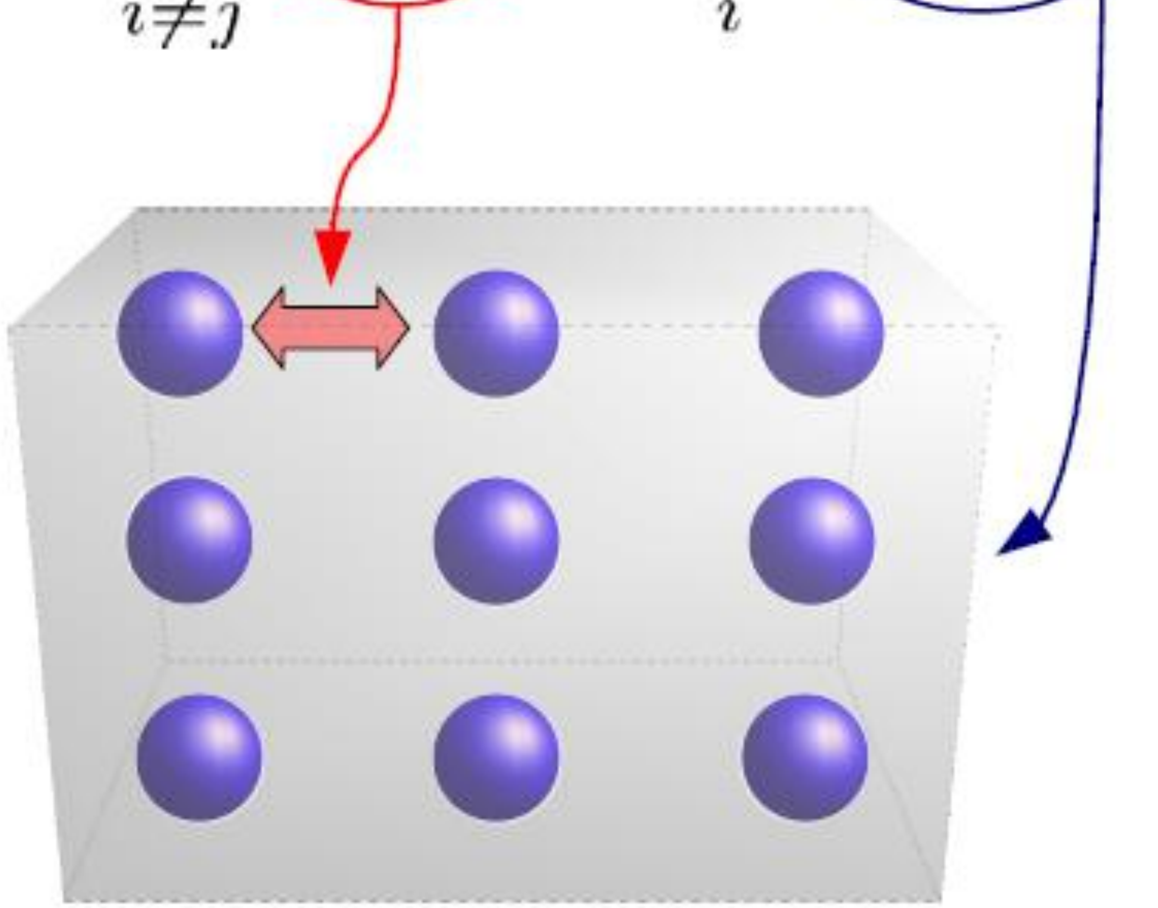


A large protein: BSA

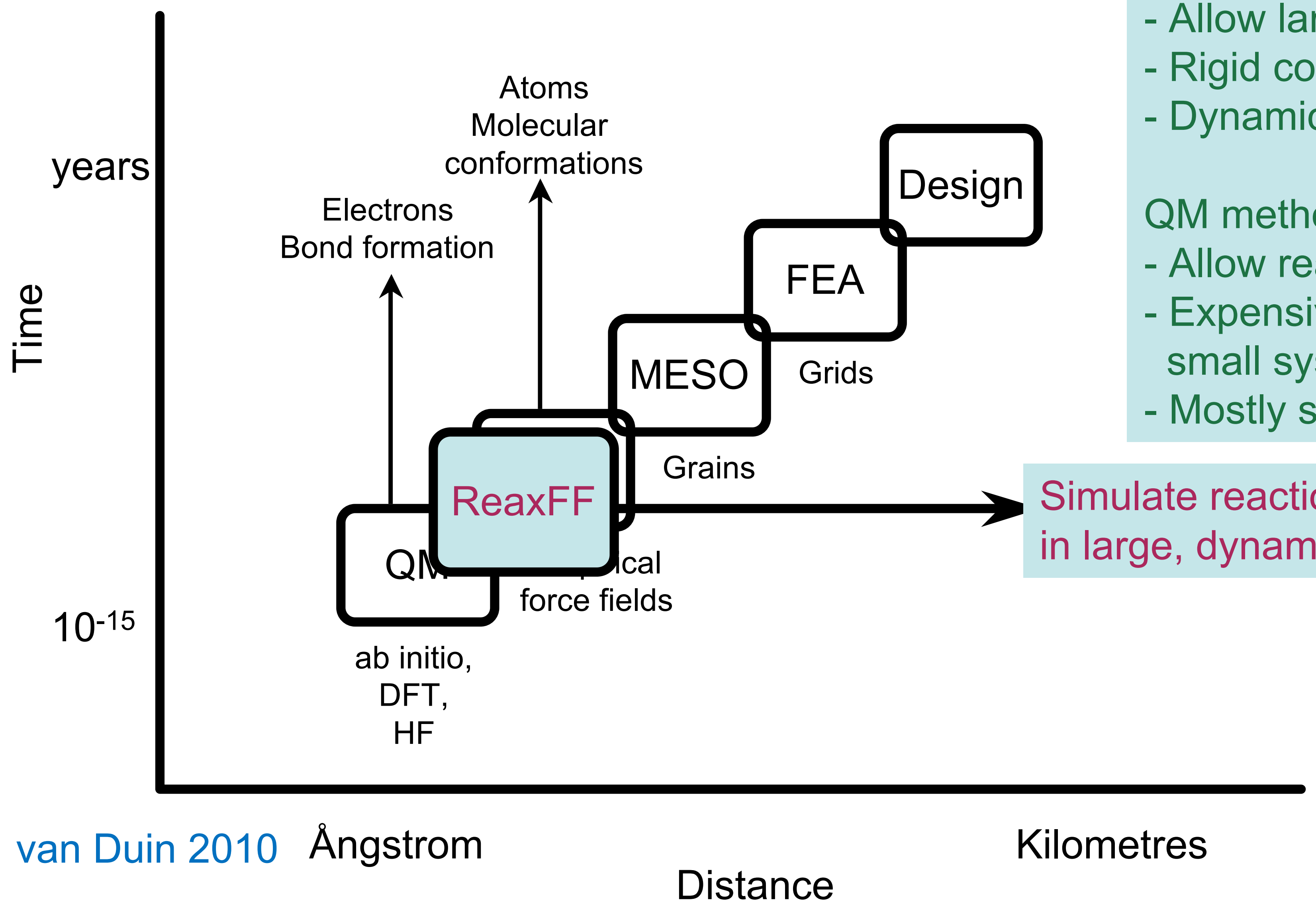
Forces between atoms: metals

Embedded-atom method

$$U = \frac{1}{2} \sum_{i \neq j} \phi(r_{ij}) + \sum_i \mathcal{F}(\rho_{h,i})$$



ReaxFF: background and rules

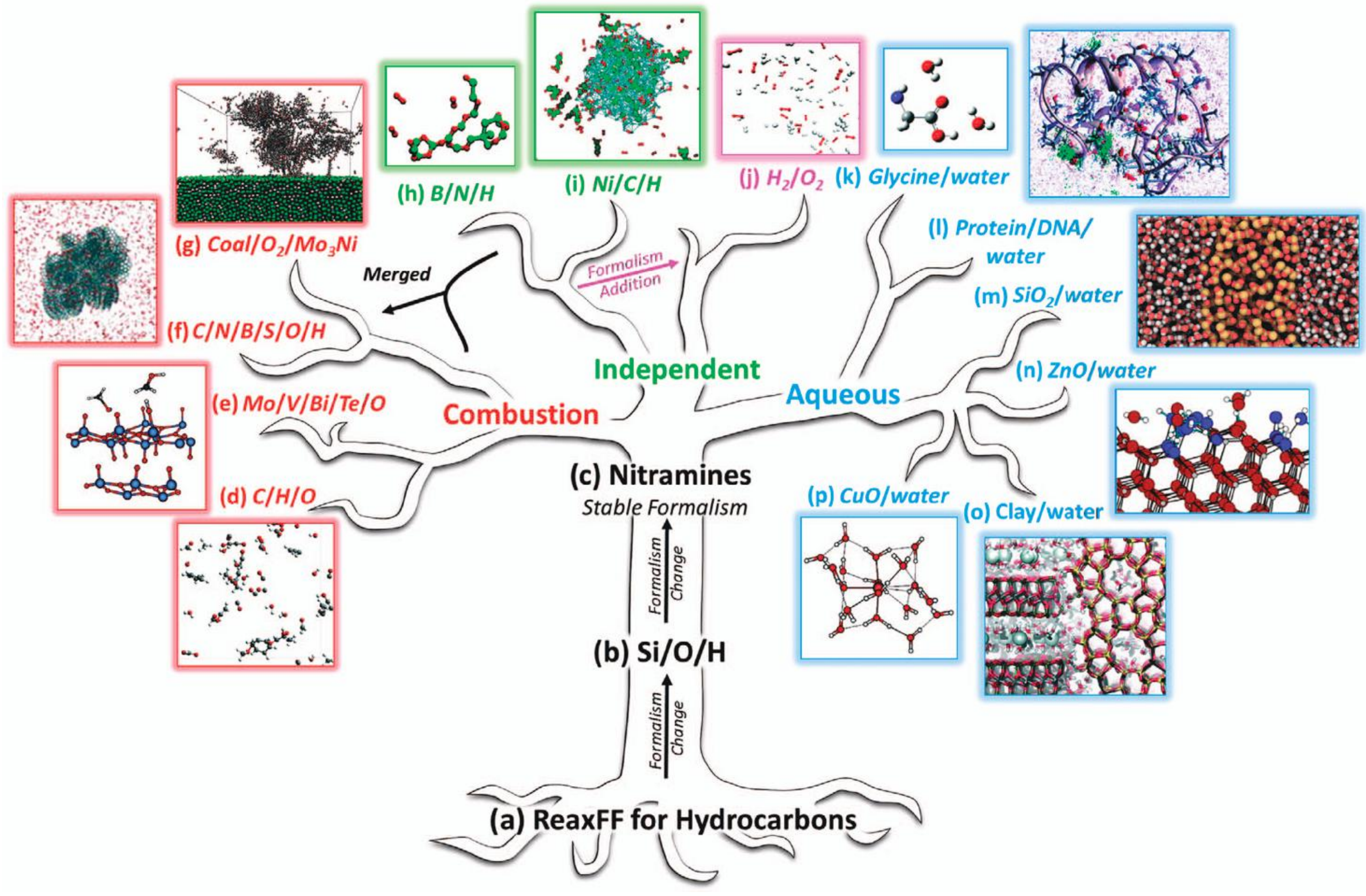


FF methods:
- Allow large systems
- Rigid connectivity
- Dynamics

QM methods:
- Allow reactions
- Expensive, only small systems
- Mostly static

Simulate reactions in large, dynamical systems

van Duin 2010



REAX development tree

van Duin 2010

Molecular dynamics

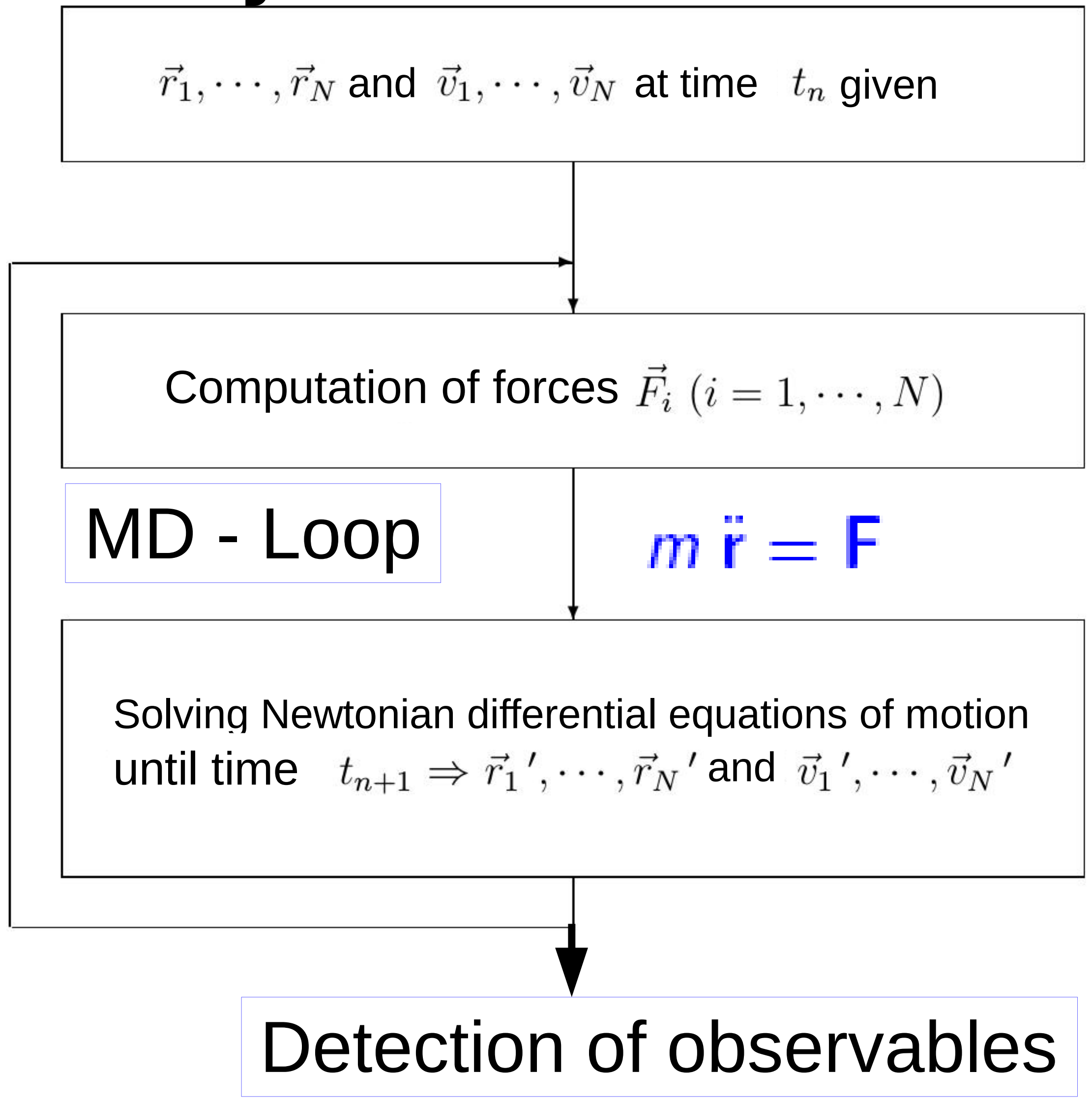
Solve Newton‘s equations $$m\,\ddot{\mathbf{r}} = \mathbf{F}$$

To consider:

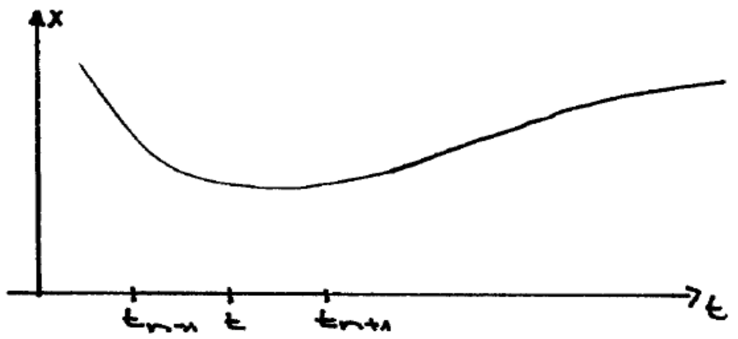
- Potentials:
  atomic interaction forces
- Integrator:
  how to solve Newton‘s equations numerically
- Boundary conditions:
  how to understand (infinitely) large systems
  from the calculation of a finite system
- neighbour lists:
  optimize force calculation
- Detectors:
  how to extract physics information
- Initialization:
  how to reach equilibrium

Algorithmic principle of molecular dynamics

- Initial positions and velocities of all particles (atoms, molecules or larger units)
- Interaction potentials result in forces
- Repeat cycle until relevant processes have completed


$\vec{r}_1, \cdots, \vec{r}_N$ and $\vec{v}_1, \cdots, \vec{v}_N$ at time t_n given
Computation of forces $\vec{F}_i \ (i = 1, \cdots, N)$
MD - Loop
$m\,\ddot{\mathbf{r}} = \mathbf{F}$
Solving Newtonian differential equations of motion until time $t_{n+1} \Rightarrow \vec{r}_1{}', \cdots, \vec{r}_N{}'$ and $\vec{v}_1{}', \cdots, \vec{v}_N{}'$
Detection of observables

Integrators



- Basic idea: infinitesimal integration of the Newtonian equation $\boldsymbol{F} = m\,\boldsymbol{a}$ or

$$\ddot{x} = f(x) \qquad \ddot{\vec{x}}_i = \sum_{j \neq i} \vec{F}_{ij}\left(\vec{x}_i, \vec{x}_j\right)$$

- Taylor approximation of $x(t)$ and $v(t)$

$$\begin{aligned} v\left(t_n + h\right) &= v\left(t_n\right) + h\dot{v}\left(t_n\right) + \ldots \\ x(t_n + h) &= x(t_n) + h\dot{x}(t_n) + \frac{1}{2}h^2\ddot{x}(t_n) + \ldots \end{aligned}$$

- Euler prediction for t_n + timestep h with force f_n

$$\begin{aligned} v_{n+1} &= v_n + hf_n \\ x_{n+1} &= x_n + hv_n + \frac{1}{2}h^2 f_n \end{aligned}$$

- Integrators improve with oder of approximation, intermediate steps and inclusion of correction terms (using new position for velocity)
- Velocity Verlet (optimal for MD)

$$\begin{aligned} x_{n+1} &= x_n + hv_n + \frac{1}{2}h^2 f_n \\ v_{n+1} &= v_n + h\frac{f_n + f_{n+1}}{2} \end{aligned}$$

- Verlet

  - Verlet is derived from the following two Taylor series:

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \delta t\,\mathbf{v}(t) + \frac{1}{2}\delta t^2\mathbf{a}(t) + \ldots$$

$$\mathbf{r}(t-\delta t) = \mathbf{r}(t) - \delta t\,\mathbf{v}(t) + \frac{1}{2}\delta t^2\mathbf{a}(t) + \ldots$$

  sum ‘em up and rearrange:

$$\mathbf{r}(t+\delta t) + \mathbf{r}(t-\delta t) = 2\mathbf{r}(t) + \delta t^2\mathbf{a}(t)$$

$$\mathbf{r}(t+\delta t) = 2\mathbf{r}(t) - \mathbf{r}(t-\delta t) + \delta t^2\mathbf{a}(t)$$

  - So we have an algorithm which essentially does:
  $\{\mathbf{r}(t), \mathbf{a}(t), \mathbf{r}(t-\delta t)\} \rightarrow \{\mathbf{r}(t+\delta t), \mathbf{a}(t+\delta t)\}$

  - However, the velocities are missing; these can be calculated from

$$\mathbf{v}(t) = \frac{\mathbf{r}(t+\delta t) - \mathbf{r}(t-\delta t)}{2\delta t}$$

- Requirements for a good MD algorithm

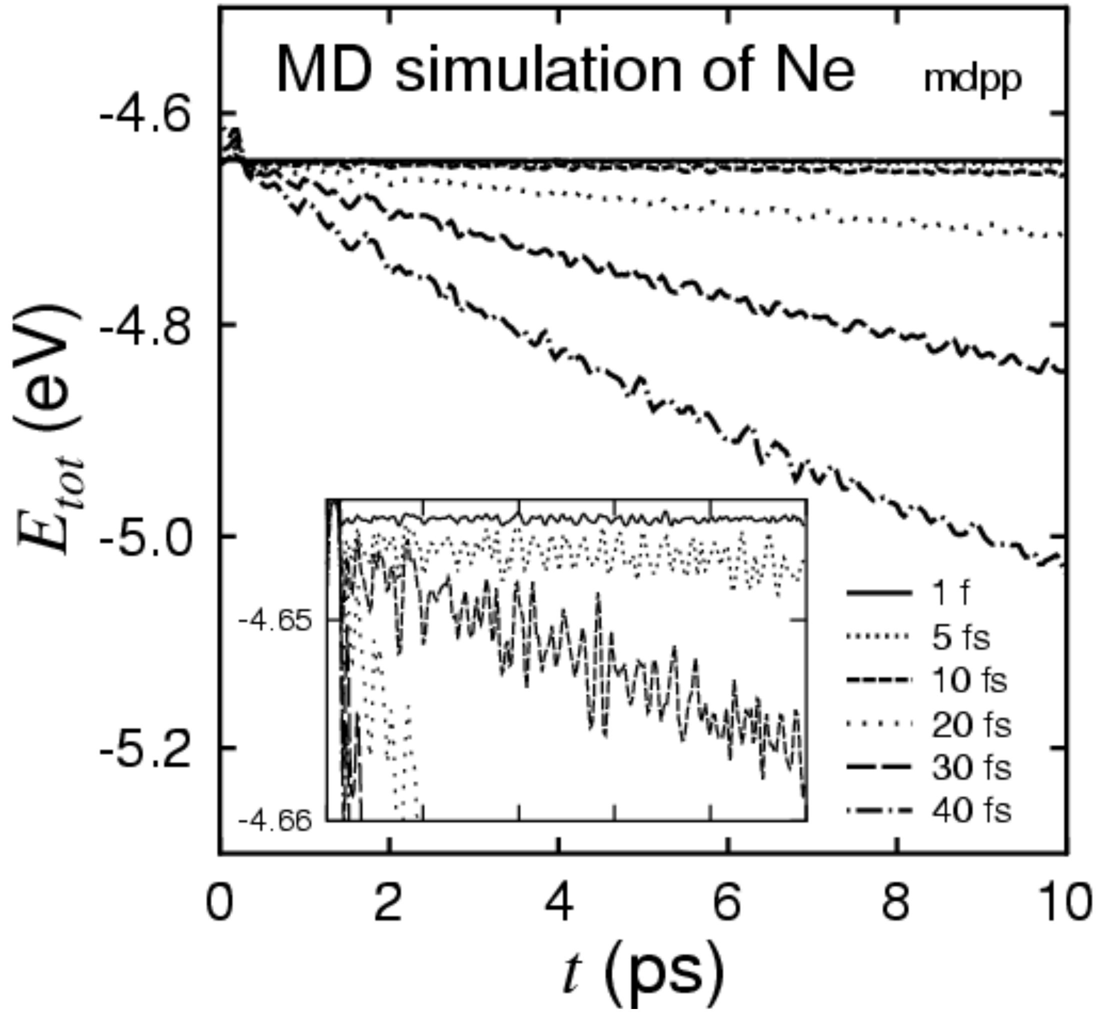
(a)	fast	(not that important)
(b)	takes little memory	(important)
(b)	allows a long time step δt	(important)
(c)	reproduces the correct path	(see below)
(d)	conserves energy (and is reversible: $\delta t \rightarrow -\delta t \Rightarrow$ back to original state)	(very important)
(f)	easy to implement	(not that important)
(g)	only one force evaluation/time step	(important for complex V)

effect of time step
on energy conservation

example: fcc Ne

rule of thumb:
Δ x < $r_{NN}/20$
during one time step

usually Δ t = a few fs



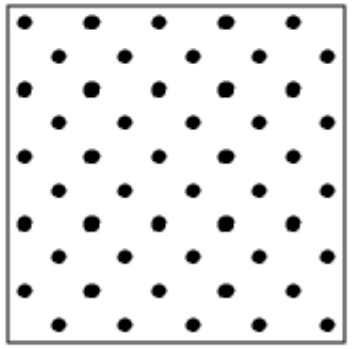
Molecular dynamics

Solve Newton's equations $m\ddot{\mathbf{r}} = \mathbf{F}$

To consider:

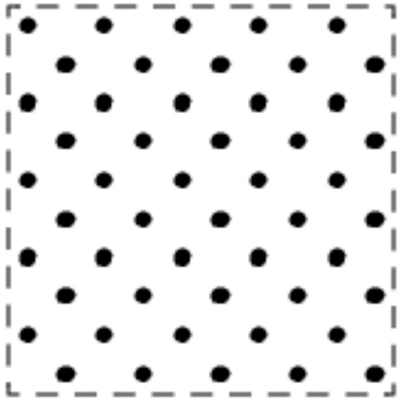
- Integrator:
  how to solve Newton's equations numerically
- Boundary conditions:
  how to understand (infinitely) large systems
  from the calculation of a finite system
- neighbour lists:
  optimize force calculation
- Detectors:
  how to extract physics information
- Initialization:
  how to reach equilibrium
- Potentials – atomic interaction forces

simulation cell

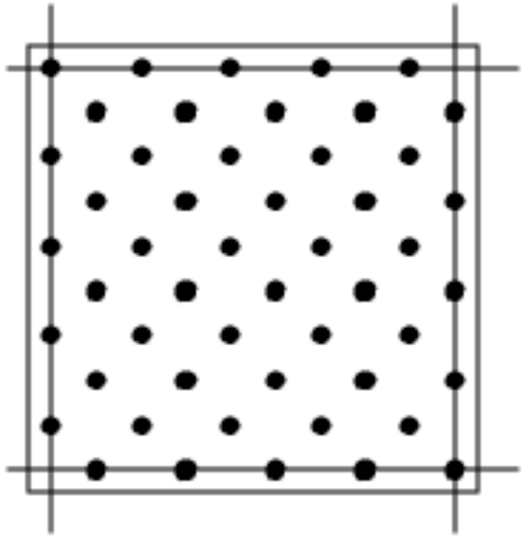


- Problem: what should we do with the atoms at the borders.

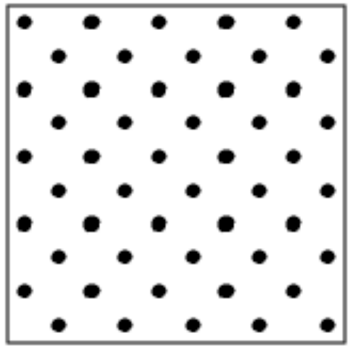
  1. Nothing => "free" boundaries



2. Fix the boundary atoms:

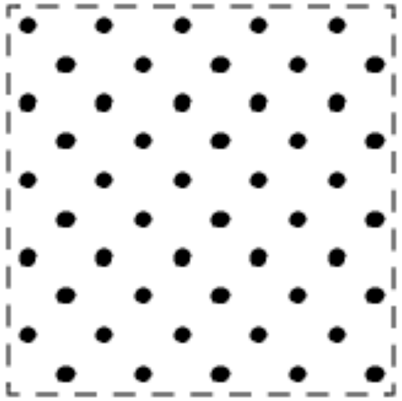


simulation cell

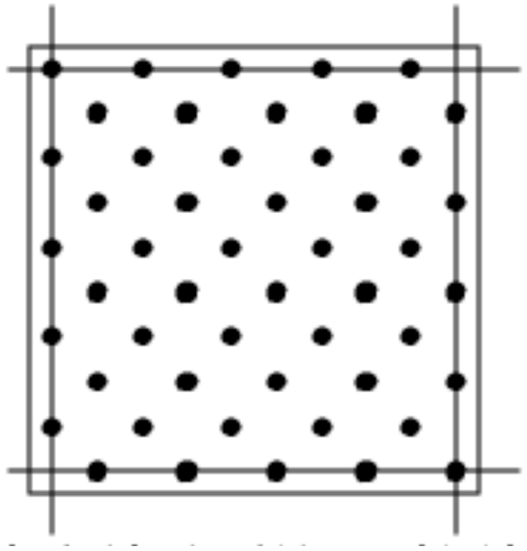


- Problem: what should we do with the atoms at the borders.

1. Nothing => “free” boundaries

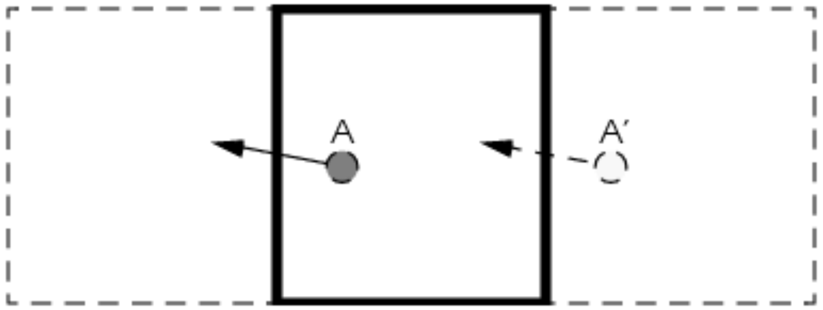


2. Fix the boundary atoms:

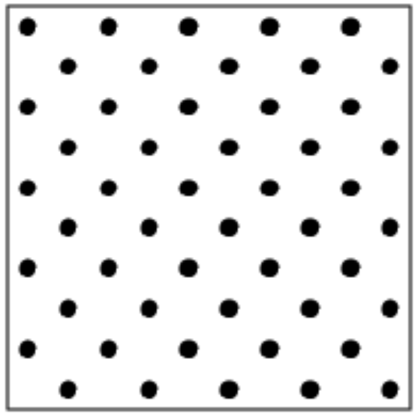


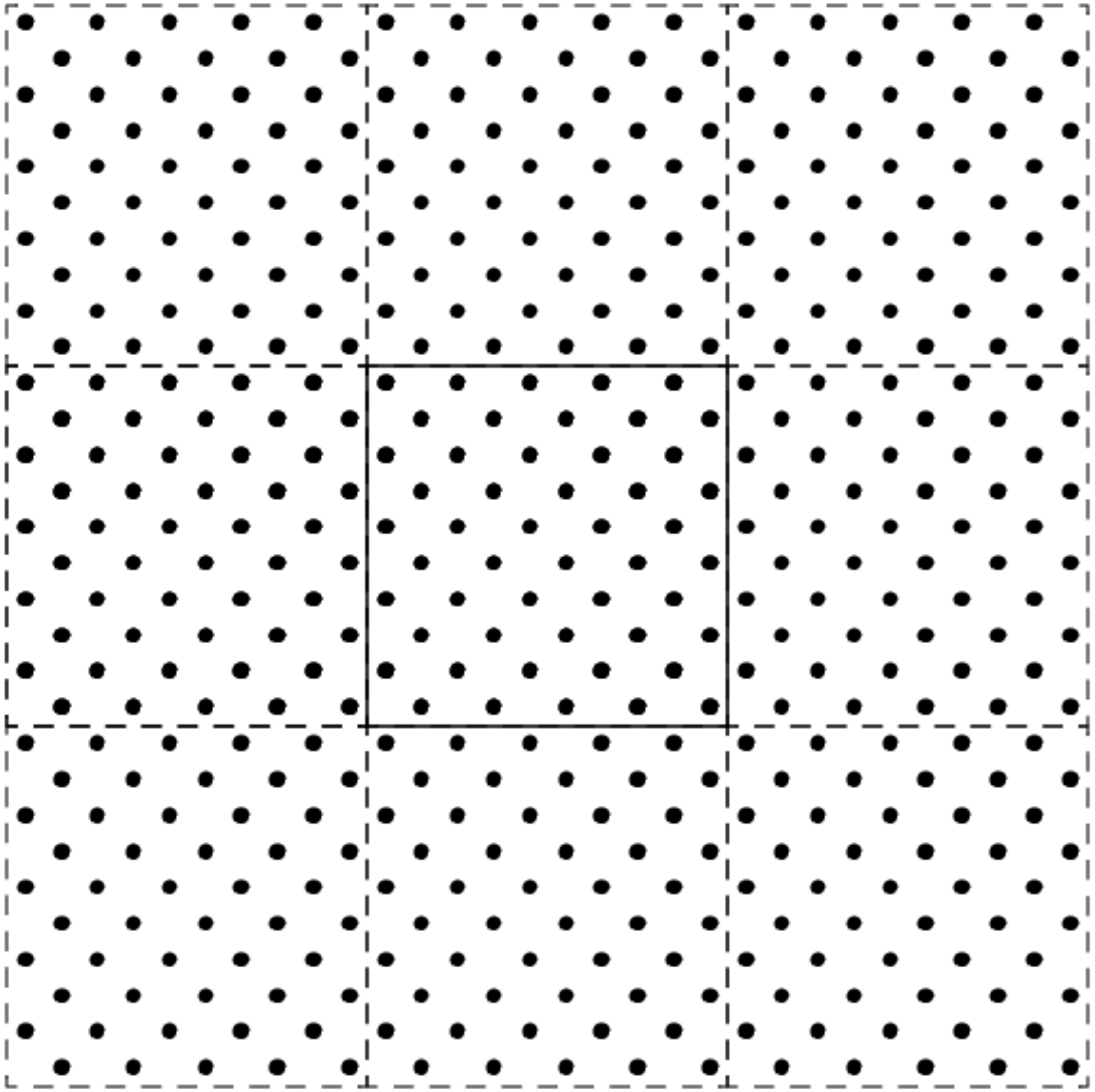
Periodic boundaries

1. An atom which passes over the cell boundary comes back on other side:



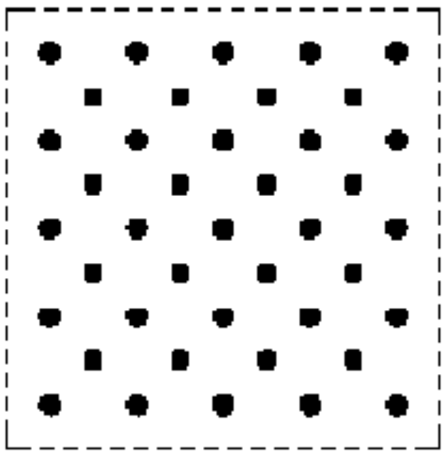
periodic boundaries



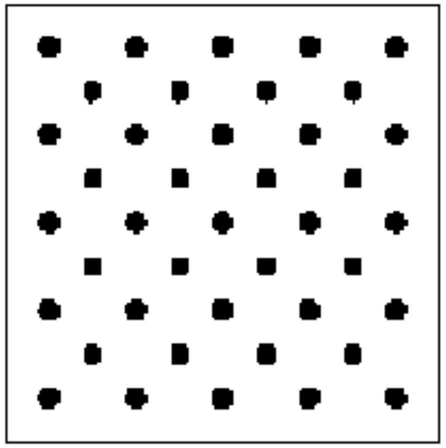


Boundary conditions

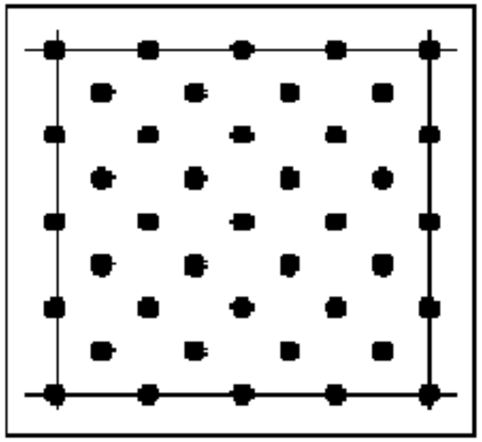
- (laterally) periodic boundary conditions
- free boundaries
-



free
for clusters

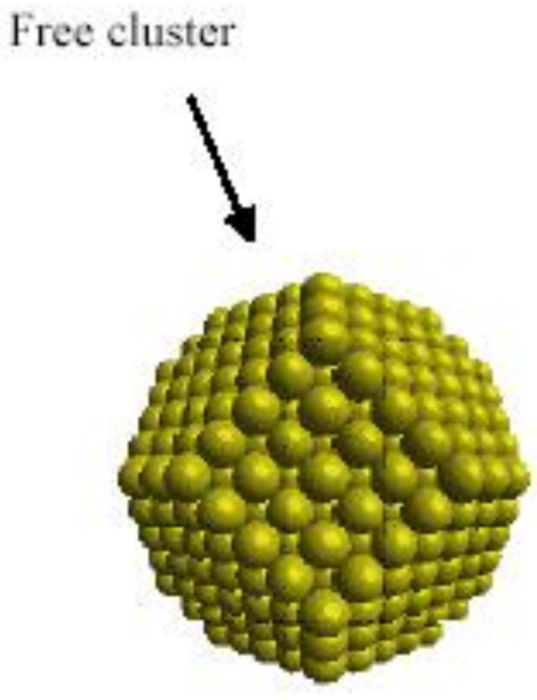


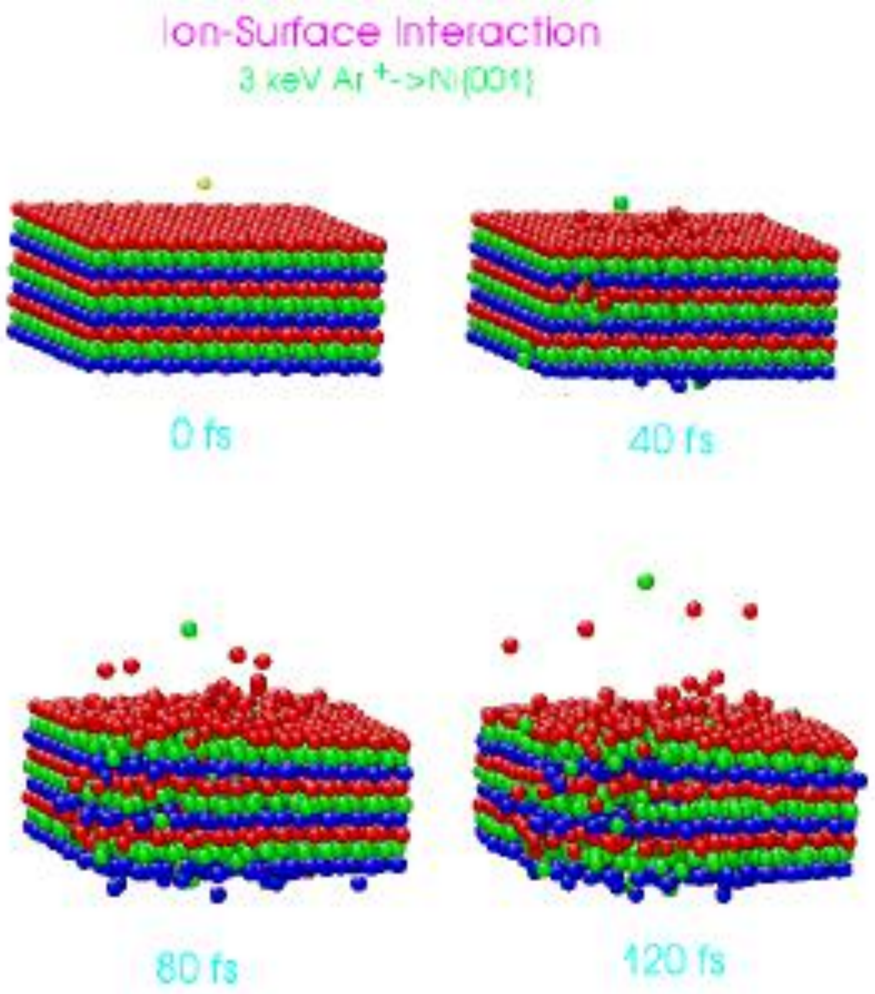
periodic
for solids



fixed
???

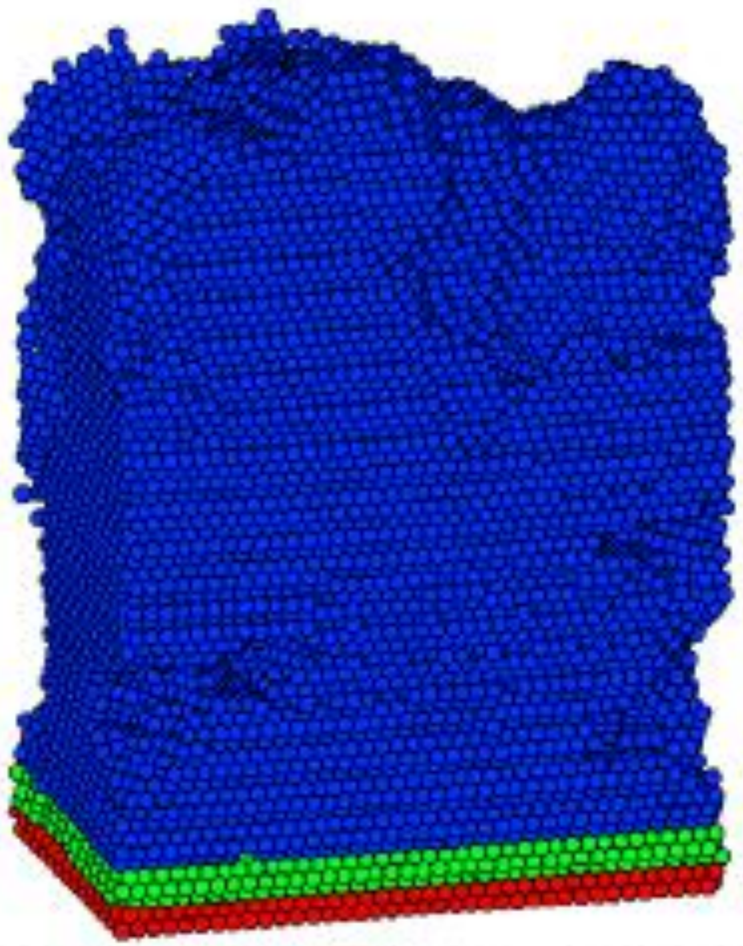
Free (open) boundaries





keV particle bombardment, by Barbara Garrison
http://galilei.chem.psu.edu/Research_bmb.html

laterally periodic boundaries


Cluster deposition film growth, by Dongare et al. Periodic boundary conditions in the directions parallel to the substrate, rigid and constant T layers at the bottom.

Molecular dynamics

Solve Newton‘s equations

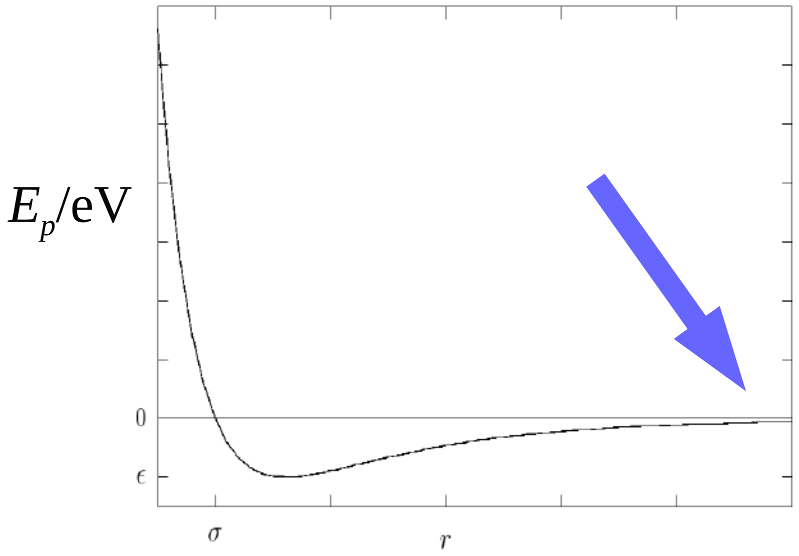
$$m\,\ddot{\mathbf{r}} = \mathbf{F}$$

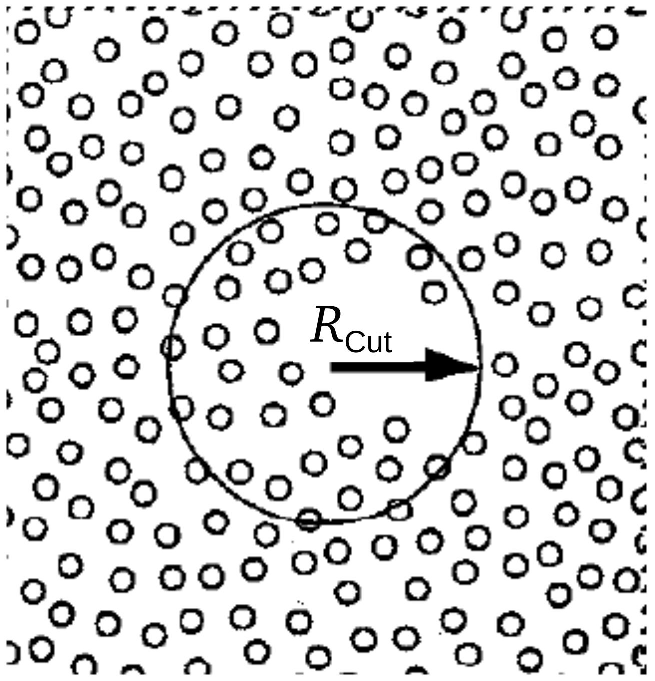
To consider:

- Potentials:
  atomic interaction forces
- Integrator:
  how to solve Newton‘s equations numerically
- Boundary conditions:
  how to understand (infinitely) large systems
  from the calculation of a finite system
- neighbour lists:
  optimize force calculation
- Detectors:
  how to extract physics information
- Initialization:
  how to reach equilibrium

Neighbor lists

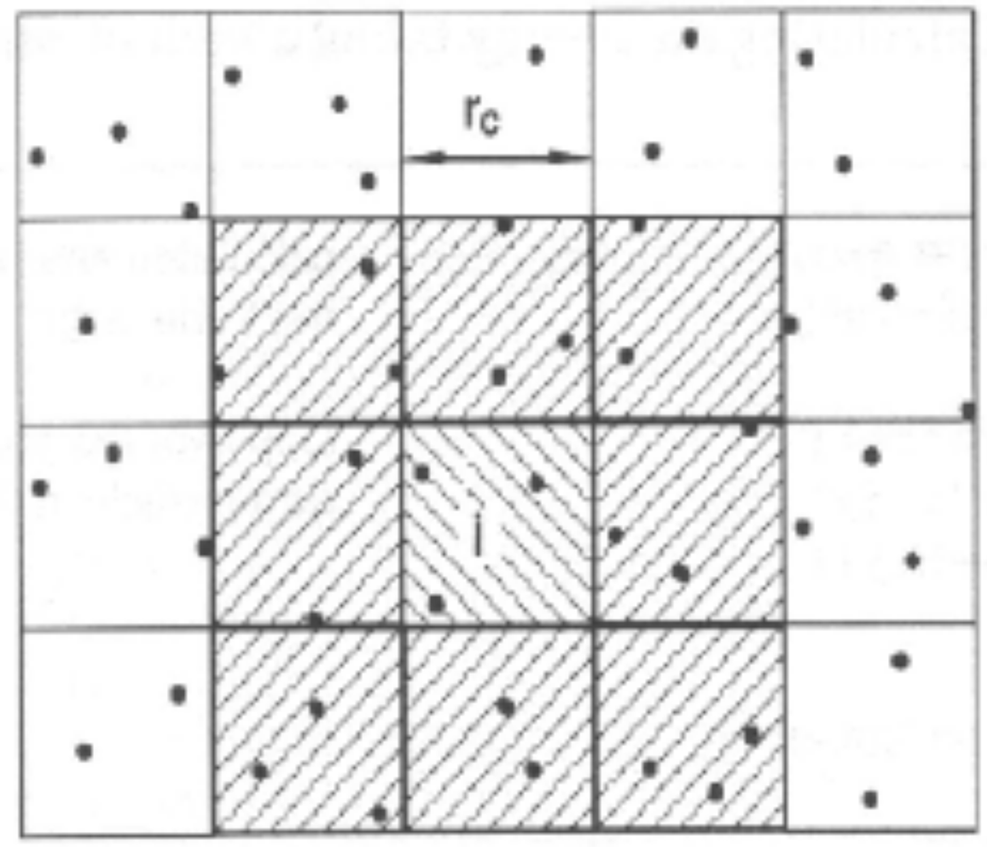
- Interaction potentials as Lennard Jones, EAM, force fields etc.
- Near neighbor interactions only
- Potential function vanishes outside cut-off-radius R_{Cut}
- Reduce force computation from all with all particles, $o(N^2)$, to $N*Neighbors$
- Key to effective simulation in cells and parallelization
- Sending cells or groups of cells to different processors (CPUs, GPUs) or even computers
- Parallelization implemented e.g. in LAMMPS [http://lammps.sandia.gov/]





[aus Kafemanns Diplomarbeit]

linked-cell algorithm



Molecular dynamics

Solve Newton‘s equations $$m\,\ddot{\mathbf{r}} = \mathbf{F}$$

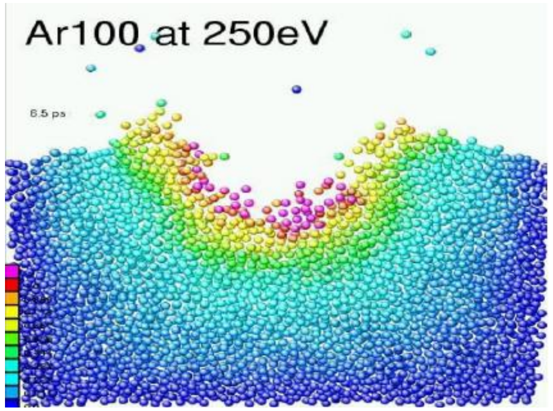
To consider:

- Potentials:
  atomic interaction forces
- Integrator:
  how to solve Newton‘s equations numerically
- Boundary conditions:
  how to understand (infinitely) large systems
  from the calculation of a finite system
- neighbour lists:
  optimize force calculation
- Detectors:
  how to extract physics information
- Initialization:
  how to reach equilibrium

Detectors and visualization tools

- Coordinates and velocities of all atoms
- Computation of temperature via kinetic energy and pressure via kenetic energy and forces as local averages
- Crystallinity detectors like Ackland, common neighbor ananlysis etc. can be implemented as required
- Atomic properties can be visualized with graphic tools as „Ovito“ and „VMD“ (visulaisation of molecules)
- Atomic trajectories can be reduced to global observables.

$$
\begin{aligned}
E_{\mathrm{kin}} &= \frac{1}{2m}\sum_{i=1}^{N}\vec{p}_i^{\,2} \\
\langle E_{\mathrm{kin}}\rangle_{NVT} &= \frac{3}{2}NkT
\end{aligned}
\longrightarrow
$$



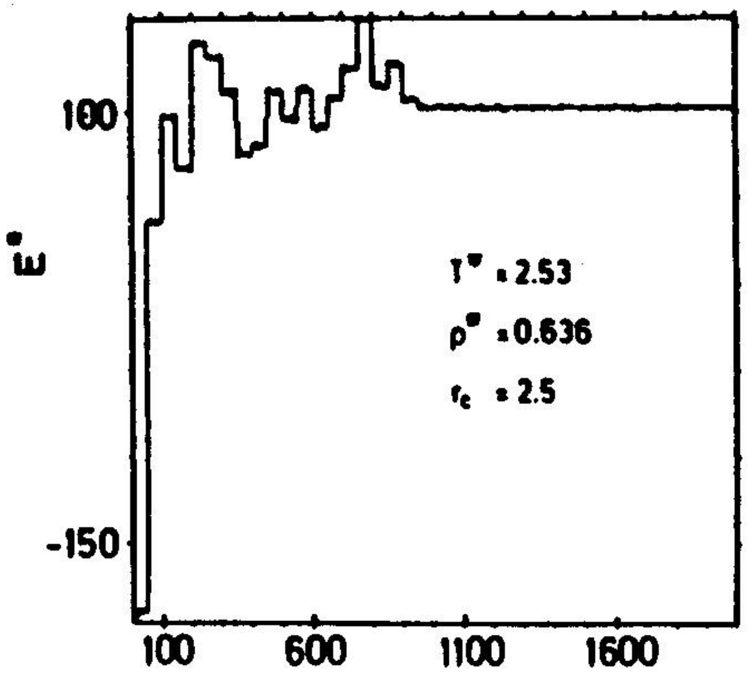
Molecular dynamics

Solve Newton‘s equations $$m\,\ddot{\mathbf{r}} = \mathbf{F}$$

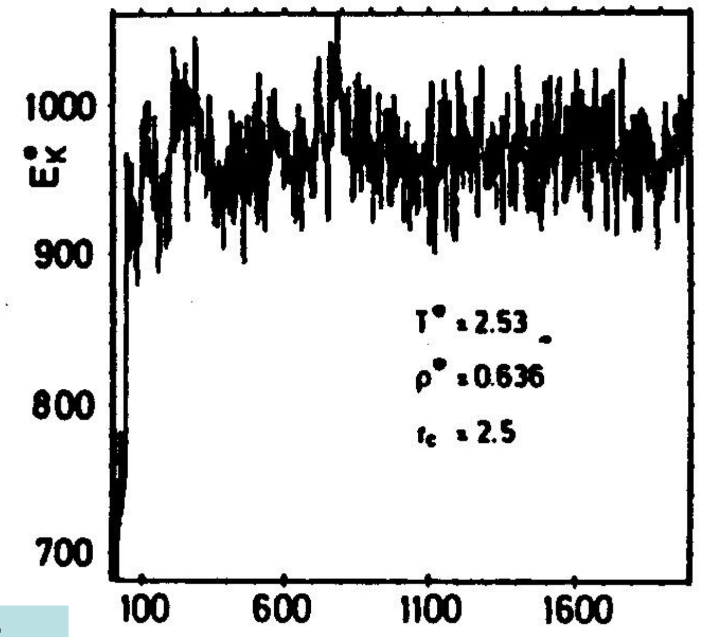
To consider:

- Potentials:
  atomic interaction forces
- Integrator:
  how to solve Newton‘s equations numerically
- Boundary conditions:
  how to understand (infinitely) large systems
  from the calculation of a finite system
- neighbour lists:
  optimize force calculation
- Detectors:
  how to extract physics information
- Initialization:
  how to reach equilibrium

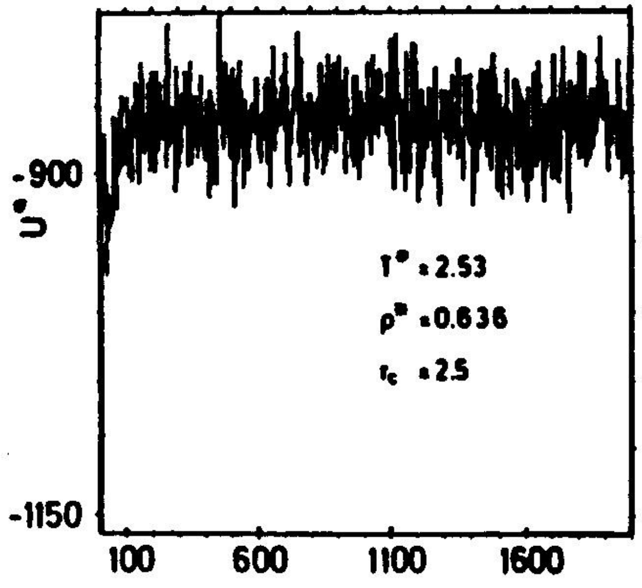
Initialization of a LJ system: velocity scaling at every 50 time steps



E_{tot}



E_{kin}



E_{pot}

Theoretical tools

- **Molecular dynamics**

Solve Newton‵s equations.

Advantages:

- as realistic as possible in comparison to analytical theory or Monte Carlo simulations
  - for many-body simulations
  - for thermal nonequilibrium situations
- easy visualization / animation:
  appeals to imagination

Disadvantages:

- slow
- cannot handle time scales >> 1 μs
- cannot handle space scales >> 100 nm

Isaac Newton (1643 – 1727)
1687: Philosophiae Naturalis
Principia Mathematica

Conclusions

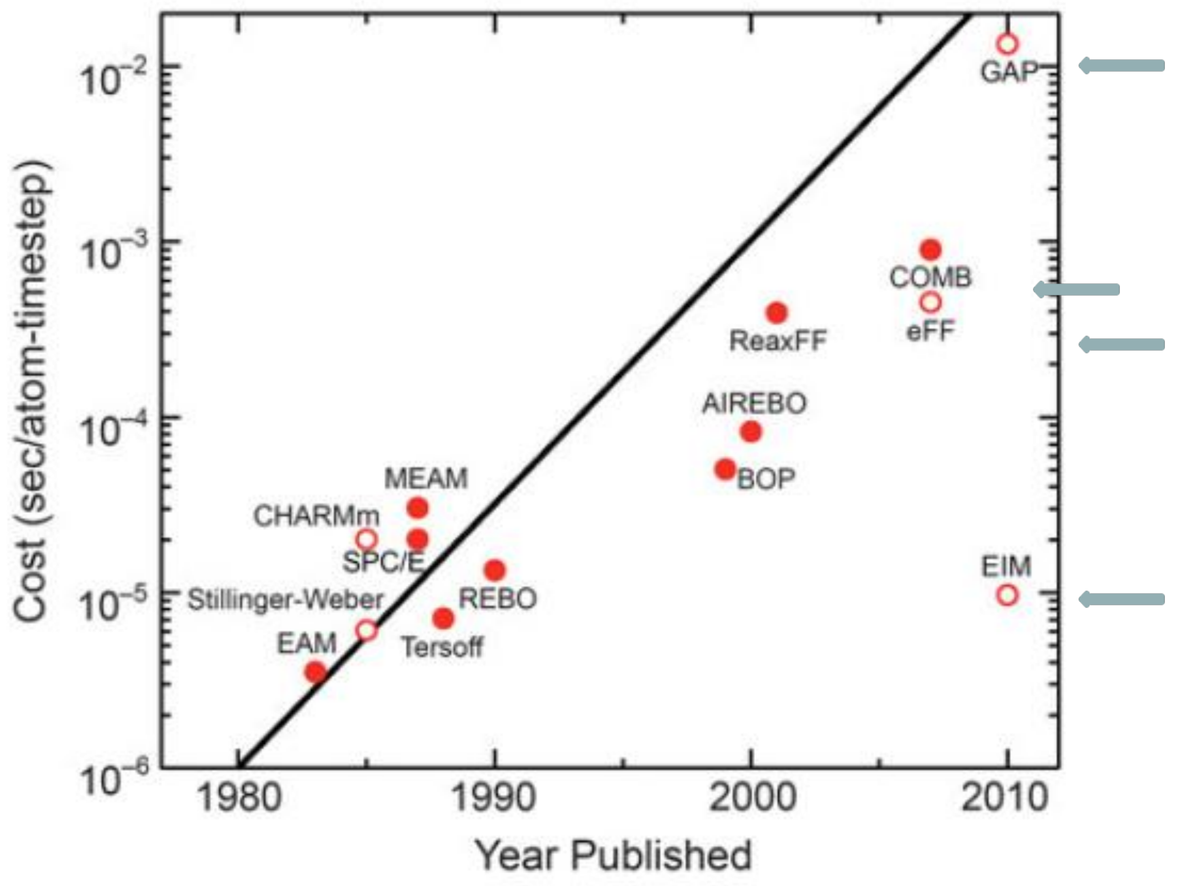
Molecular dynamics simulation provides:

- detailed information on atomistic level
- insight into processes
- space scales: 100 nm
- time scales: ns ... (µs)
- reliability: depends on interatomic potential

Besides hardware progress (Moore‘s law)
progress in models

- interaction potentials
- acceleration

Over the years potentials have become more sophisticated



„near quantum accuracy“

charge transfer
chemical reactions

classical „force fields“

DFT: 100 – 1000 sec /(atom timestep) for ~1000 electron system and do not scale linearly with number of atoms or electrons

Plimpton 2012