ARCHIE-WeSt

Basic Principles of Molecular Dynamics (MD) theory

Dr Karina Kubiak - Ossowska
High Performance Computing Support Team
Department of Physics

e-mail: karina.kubiak@strath.ac.uk

MD Definition

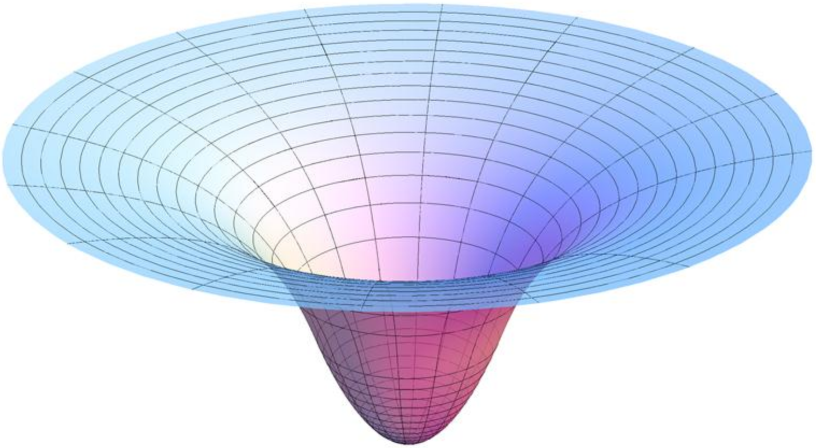
Molecular Dynamics is a computer simulation of physical movements of atoms and molecules. The trajectories of molecules and atoms are determined by numerically solving the classical equations of motion (Newton’s equations) for a system of interacting particles, where forces between the particles and **potential energy** are defined by **molecular mechanics force fields.**

This was first accomplished in **1957 and 1959** for a system of hard spheres by Adler. Now is applied mostly in **materials science and modeling of biomolecules.**

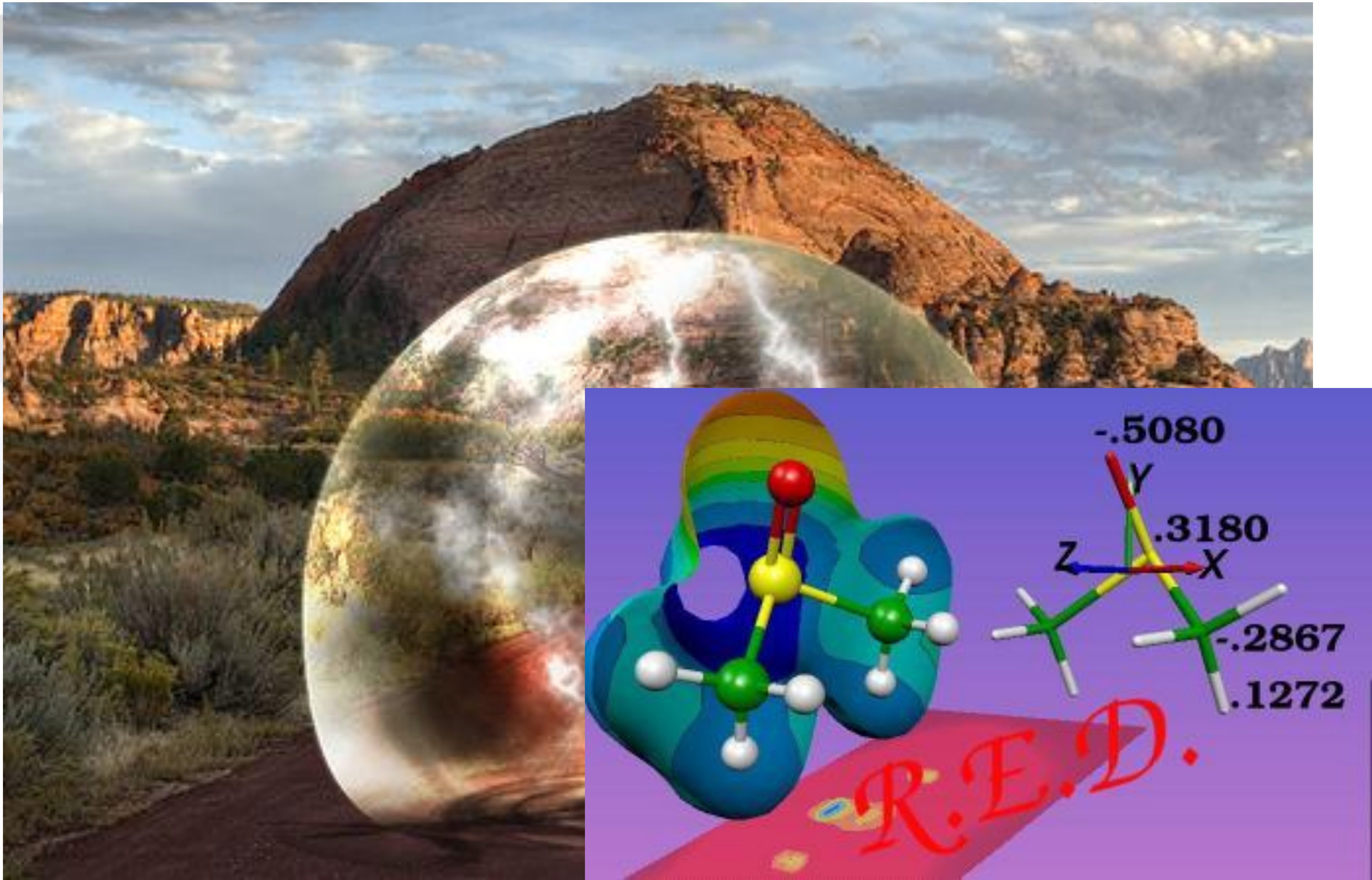
Definitions

Potential energy is associated with a set of forces that act on a body in a way that **depends only on the body position** in space. The set of forces can be considered as a force field. In other words potential energy is the energy of the object or the system due to its position. The SI unit for energy is the joule (J)

Force field is an area where the forces acts. It can be for example a gavitational force field around a uniform spherical body (picture)



Force field



Definitions

Molecular mechanics (MM) uses classical mechanics to model molecular systems. The potential energy of all systems in molecular mechanics is calculated using force fields. MM can be used to study small and large molecules with thousands or millions of atoms.

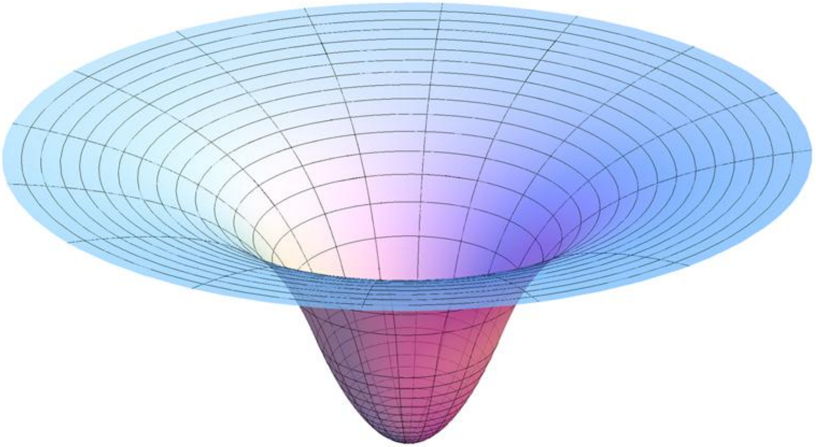
Basic principles of MM:

- Each atom is simulated as a single particle
- Each particle has an radius (typically the van der Waals radius), polarizability and constant net charge
- Bonded interactions are treated as “springs” with an equilibrium distance equal to the experimental or calculated bond

Molecular Mechanics

Molecular mechanics (MM) in MD is **used for energy minimization.** The force field is used as an optimization criterion and the **local minimum** is searched by an appropriate algorithm (conjugate gradient, steepest descent). The main aim is **to find the lowest (one of the lowest) energy conformation** of a molecule.

MM can be used for biomolecule optimization, design of (drug, small molecules) binding sities, protein folding kinetics …



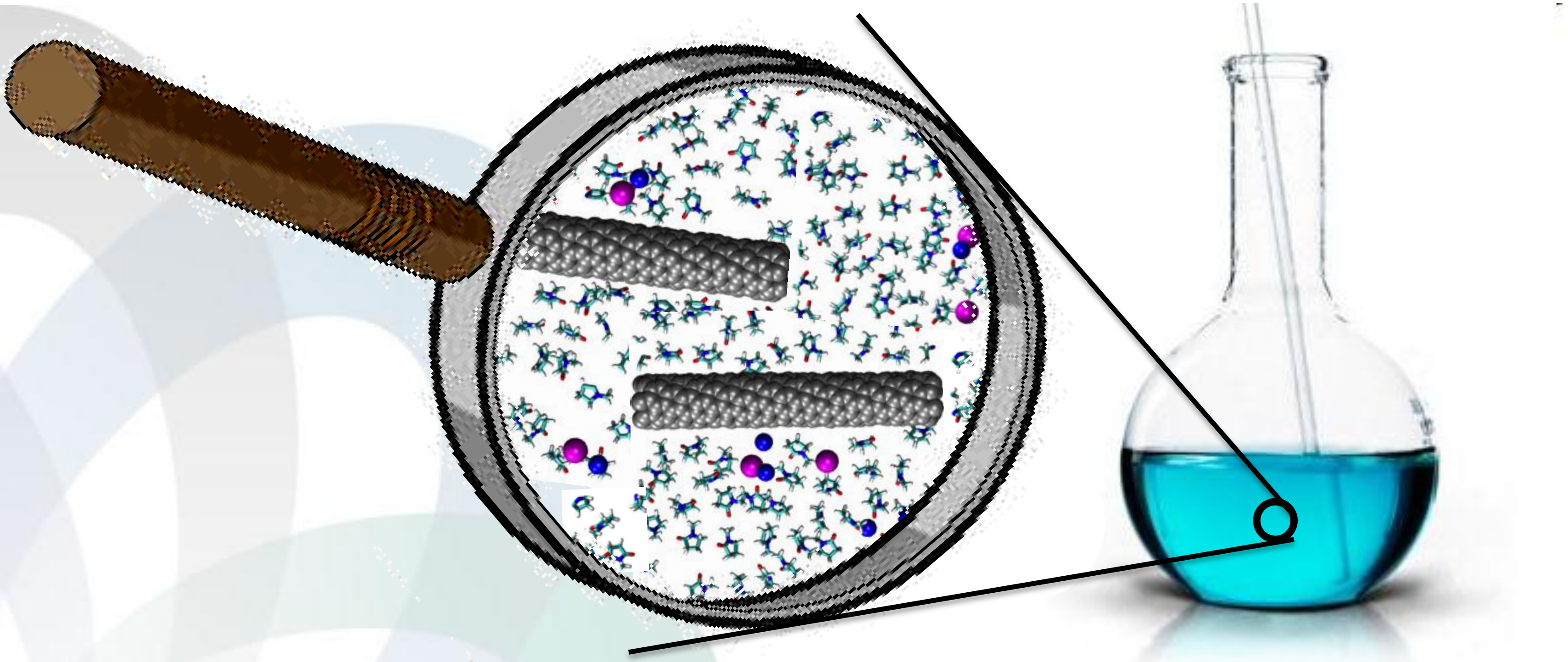
Molecular Dynamics

MD trajectory is a series of atoms positions (snapshots) in time.

MD gives an access to events which cannot be observed experimentally. We can see the molecule! **MD predict the protein dynamics, structure changes upon binding of ions and small molecules (like drugs), UV irradiation or other structure alteration (mutations).** It is possible to monitor protein **adsorption, protein aggregation, folding and unfolding, structure changes upon changing the environment**.

Theoretical methods are typically cheaper and faster than the experimental ones.

Molecular Dynamics



MD is a computational microscope. Understanding how things work on a detailed *molecular* level ranging from electronic structures to long-time phase behavior of molecules

**If one understands how things work,
one can manipulate them!**

MD limitations

We would like to simulate bigger and **bigger systems for the longer timescales** relevant to the timescales of natural processes. But ….

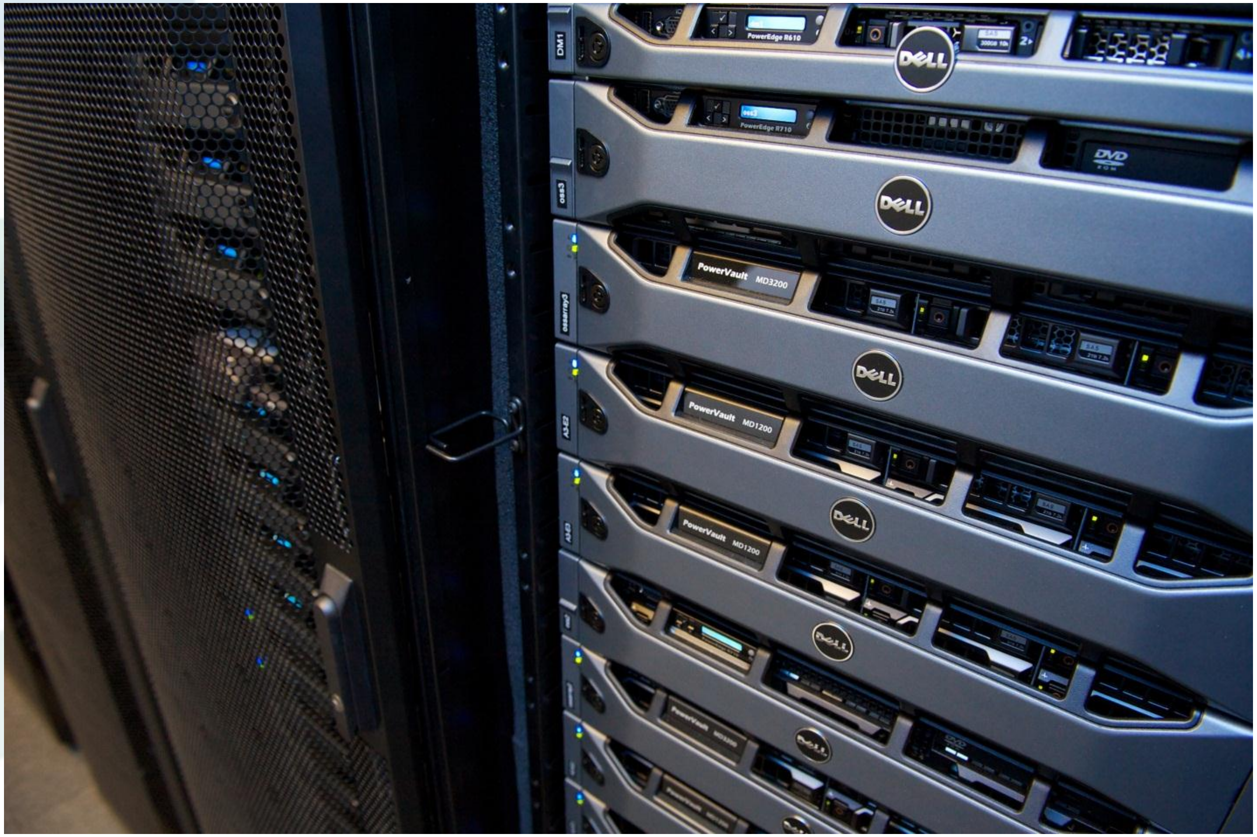
- The bigger system – the longer computation time
- The longer simulation timescale – the longer computation time.
- Explicit water – the longer computation time

MD simulations are computationally costly, they require hundreds or even millions of CPU years…..

There are methods to reduce the computational cost (timestep, SHAKE algorithm, cut-offs, PME, implicit solvent and … **H**igh **P**erformance **C**omputers (**HPC**s)

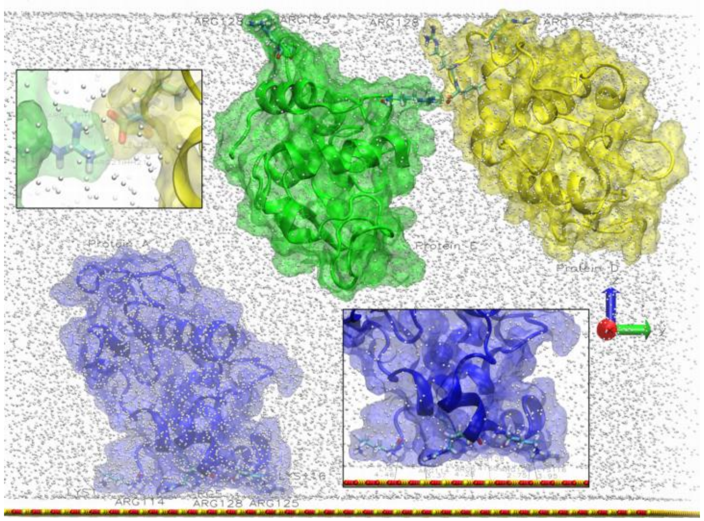
Molecular Dynamics

It is relatively **cheap and fast method** to verify experimental predictions and to propose hypothesis to be checked in the experiment. Requires a powerful computers like **ARCHIE-WeSt.**

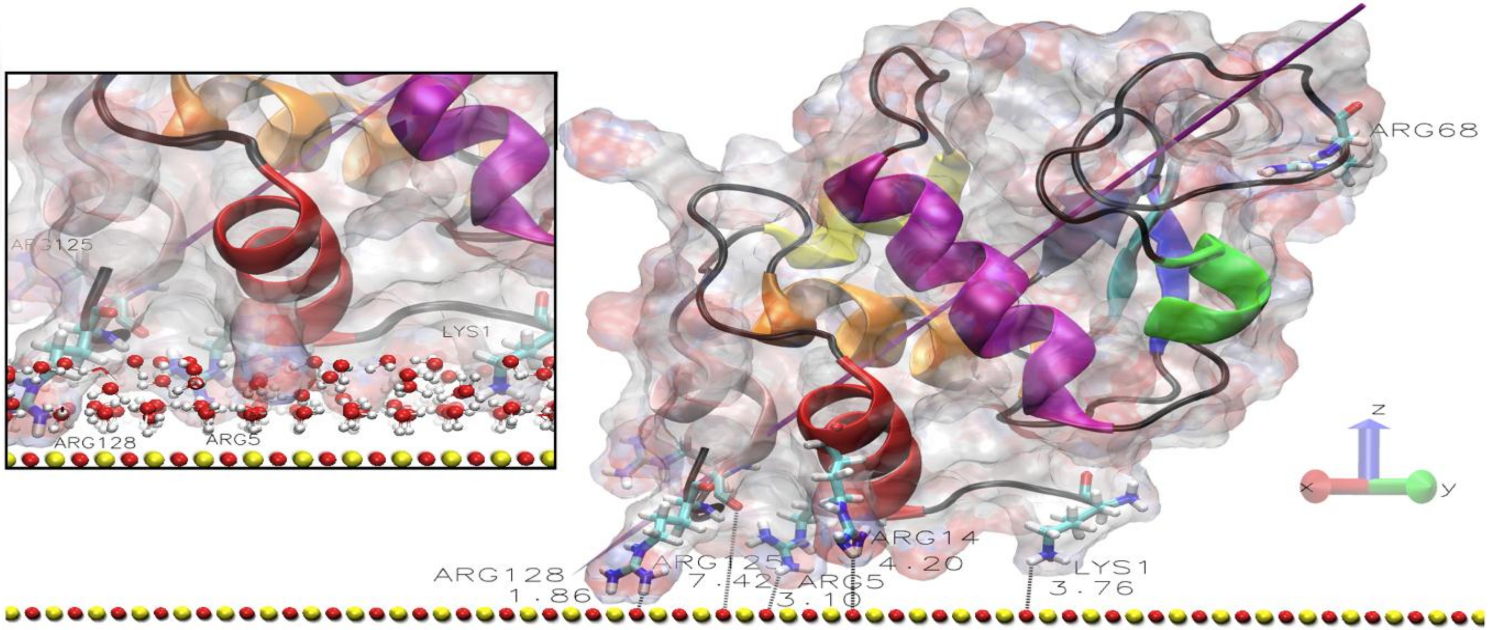


Sample Application:

Protein adsorption

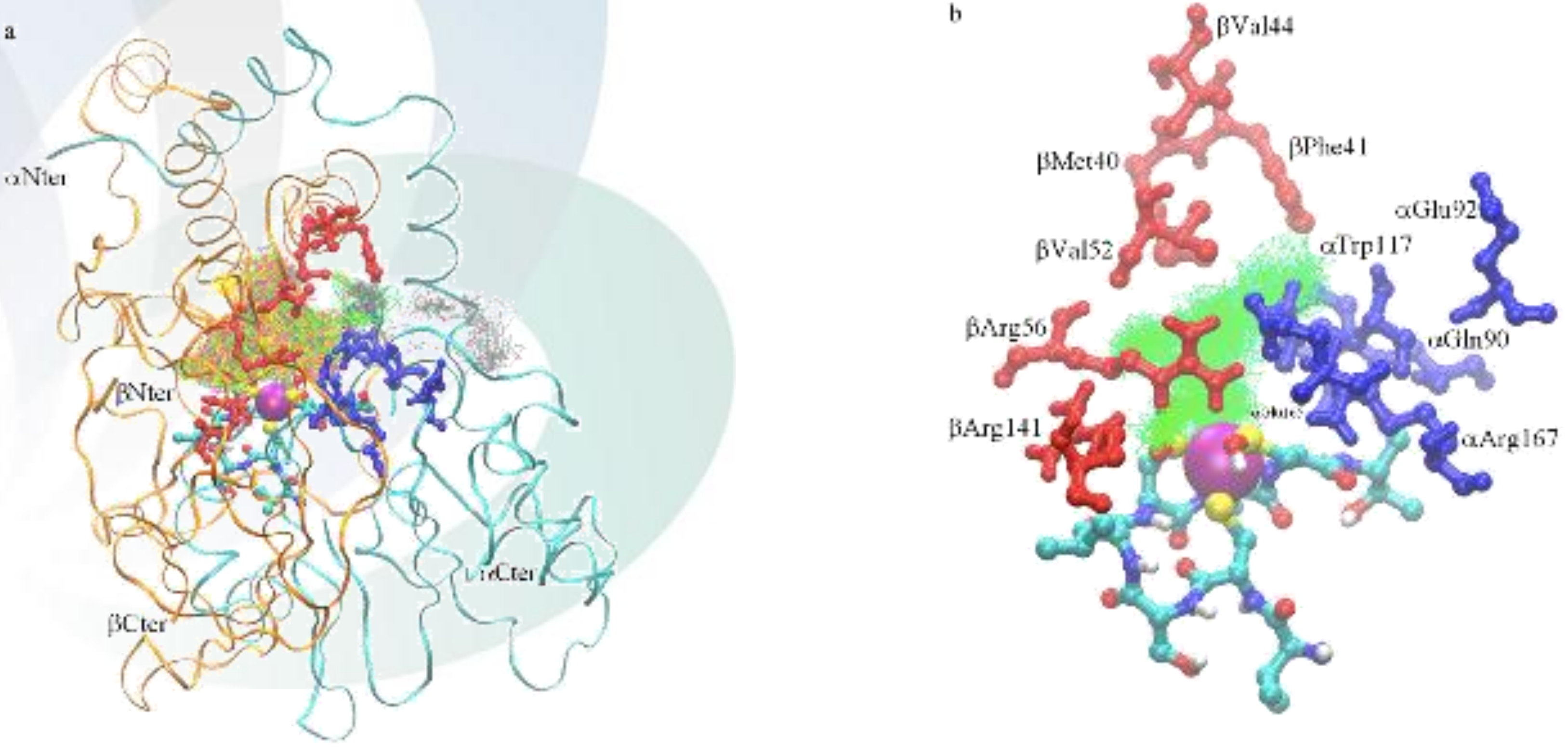


Thanks to the simulations we can see how the protein aggregates on atomistic level, describe the adsorption and the diffusion mechanism.



Sample application:
enzymatic activity

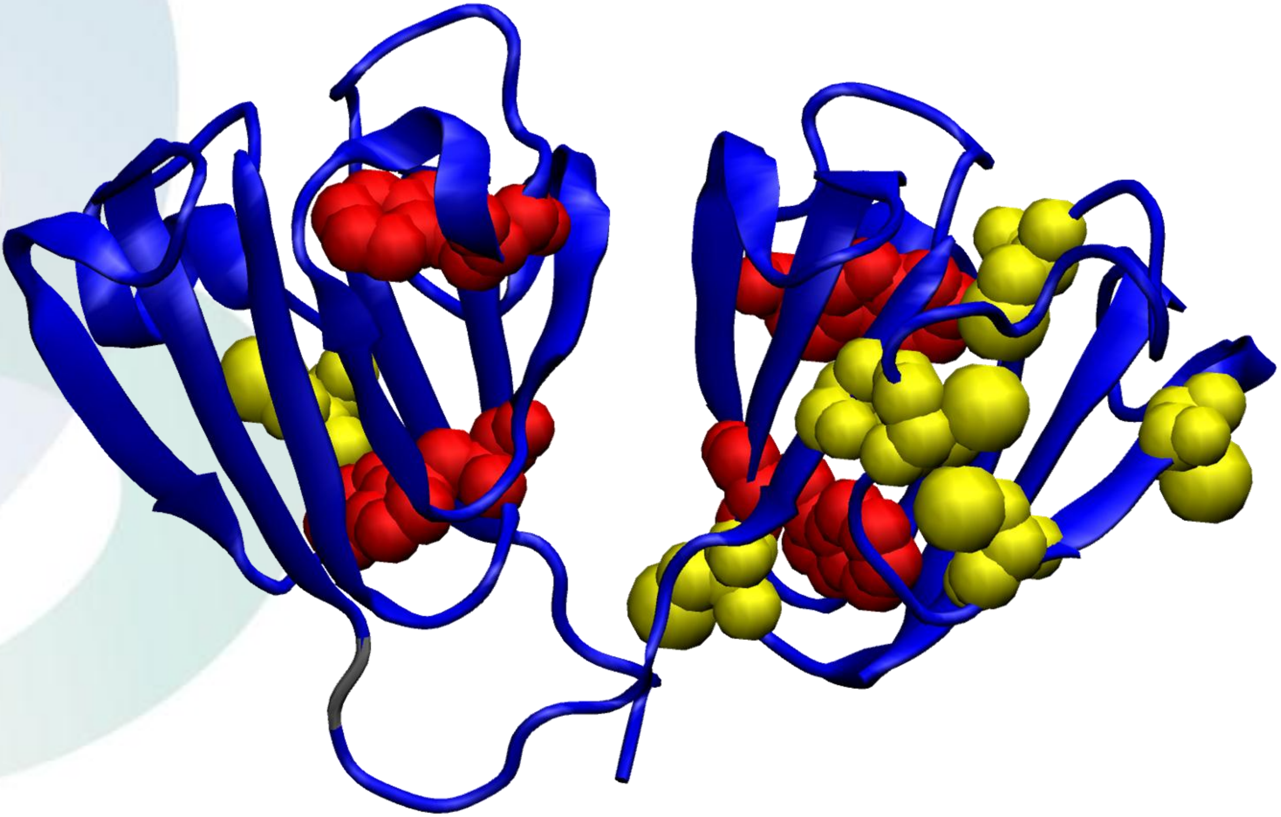
Using MD approach we can study enzymes activity and have an insight into detailed catalytic activity mechanism. NHase is industrial, light-active enzyme catalyzing nitrile (toxic) hydratation to amides (very useful).



Sample application:

UV alterations

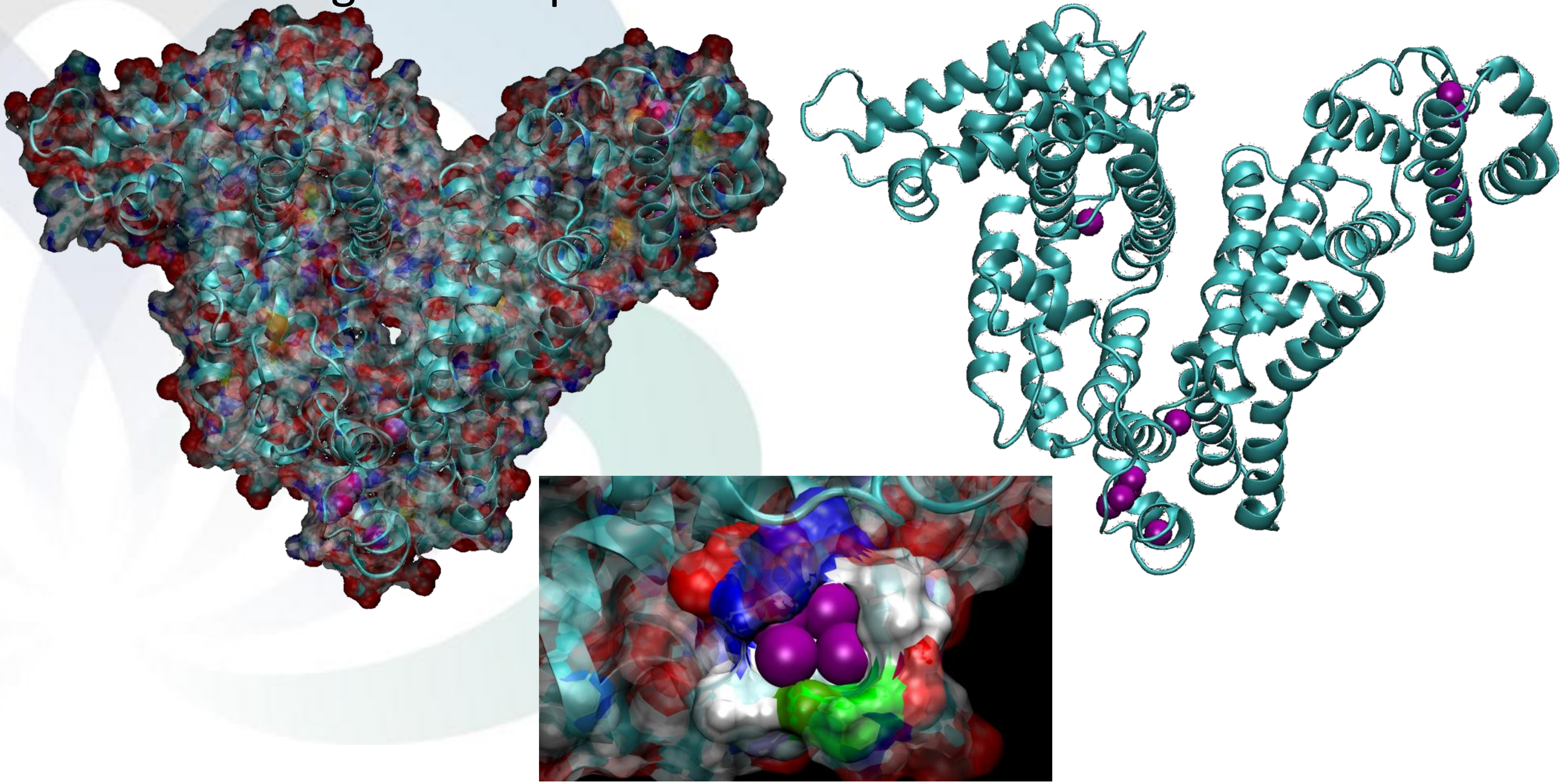
Gamma-crystalins from human eye lens are responsible for lens transparency. Upon UV irradiation the lens becomes opaque due to crystallins' aggregation. The main reason for that are tryptophan alterations (dyes creation) what influences on protein structure and results in aggregation.



Sample application:

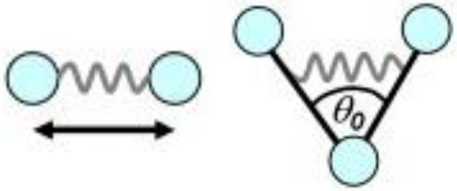
BSA gold binding

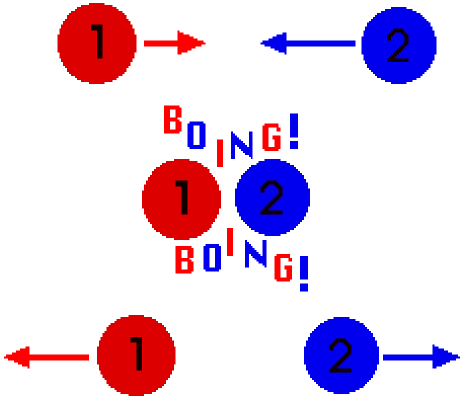
Bovine Serum Albumin (BSA) can bind gold ions and serve as a matrix for gold nanoparticles creation.



Approximations

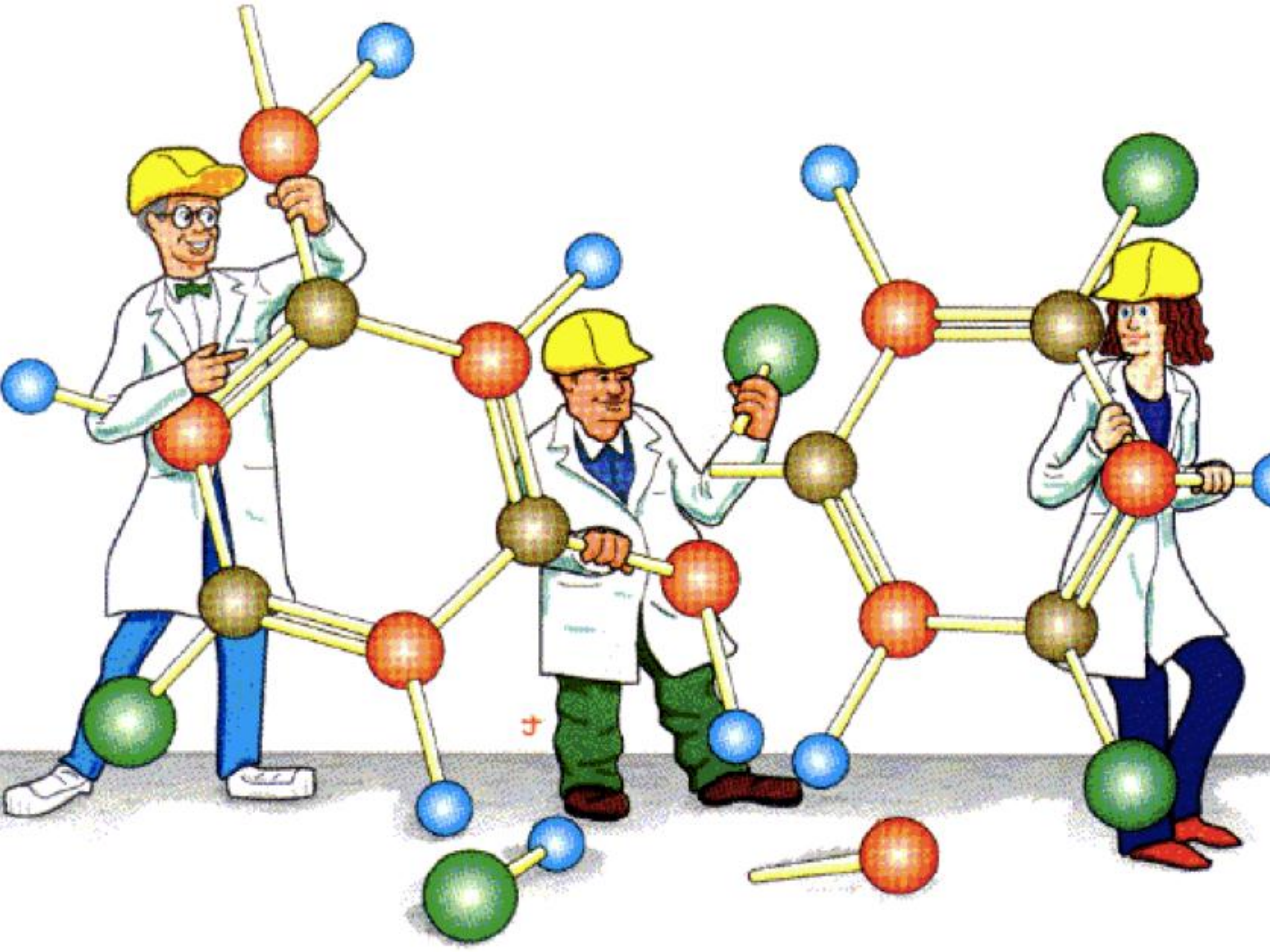
- Bonding interactions are modeled as springs (**harmonic oscillators**)
- All atoms are treated as a **hard spheres** (balls)
- All collisions are **perfectly elastic** (the total kinetic energy and momentum is conserved, no changes in the particle shape) as in noble gas atoms
- Atomic **charges are not changed** during the trajectory
- **Bonds cannot be created** (or destroyed) during the simulation





To run MD we have:

1. Build the **molecule** – find the (*x,y,z*) position of all atoms
1. Create (or apply existing) the **force field** (chose the software package)
1. Find the optimal structure (**energy minimization**)
1. **Solve the Newton's equations of motions** of all atoms in the force field at each time moment (timestep) -> the trajectory
1. **Analyze the trajectory** obtained, make or verify hypothesis and … write the paper ☺



Build the protein?

Building the protein is not very good idea, there is too many atoms with certain (*x,y,z*) positions. The 3D protein structure is crucial for its activity…

How to put the protein into the computer?

Protein Data Bank (pdb). Visit the page http://www.rcsb.org/pdb/home/home.do and download the structure!

Protein in the computer

Protein structures are solved by:

1. **X-ray diffraction**. Experimentalist has to make the protein crystal first (difficult), **H-hydrogen atoms are not visible** within this method (so we have to add them)
2. **NMR**. Works well only for small molecules, the entry includes several possible conformations – we have to chose one of them

Sometimes there are numerous entries for the same protein. **How to chose the best structure**? Check the deposition date, the resolution, experimental conditions, presence of any additives, if there are missing atoms or residues. It is easy to add the missing atoms, adding missing residues is more complicated.

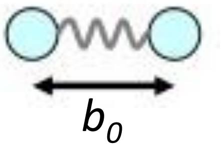
Force field

Protein atoms exist and move on the curved, multi-dimensional energy landscape which we want to reproduce.

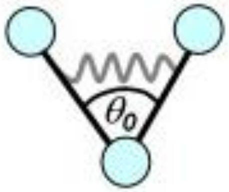
Force field - bonding interactions

This image cannot currently be displayed.

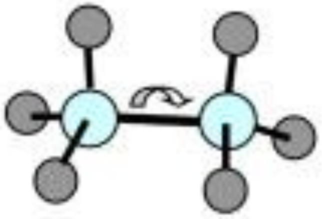
$$U(\vec{R}) = \frac{1}{2}\sum_{bonds} K_b (b - b_0)^2 +$$



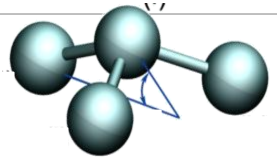
$$\frac{1}{2}\sum_{angles} K_\theta (\theta - \theta_0)^2 +$$



$$\sum_{dihedrals} K_\phi \left[1 + \cos(n\phi - \delta)\right] +$$



$$\sum_{impropers} K_\omega (\omega - \omega_0)^2 +$$

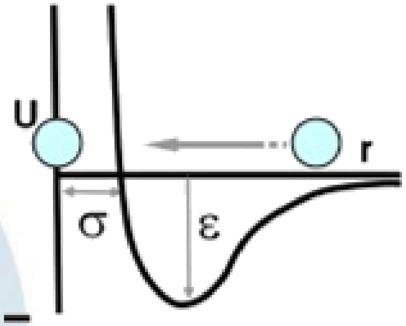


Simple harmonic oscillator approximation (neither driven nor damped)

$$F = -kx \quad U = \frac{1}{2}kx^2$$

Force field – non bonding interactions

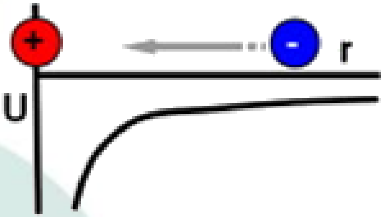
$$\sum_{i>j}\left(\frac{A_{ij}}{r_{ij}^{12}}-\frac{B_{ij}}{r_{ij}^{6}}\right)+$$



$$A_{ij}=2\sqrt{\sigma_i^{12}\sigma_j^{12}\varepsilon_i\varepsilon_j}$$

$$B_{ij}=2\sqrt{\sigma_i^{6}\sigma_j^{6}\varepsilon_i\varepsilon_j}$$

$$\sum_{i>j}\frac{q_iq_j}{4\pi\varepsilon_0 r_{ij}}+$$



$$\sum_{i>j}^{(4)}\left[v_{1,4}\left(\frac{A_{ij}}{r_{ij}^{12}}-\frac{B_{ij}}{r_{ij}^{6}}\right)+e_{14}\frac{q_iq_j}{r_{ij}}\right]$$

Morse potential

It explicitly includes the effect of bond breaking. Realistic but difficult to model.

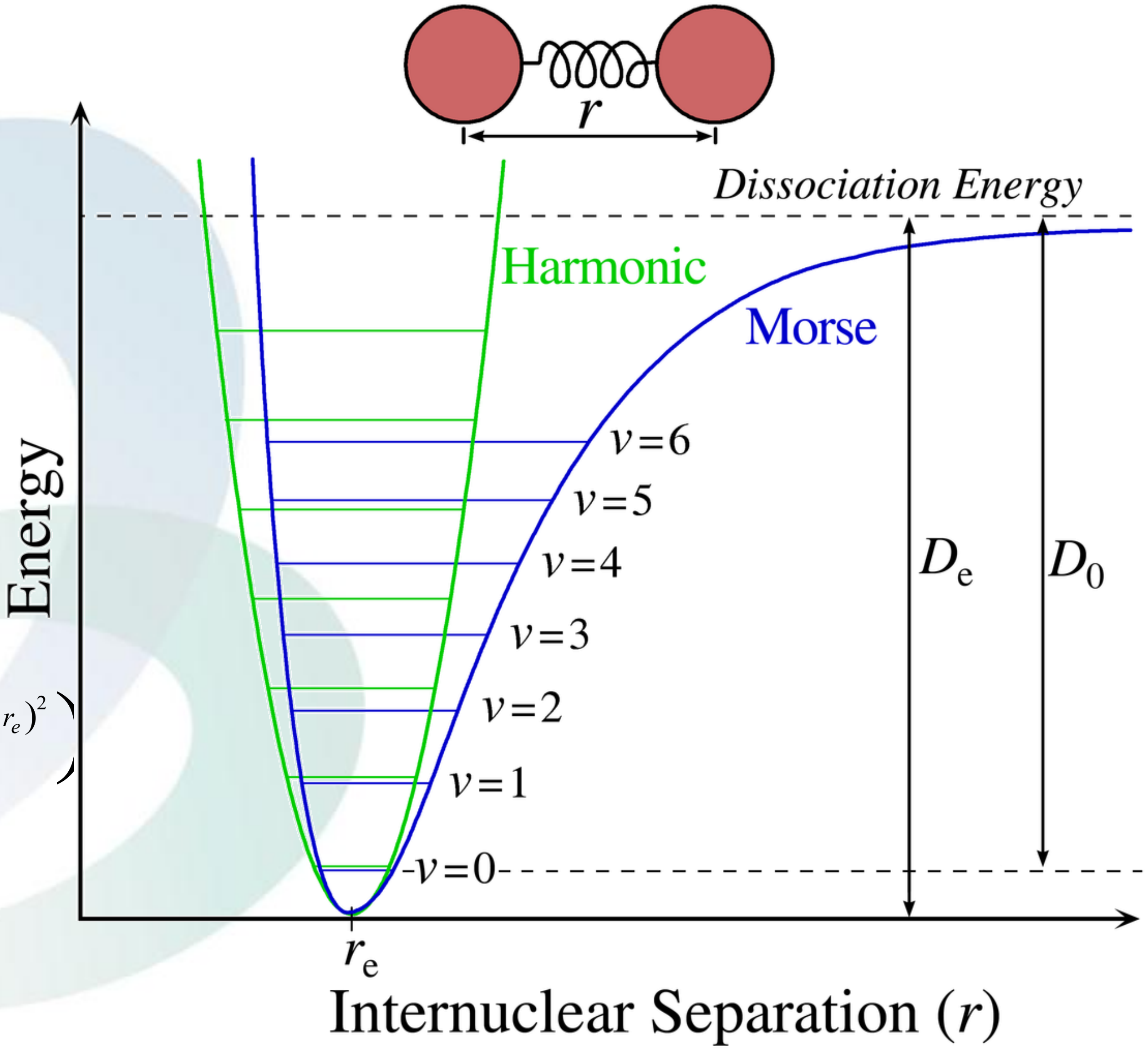
$$\alpha = \sqrt{\frac{k_e}{2D_e}}$$

$$U(r) = D_e\left(1 - e^{-\alpha(r-r_e)^2}\right)$$

De – the well depth

α - the well width

k_e - the force constant at the minimum of the well



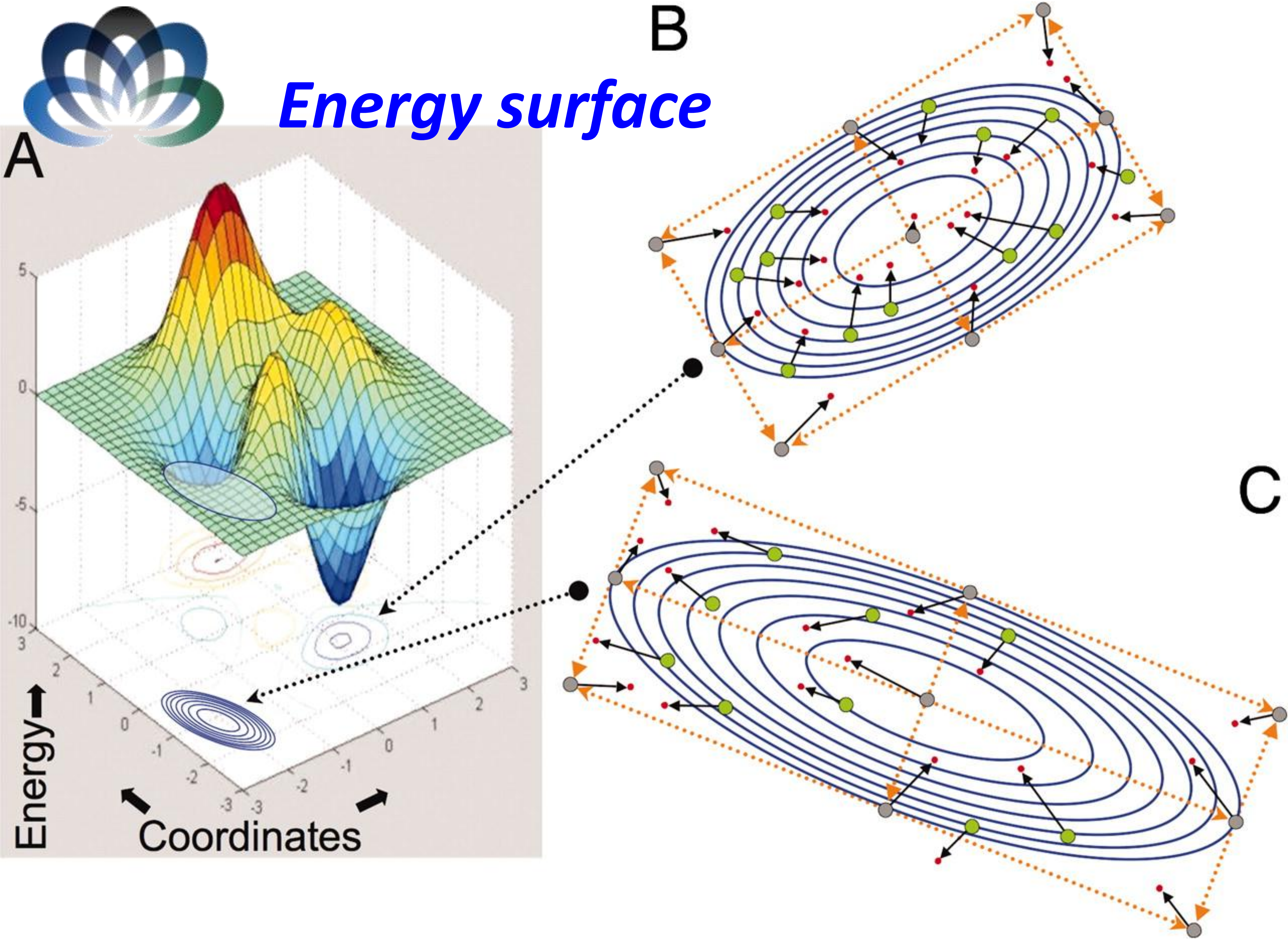
Energy minimization

Energy minimization methods are used to **compute the equilibrium configuration of molecules**. Stable states of molecules correspond to global and local minima on their potential energy surface. Starting from non-equilibrium geometry the mathematical procedure of optimization moves atoms to find the **lowest energy configuration**.

Energy minimization **does not include the temperature**. From physical point of view the final state of the system corresponds to the configuration of atoms in zero temperature.

Potential energy surface – multidimensional surface (hypersurface) with atomic positions as variables. During the minimization procedure we change atoms coordinates to find the global (ideally) minima.

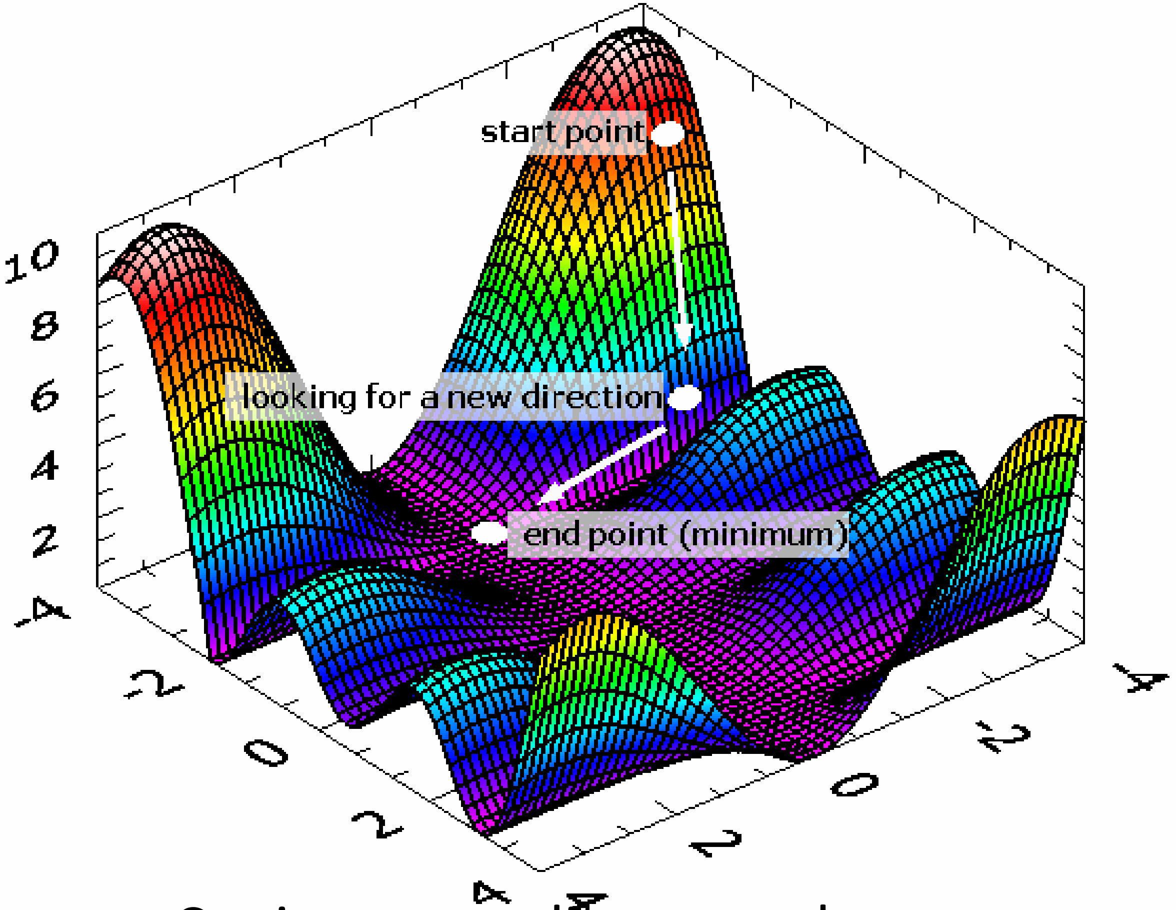
Energy surface



Energy minimization

Gradient-based algorithms (the most popular: **simple gradient, conjugate gradient.**

The **steepest descent** method represents other approach.



start point

looking for a new direction

end point (minimum)

Conjugate gradient search

Newton's equations

The force field defines the potential energy $U(\vec{R})$ where $\vec{R} = \{\vec{r}_1, \vec{r}_2, \vec{r}_3, \ldots, \vec{r}_N\}$ is a set of separate atom vectors $\vec{r}_1 = (x, y, z)$ and N is a number of atoms in the system.

We know that the force acting on atom i is:

$$\vec{F}(\vec{r}_i) = -\nabla U(\vec{r}_i)$$

and simultaneously from Newton's second law:

$$\vec{F}(\vec{r}_i) = m_i \vec{a}_i = m_i \frac{d^2 \vec{r}_i(t)}{dt}$$

and we want to find $\vec{r}_i(t)$ (solve the equation on motion) for atoms i=1, 2,...,N

The Verlet algorithm

The Verlet algorithm is a **numerical method used to integrate Newton's equations of motion**. The basic idea is to write two third-order Taylor expansions for the **atom i positions** $\vec{r}(t)$ one forward and one backward in time (*Δt* is a time step):

$$\vec{r}_i(t+\Delta t)=\vec{r}_i(t)+\frac{d\vec{r}_i(t)}{dt}\Delta t+\frac{1}{2}\frac{d^2\vec{r}_i(t)}{dt^2}\Delta t^2+\frac{1}{6}\frac{d^3\vec{r}_i(t)}{dt^3}\Delta t^3+O\left(\Delta t^4\right)$$

$$\vec{r}_i(t-\Delta t)=\vec{r}_i(t)-\frac{d\vec{r}_i(t)}{dt}\Delta t+\frac{1}{2}\frac{d^2\vec{r}(t)}{dt^2}\Delta t^2-\frac{1}{6}\frac{d^3\vec{r}_i(t)}{dt^3}\Delta t^3+O\left(\Delta t^4\right)$$

Adding the two expressions gives **the basic form of the Verlet algorithm:**

$$\vec{r}_i(t+\Delta t)=2\vec{r}_i(t)-\vec{r}_i(t-\Delta t)+\frac{d^2\vec{r}_i(t)}{dt^2}\Delta t^2+O\left(\Delta t^4\right)$$

The Verlet algorithm

$$\vec{r}_i(t+\Delta t) = 2\vec{r}_i(t) - \vec{r}_i(t-\Delta t) + \frac{d^2\vec{r}_i(t)}{dt^2}\Delta t^2 + O\left(\Delta t^4\right)$$

The truncation error is of order of Δt^4 ☺

The acceleration $\frac{d^2\vec{r}_i(t)}{dt^2}$ can be obtained from:

$\vec{F}(\vec{r}_i) = -\nabla U(\vec{r}_i)$ and $\vec{F}(\vec{r}_i) = m_i\vec{a}_i = m_i\frac{d^2\vec{r}_i(t)}{dt}$ as:

$$\vec{a}_i(t) = \frac{d^2\vec{r}_i(t)}{dt^2} = -\left(\frac{1}{m_i}\right)\nabla U(\vec{r}_i)$$

In this algorithm the **velocities are not directly generated**. While they are not needed for evolution, their knowledge is **necessary to compute the kinetic energy K**. The total energy E=K+U should be conserved during the trajectory.

The Verlet algorithm

The velocity can be calculated from

$$\vec{v}_i(t) = \frac{d\vec{r}_i(t)}{dt} = \frac{\vec{r}_i(t+\Delta t) - \vec{r}_i(t-\Delta t)}{2\Delta t}$$

and used to calculate the kinetic energy in the time moment t:

$$K(t) = \sum_{i=1}^{N} \frac{m\vec{v}_i(t)^2}{2}$$

Nevertheless the truncation error for velocity calculation is of order Δt^2 rather than Δt^4 ☹

We would like to calculate it more precisely ➔ **The velocity Verlet algorithm**

The Velocity Verlet algorithm

The position of the atom is updated every Δt step

$$\vec{r}_i(t+\Delta t) = \vec{r}_i(t) + \vec{v}_i(t)\Delta t + \frac{1}{2}\vec{a}_i(t)\Delta t^2$$

while the velocity is updated every $\left(\Delta t + \Delta t/2\right)$ step:

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

The acceleration in the next step $(t+\Delta t)$:

$$\vec{a}_i(t+\Delta t) = -\left(\frac{1}{m_i}\right)\nabla U\left(\vec{r}_i(t+\Delta t)\right)$$

The velocity in the next step $(t+\Delta t)$:

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

☺

The Velocity Verlet algorithm

- The same order as Basic Verlet algorithm
- Fast and accurate
- Linear and angular momentum preserved
- Time reversible

The Velocity Verlet algorithm generates a sequence of “snapshots” for the particle coordinates and velocities at all intermediate times *Δt*

☺

Calculation procedure

1. Chose the molecule from the Protein DataBank
2. Chose the force field
3. Minimize the potential energy (find the optimal conformation) -> calculate $U\left(\vec{R}(t=0)\right)$ which will be used to calculate initial accelerations in the trajectory calculations
4. Calculate the trajectory in given temperature

Trajectory calculations

For atom i=1:

1. Calculate the initial acceleration $\vec{a}_1(t)$ from the force field

$$\vec{a}_1(t) = -\left(\frac{1}{m_1}\right)\nabla U\left(\vec{r}_1(t)\right)$$

2. Guess the initial velocity $\vec{v}_1(t)$ (Gaussian distribution with a temperature dependent variance
3. Calculate the position at next time interval $(t+\Delta t)$

$$\vec{r}_1(t+\Delta t) = \vec{r}_1(t) + \vec{v}_1(t)\Delta t + \frac{1}{2}\vec{a}_1(t)\Delta t^2$$

1. Repeat 1-3 for atoms i=2,…,N
2. Calculate the force field at the next time interval $U\left(\vec{R}(t+\Delta t\right)$

$\vec{R} = \{\vec{r}_1, \vec{r}_2, \vec{r}_3, \ldots, \vec{r}_N\}$ using new positions of all atoms i=1,…,N.

Trajectory calculations cd

For atom i=1:

6. Calculate the acceleration $\vec{a}_1(t+\Delta t)$at the next time interval

$$\vec{a}_1(t+\Delta t) = -\left(\frac{1}{m_1}\right)\nabla U\left(\vec{r}_1(t+\Delta t)\right)$$

6. Calculate the velocity at the next half- time interval

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

6. Calculate the velocity $\vec{v}_1(t+\Delta t)$ at the next time interval

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

7. Repeat for atoms i=2,…,N
8. Repeat 3-9
9. At given intervals rescale velocities to given temperature

Ensembles

Microcanonical ensemble (NVE) – number of atoms (N), the system volume (V) and the total energy (E) are conserved. **Canonical ensemble (NVT)** - number of atoms (N), the system volume (V) and the temperature (T) are conserved. Energy is exchanged with a thermostat.
Isothermal-isobaric ensemble (NPT) - number of atoms (N), the system pressure (P) and the temperature (T) are conserved. Thermostat and a barostat are needed.

Basing on positions and velocities we have to calculate statistical quantities.

Simulation temperature

Temperature is a **statistical quantity** which has to be expressed as a function of positions and momenta of all particles in the system. For a system containing large enough number of atoms the **temperature can be estimated from the kinetic energy**:

$$\left\langle \frac{1}{2} m v^2 \right\rangle = \frac{1}{2} N_f K_B T$$

where N_f is number of degreed of freedom in the system with N particles (atoms) and K_B is Boltzman constant =1.38x10^{-23} $\frac{m^2 kg}{s^2 K}$

Temperature depends on time because positions and velocities depend on time

$$T(t) = \sum_{i=1}^{N} \frac{m_i v_i^2(t)}{K_B N_f}$$

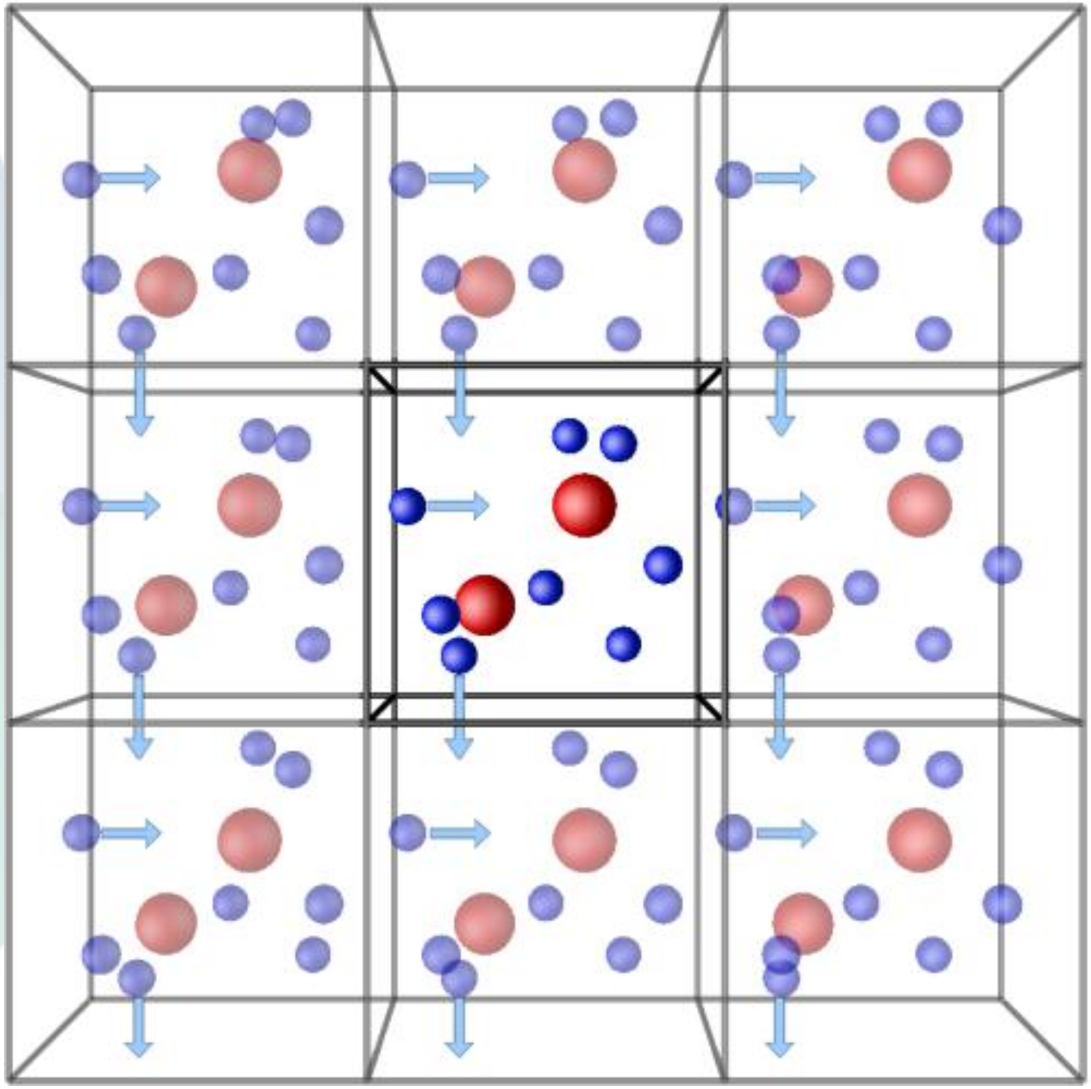
Thermostats

Thermostats are algorithms to re-scale the velocities of particles to control the simulation temperature.

Berendsen thermostat – the most straight forward method, not very accurate
Langevin dynamics – more advanced and more accurate method. Now commonly used in most MD packages.

Periodic boundary conditions

The simulation box must be large enough (much larger than a molecule). Commonly used in MD, various cell geometries are available.



Constraint MD

Constraint algorithm is a method to constrain bodies in Newton’s equations of motion. Constraint algorithms are often used in MD to omit some part in trajectory calculations.

SHAKE algorithm satisfies bond geometry constraint in MD, is limited to mechanical systems with a tree structure (no closed loops of constraints). Other versions: **Q-SHAKE, MSHAKE**, and **RATTLE** which works well with Velocity Verlet algorithm.

Locally Enhanced Sampling

Locally **E**nhanced **S**ampling (**LES**) increases sampling and transition rates for a portion of a molecule by use of **multiple non-interacting copies of the enhances atoms**. These enhanced atoms experience an interaction (electrostatics, vdw, bonding) potential that is divided by the number of copies present. In this way the enhanced atoms can occupy the same space, while the multiple instances and reduces barriers increase transition rates.

Copies can't see each other, they see the rest of the system. The rest of the system see the average potential and mass of enhanced part

Useful in studying small ligand diffusion in proteins.

Steered Molecular Dynamics

The basic idea behind any SMD simulation is to **apply an external force to one or more atoms**, which we refer to as SMD atoms. In addition, you can keep another group of atoms fixed and study the behaviour of the protein under various conditions.

SMD allows to explore biological processes on time scale accessible to MD simulations. There are two types of SMD: **constant force pulling and constant velocity pulling**. In constant force pulling a chosen atom is kept fixed and another one experiences a constant force in the direction defined by the vector that links the fixed and pulled atom.

Steered Molecular Dynamics

In constant velocity pulling **the SMD atom (pulled atom) is attached to a dummy atom via a virtual spring**. This dummy atom is moved with a constant velocity and then the force between them is measured:

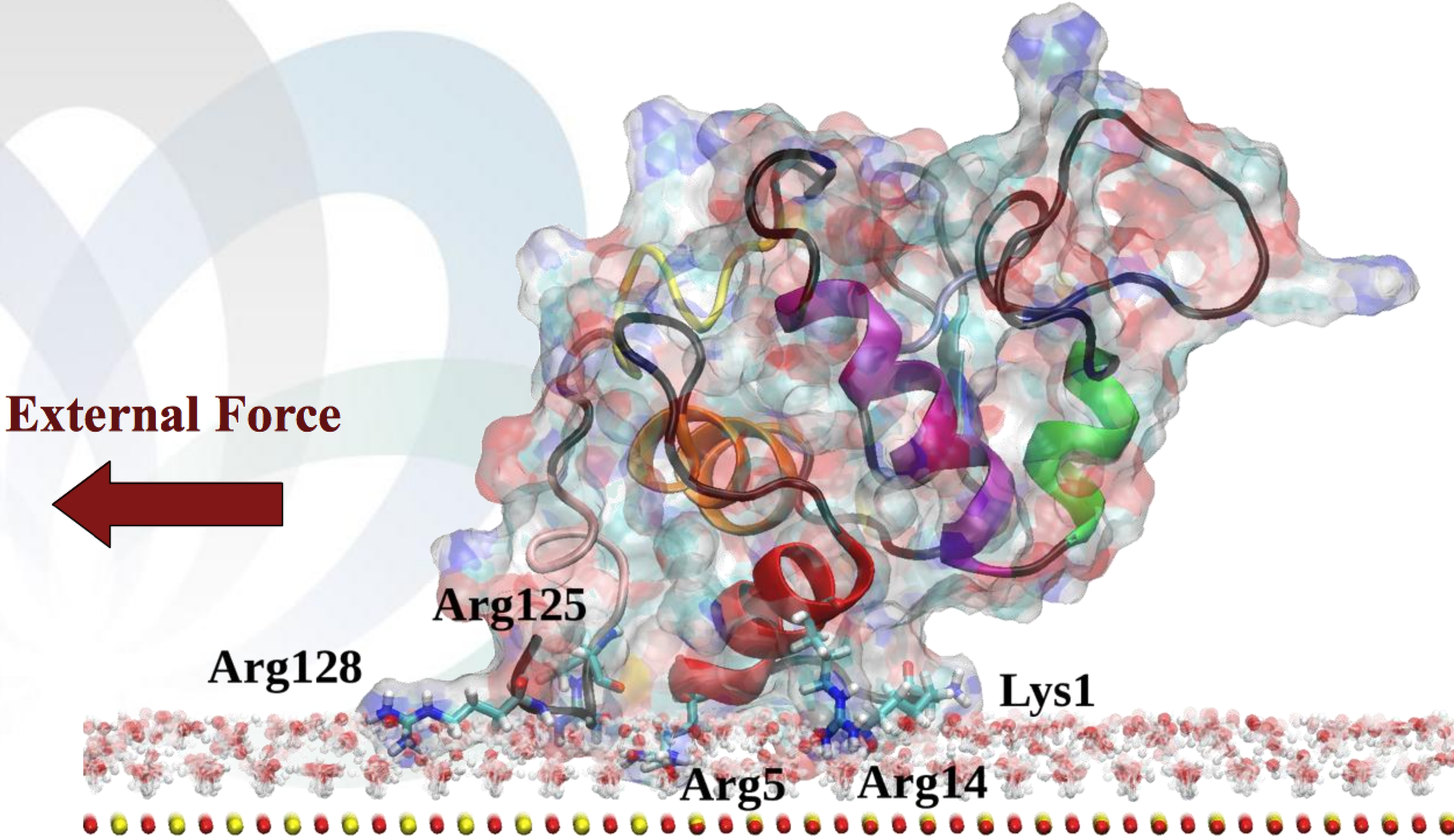
$$\vec{F}(\vec{r}) = -\nabla U(\vec{r})$$

$$U(\vec{r}) = \frac{1}{2}k\left[vt - (\vec{r} - \vec{r}_0)\vec{n}\right]^2$$

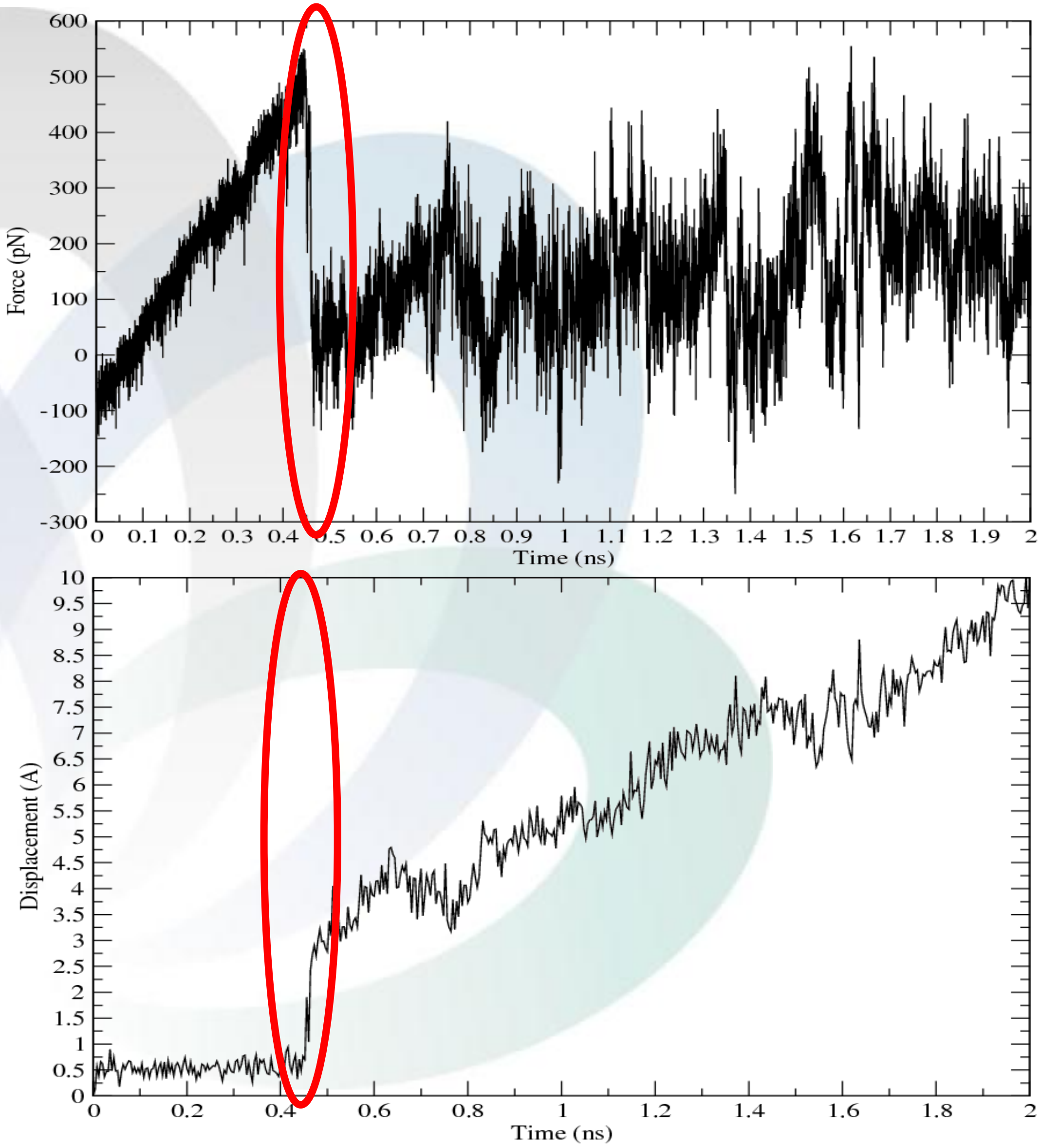
Where:

k	- spring constant
v	- pulling velocity
$\vec{r}$	-actual position of SMD atom
$\vec{r}_0$	- initial position of SMD atom
$\vec{n}$	- direction of pulling

Steered Molecular Dynamics



Steered Molecular Dynamics



$$dE = \left(F_0 + \frac{dF}{2} \right) \frac{dF}{k}$$

Where:

F_0 - the force at the end of transition,

k – spring constant

dF – the force change

Here:

F_0 ~ 0 N

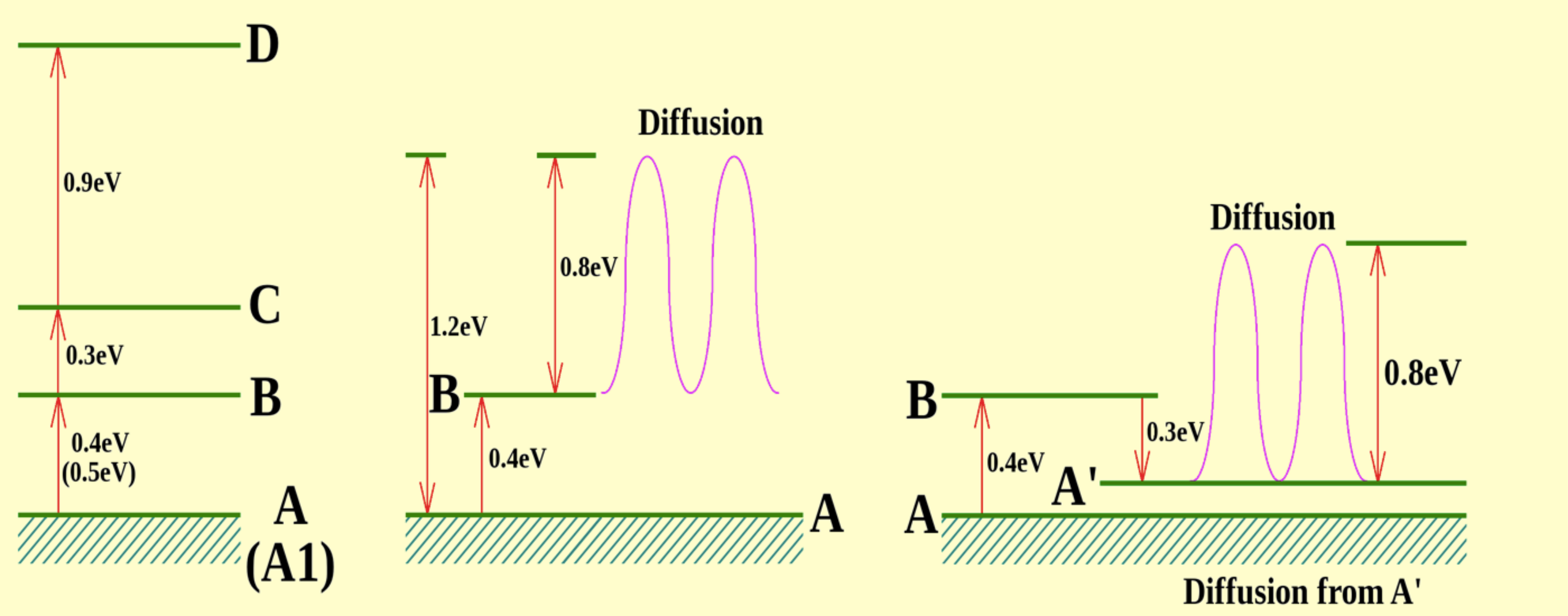
dF ~ 575 pN

k = 278 Å/pN

dE=543pN*Å (10^{-22J}J)

dE=0.34 eV

Steered Molecular Dynamics



Particle Mesh Ewald

Ewald summation is a method for computing the interaction energies of periodic systems, particularly electrostatic energies.

Particle Mesh Ewald (PME), since 1970s
As in normal Ewald summation, the interaction is separated into two terms: short and long-range. PME replaces the direct summation of interaction energy between point particles with **two summations: a direct sum of the short-ranged potential in real space and a summation in Fourier space of the long-ranged part.** Both summations converge quickly, so they may be truncated with little loss of accuracy and great improvement in required computational time. The method uses FFT which requires that the density field be evaluated on a discrete lattice in space (mesh).

Particle Mesh Ewald

PME method can be applied to systems with **periodic symmetry**. In MD a charge-neutral unit cell is created which is infinitely “tiled” to form images. It means that PBC are used. The unit cell must be big enough to avoid improper motion correlations between two faces of the cell and still small enough to be computationally feasible.

PME is an efficient **full electrostatic method** for use with PBC. It should not affect energy conservation, although it may affect the accuracy of the result and momentum conservation. **PME is more accurate and less expensive than larger cutoffs.** The particle mesh is a 3D grid created in the system over which the system charge is distributed. From this charge, potentials and forces on atoms in the system are determined.

Cost reduction

Cut offs

The most time-consuming part in MD calculations are calculations of non-bonding interactions: van der Waals and electrostatics interactions. They should be calculated to the infinitum, what is impossible, usually we cut this calculations in 12A distance.

Constrained MD

Usually we are not interested in calculations of the fastest bond stretching (like O-H bonds). Therefore the constrained MD is used. SHAKE and RATTLE (velocity version of SHAKE) algorithms allow to constrain the length of all bonds with hydrogen atoms involved. Water molecules are kept rigid.

Thank you for your attention!

Dr Karina Kubiak - Ossowska
High Performance Computing Support Team
Department of Physics

e-mail: karina.kubiak@strath.ac.uk