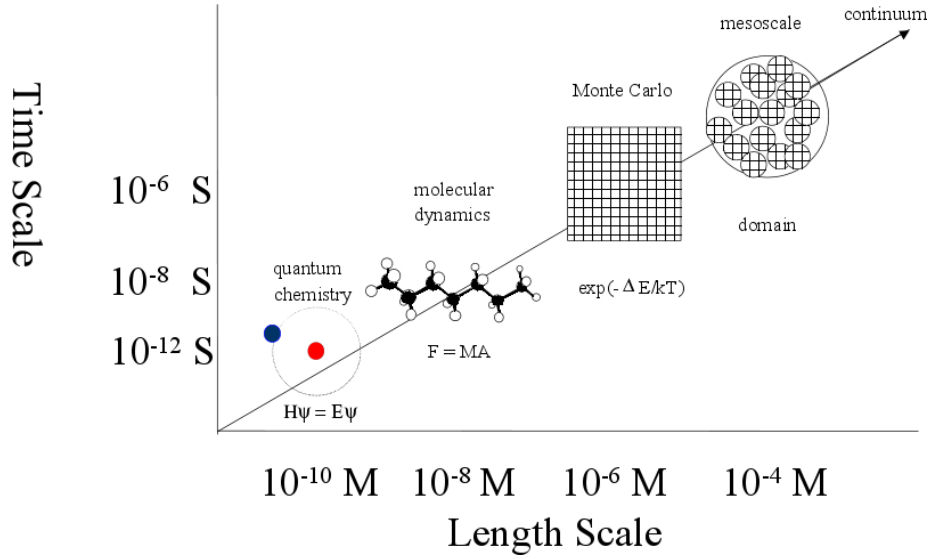


Figure 1: Time scales for MD simulations

Biomolecular Simulations

Biomolecular simulations have become instrumental in replacing our view of proteins as relatively rigid structures with the realization that they were dynamic systems, whose internal motions play a functional role. Over the years, such simulations have become a central part of biophysics. Applications of molecular dynamics in biophysics range over many areas. They are used in the structure determination of macromolecules with x-ray and NMR data, the modeling of unknown structures from their sequence, the study of enzyme mechanisms, the estimation of ligand-binding free energies, the evaluation of the role of conformational change in protein function, and drug design for targets of known structures.

1 Time Scales

Figure 1 shows the time scales for various atomic and molecular processes: Atomic vibrations take place over femtoseconds (10^{-15} of a second) and less than 1Å ($10^{-10}M$) length scale. to picoseconds (10^{-12} of a second). Protein domain interactions and small molecule dynamics can take place over nanoseconds (10^{-12} of a second) or microseconds (10^{-12} of a second) or more. Protein folding may take place over milliseconds (10^{-3} of a second) or even seconds. Large scale cellular processes may take even more time. Since computational resources are limited, different processes require different simulation methods. Often we have to sacrifice accuracy for tractability.

2 Quantum Mechanics Calculations in a Nutshell

Quantum mechanics (QM) is often used for electronic structure calculations. The tern *ab initio* is often used, denoting using first principles without empirical parameters. To model electronic rearrangements during a chemical reaction, a QM model is required for the parts of the system that are involved in the reaction. Molecular mechanics simulations are not able to model forming and breaking of bonds, for example.

Electronic structure calculations involve solving the time dependent Schrödinger's equation [**?**]:

$$i\hbar\frac{\partial}{\partial t}\Psi(x,t) = \hat{H}\Psi(x,t) \tag{1}$$

where $\hbar = h/2\pi$ is the Planck constant, Ψ is the wave function of the system, $\hat{H}$ is the Hamiltonian operator, E is the energy eigenvalue of system, i is the imaginary unit and t is time. The Hamiltonian operator contains the kinetic and the potential energies of the studied system. The wave function defines the probability density, i.e., $ro(x,t)$ – the probability of the system to be in the configuration defined by x at time t. The partial differential equation needs to be closed, with a defined initial value $\Psi(x,0) = \Psi_0$, and boundary conditions. Solving this equation describes the dynamic behavior of a system, but it is prohibitively expensive for anything but very small molecular systems.

3 Classical Mechanics Molecular Dynamics

Molecular Dynamics (MD) is a method that simulates the dynamics of molecules under physiological conditions over a period of time. It uses physics to find the potential energy between and forces acting on all the pairs of atoms in a molecule. At the basis of classical mechanics MD simulation is the application of a step-by-step numerical integration to Newton's equation of motion. While classical mechanics cannot model electronic structure calculations which involve quantum effects, it is much faster and can be used for larger molecular systems.

4 Classical Molecular Mechanics and Newton's Laws

Let us start with a quick reminder of Newton's second law. Imagine Looking at an atom at a certain point in time, t. The atom is located at position $\vec{r1} = (r1_x, r1_y, r1_z)$ in space, and is moving at a certain velocity $\vec{v1} = (v1_x, v1_y, v1_z)$. Notice that the velocity is a vector and therefore has not only a magnitude but also a direction in space. Imagine now that a force $\vec{F} = (F_x, F_y, F_z)$ is applied to the atom. The force is also a vector, and hence has a direction. Let us denote the time that passed since the force has been applied by dt, an infinitesimally step, just enough for the force to impact. As a result, the atom changes its location to $\vec{r2} = (r2_x, r2_y, r2_z)$ and its velocity to $\vec{v2} = (v2_x, v2_y, v2_z)$ at time $t + dt$. The process is illustrated in Figure 2.

Newton's second law provides the relationship between force and acceleration:

$$\vec{F} = m\vec{a} \tag{2}$$

where m is the atomic mass and $\vec{a}$ is the acceleration. Remember that the acceleration is the second derivative of the position with respect to time, and the velocity is the first derivative of the

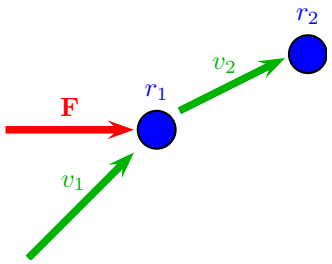


Figure 2: The change in position and velocity of an atom (in blue) after a force is applied to it.

position with respect to time. Hence, this is a second-order differential equation. With this, if we know the mass of the atom, we can derive the relationship between its position and velocity at time t and its position and velocity at time $t + \Delta t$. All we need is to integrate Newton's second law with respect ot time:

$$\vec{F} = m\vec{a} = m\frac{d\vec{v}}{dt} = m\frac{d^2\vec{r}}{dt^2} \tag{3}$$

When integrating in a computer, we approximate dt by a very small discrete time step Δt to get:

$$\Delta\vec{v} = \frac{\vec{F}}{m}\Delta t \rightarrow \vec{v_2} = \vec{v_1} + \frac{\vec{F}}{m}\Delta t\vec{r_2} = \vec{r_1} + \vec{v_2}dt = \vec{r_1} + \vec{v_1}dt + \frac{\vec{F}}{m}dt^2 \tag{4}$$

In other words, the new position, r_2 is determined by the old position, r_1 and the velocity v_2 over time Δt (which should be very small!). The above equation describes the changes in the positions of the atoms over time. An MD simulation, in its most basic form, is a repeated numerical integration of the Newton equations over time, where the positions and velocities at time t give us the positions and velocities at time $t + \Delta t$

The set of positions + velocities over a period of time, also called the *trajectory*, give us a description of how the molecule behaves over this period of time. However, we need more information in order to conduct the simulation. First of all, we need the starting conditions. As this is a second-order differential equation, we need two sets of initial conditions – the initial positions and velocities of all the atoms as input. To carry out the integration we all have to know the masses of all the atoms. This information is given as part of the initial parameters. Now we are ready to go... but wait, what about the force?

We use the relationship between force and potential energy. The force is the negative gradient (directional derivative) of the potential energy. The potential energy, denoted U, is a scalar (a directionless number). Differentiating it with respect to the position vector gives us the force.

$$\vec{F} = -dU/d\vec{r} \rightarrow U = -\int \vec{F}dr = -1/2 * m\vec{v}^2 \tag{5}$$

The Energy is conserved, hence $\frac{1}{2} * \sum_{i=1}^{n} M_i v_i^2 + \sum E_{pot,i} = const$

Note, that mixing equations and parameters from different systems always results in errors!

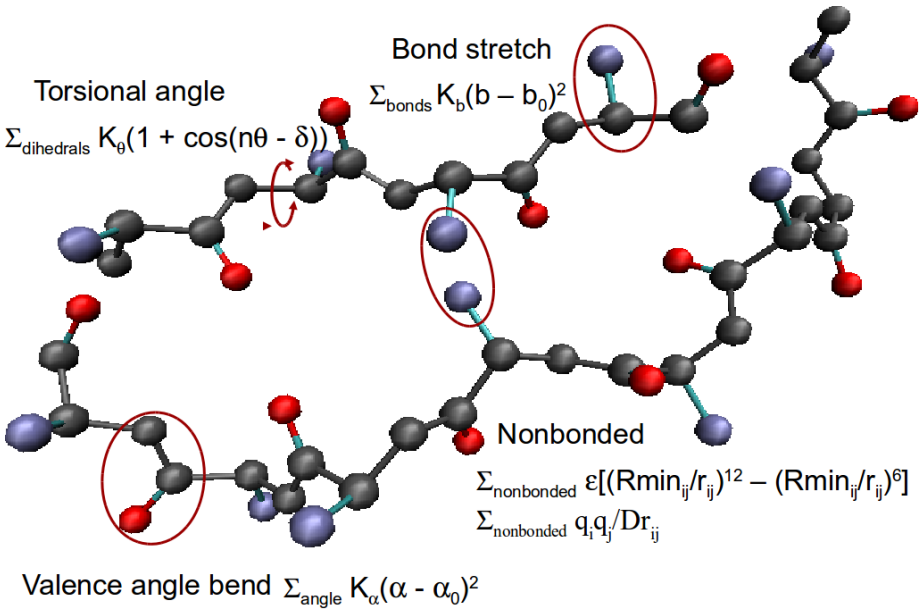


Figure 3: An illustration of a molecular potential energy (force field) equation.

5 Force Field Equations

All the **equations** and the **adjusted parameters** that allow to describe quantitatively the energy of the chemical system are hence called the *force field* parameters.

Here is an example of a typical potential energy equation. It is a sum of all the interactions among atoms. The meaning of the equation is illustrated in Figure 3.

$$
\begin{aligned}
U = & \\
& \sum_{bonds} K_b(b - b_0)^2 + && \text{Bonds} \\
& \sum_{angles} K_\alpha(\alpha - \alpha_0)^2 + && \text{Angles} \\
& \sum_{torsion} \frac{V_n}{2}(1 + \cos[n\theta - \delta]) + && \text{Dihedrals} \\
& \sum_{i,j} \frac{q_i q_j}{\epsilon r_{ij}} + && \text{Electrostatic} \\
& \sum_{i,j} \varepsilon[(\frac{Rmin_{ij}}{r_{ij}})^{12} - (\frac{Rmin_{ij}}{r_{ij}})^6] && \text{Van der Waals (VdW)}
\end{aligned}
$$

The covalent bonds, planar angles and dihedral angles are called bonded terms, illustrated

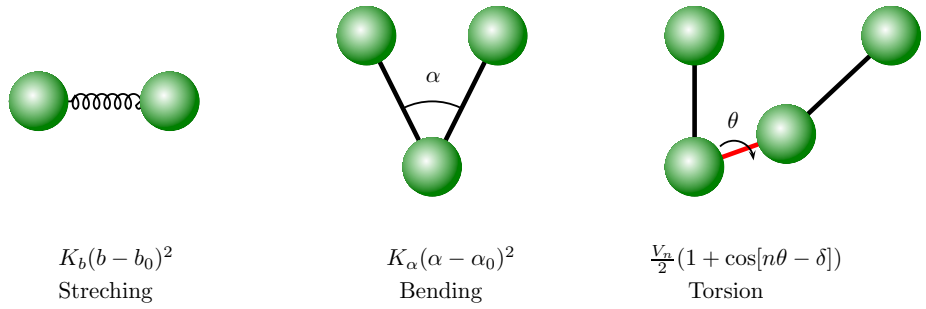


Figure 4: An illustration of a bond and angle energy, modeled as a spring potential function. The dihedral (torsion) term is modeled as a trigonometric function.

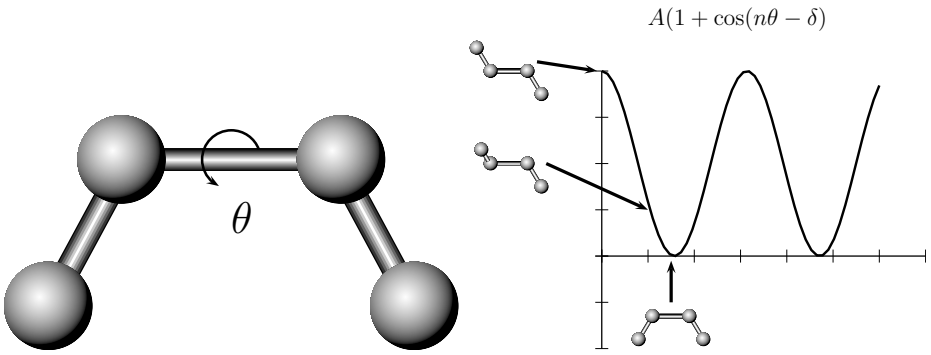


Figure 5: (a) The torsion angle explained as the angle around a bond. Several values are shown in (b)

in Figure 4. bonds are defined as covalent bonds between atoms. The term above is a simple harmonic potential with a spring constant K_b. b is the measured distance between the two atoms and b_0 denotes the "ideal" bond distance. The Bond potential is summed over all the covalent bonds in the molecule. The planar angles are defined over three consecutive atoms. Similarly, it is a harmonic potential with α being the measured angle, α_0 being the ideal angle and K_α is the spring constant. As with bonds, the term is a summation over all triplets of atoms that form an angle. Notice that this is a spring potential function, so these two terms are non-negative. They become zero when the bond or angle are equal to their ideal value. Otherwise the term is positive – which is a penalty for deviation from the ideal (resting) bond length or angle. The dihedral (torsion) term is slightly more complicated. It is a phase trigonometric function defined over the dihedral angle between four atoms (see Figure 5). A controls the amplitude of the curve, n controls its periodicity, and δ shifts the entire curve along the rotation angle axis (θ). The torsion energy parameters are determined from curve fitting. Unique parameters for torsional rotation are assigned to each bonded quartet of atoms based on their types (e.g. C-C-C-C, C-O-C-N, H-C-C-H, etc.). All these constants and parameters are given in the force field input file.

Figure 6 shows the meaning of the three parameters and their effect on the trigonometric

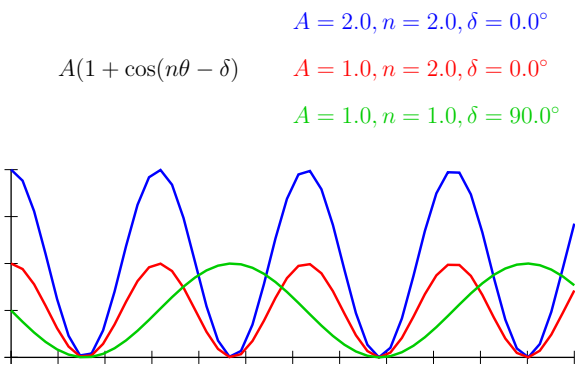


Figure 6: The effect of the different parameters on the torsion term.

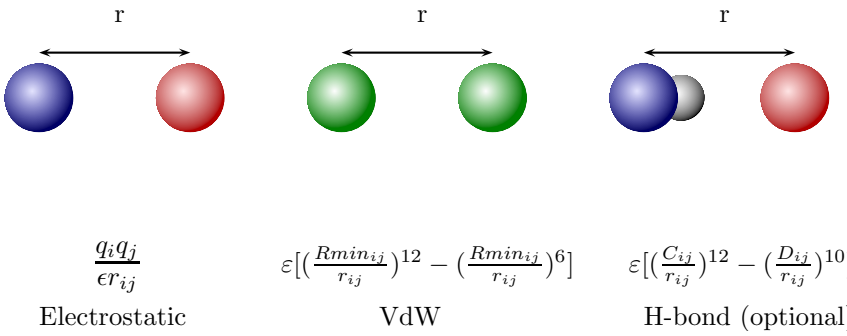


Figure 7: An illustration of the electrostatic, VdW and hydrogen bond energy.

function: A is the amplitude, n reflects the type symmetry in the dihedral angle, δ is used to synchronize the torsional potential to the initial rotameric state of the molecule (phase shift).

Non-Bonded Energy Parameters The non-bonded terms are applied to all the pairs of atoms that are at least four atoms apart on the chain.

Most all-atom force-fields contain at least an electrostatic term and a van-der-Waals (VdW) term. The electrostatic term measures the attraction or repulsion from atomic partial charges, according to Coulomb's law. For each two atoms i and j with partial charges q_i and q_j, respectively, the electrostatic term is

$$\frac{q_i q_j}{\epsilon r_{ij}} \tag{6}$$

Where r_{ij} is the distance between atoms i and j. Notice that two atoms with opposite sign charges contribute a negative (favorable) term and two atoms whose charges have the same sign contribute a positive (unfavorable) term. The electrostatic potential decays linearly with the distance between the atoms. ϵ is the dielectric constant, also known as the relative permittivity of the environment surrounding the two atoms relative to vacuum. The permittivity of vacuum is therefore 1, by definition. The term dielectric denotes an electrical insulator that can be polarized

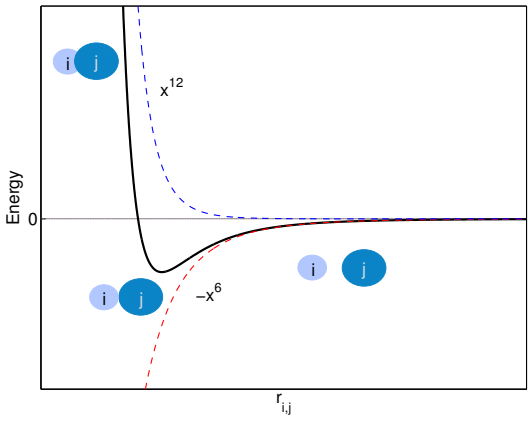


Figure 8: The VdW energy illustrated by a Lennard-Jones potential.

by an electric field. Intuitively, the more polarizing the environment, the more it tends to "mask out" electric charges, and therefore its relative permittivity is higher. The permittivity changes with the temperature and pressure. At room temperature the permittivity of water is 80.4. This means that the electrostatic potential of two atoms is 80.4 times lower in water than in vacuum. Most hydrophobic solvents have relative permittivity of 2-4. The VdW term is

Solvation Models We are not done yet! Remember that our goal is to simulate a molecule under physiological conditions which resemble its real live behavior. Unless the molecule is in vacuum, we also have to model the solvent it is immersed in. There are several possible solvation models:

- No solvent – constant dielectric.
- Continuum – referring to the solvent as a bulk. No explicit representation of atoms (saving time).
- Explicit – representing each solvent molecule explicitly (this is more accurate, but expensive).
- Mixed – mixing two models (for example: explicit + continuum. To save time).

Periodic Boundary Conditions Only a small number of molecules can be simulated and the molecules at the surface experience different forces than those at the inner side, due to the fact that they are on the boundary and therefore not all of their sides are "covered" by molecular forces.

Periodic boundary conditions enable a simulation to be performed using a relatively small number of particles in such a way that the particles experience forces as though they were in a bulk solution. See, for example, the two dimensional box in Figure 9. The central box is surrounded by eight neighbors. In three dimensions, the simulation box is replicated infinitely along these dimensions (to integrate the boundaries of the box). When the molecule moves, the images move in the same fashion. The assumption is that the behavior of the infinitely replicated box is the same as a macroscopic system.

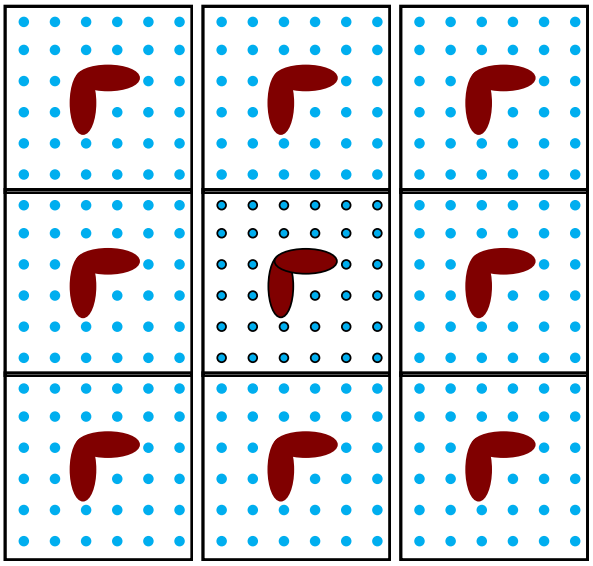


Figure 9: An illustration of periodic boundary conditions with infinite dilution.

The simulation box (unit cell) is defined by the 3 box vectors a,b and c. The simplest box type is cubic, where $a = b = c$ and the angle between the vectors is 90 degrees. Other common box types exist, the most widely used is orthorhombic, with $a \neq b \neq c$ and the angle between the vectors is 90 degrees.

The coordinates of the image particles, those found in the surrounding box are related to those in the primary box by simple translations. Forces on the primary particles are calculated from particles within the same box as well as in the image box. The cutoff is chosen such that a particle in the primary box does not see its image in the surrounding boxes: The minimum image convention implies that the cut-off radius used to truncate non-bonded interactions may not exceed half the shortest box vector, because otherwise more than one image would be within the cut-off distance of the force. When a macromolecule, such as a protein, is studied in solution, this restriction alone is not sufficient: in principle, a single solvent molecule should not be able to ?see? both sides of the macromolecule. This means that the length of each box vector must exceed the length of the macromolecule in the direction of that edge plus two times the cut-off radius R_c.

5.1 An Example of a Protocol

Here is a typical equilibrium state MD protocol:

1. Read the force fields data and parameters.

2. Read the coordinates and the solvent molecules.

3. Perform energy minimization of the coordinates (the created model may contain collisions).

4. Warm the system to the desired temperature (could be room temperature or any other temperature).

5. Equilibrate the system.

6. Start the dynamics (production runs) and save the trajectories every 1ps (trajectory=the collection of structures at any given time step).

Now let us go over each step separately.

5.1.1 Why is Minimization Required?

Most of the coordinates in the PDB are obtained using X-ray diffraction or NMR. X-ray diffraction does not map the hydrogen atoms of the system, since they are too small to be detected by the X-ray. Those hydrogen atoms are added later using modeling programs, which are not 100% accurate. The modeling programs use bond lengths and angle constraints from idealized geometry to compute the location of the hydrogen atoms. However, the crystal structure is never idealized. It is a snapshot of the protein structure at a certain moment in time. NMR models usually contain hydrogens but the geometry is far from idealized as well.

Even small clashes and deviation from the idealized geometry may "blow up" the energy function. As seen above, it is especially the Lennard-Jones potential function that blows up very quickly even with very small deviations from the ideal distance (see above). The potential function can be minimized with respect to the x, y, z coordinates. The minimization is a process where the atoms are rearranged, usually through very small displacements, in a way that reduces the overall value of the potential function. This process is also called relaxation or geometry optimization. Figure 10 shows a simplified example of a function with several local minima. The actual potential energy function for a real molecule is much more complicated, but the idea is similar: We start out from some non-minimum point, and find a way to get to the nearest minimum. Most energy minimization methods we will describe below will get us to the nearest local minimum and not the global minimum, but usually this is good enough for our purpose of only getting rid of clashing atoms.

5.1.2 Common Minimization Protocols

There are several energy minimization techniques. If we go back to our discussion in Section 4 above, the force is the gradient (directional derivative) of the potential energy function with respect to the $\vec{r} = (x, y, z,)$ coordinates. Therefore, if we differentiate the potential energy with respect to $\vec{r}$ and take a step in the opposite direction of the gradient, we will approach the minimum. There are several minimization methods:

1. First order algorithms: Steepest descent, Conjugated gradient

2. Second order algorithms: Newton-Raphson, Adopted basis Newton Raphson (ABNR)

Steepest Descent or Gradient Descent: This is the simplest minimization method. It is known that the force decreases fastest in the direction of the negative gradient of the force. The first directional derivative (gradient) of the potential is calculated and displacement is added to every coordinate in the opposite direction (the direction of the force). The step is increased if the new conformation has a lower energy. Although this method is simple and fast, it is not always accurate and usually does not converge.

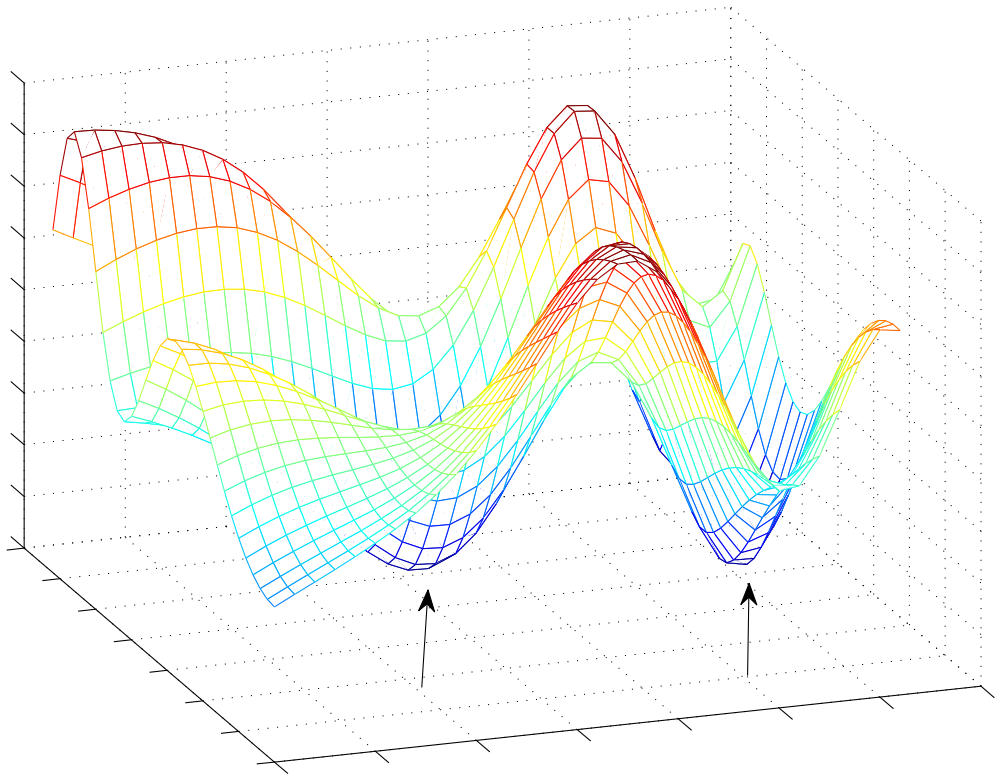


Figure 10: Energy minimization aims to find a local or global minimum in an energy function.

Conjugated Gradient: This is a more sophisticated method. It uses first derivative information + information from previous steps – the weighted average of the current gradient and the previous step direction. The weight factor is calculated from the ratio of the previous and current steps. This method converges much better than Steepest Descent.

Newton-Raphson's Algorithm: This method uses both first derivative (slope) and second (curvature) information. In the one-dimensional case: $x_{k+1} = x_k + \frac{F'(x_k)}{F''(x_k)}$ In the multi-dimensional case it is much more complicated (calculates the inverse of a hessian [curvature] matrix at each step) It is accurate and converges well, but computationally expensive. For convergence, it should start near a minimum.

Adopted Basis Newton-Raphson's Algorithm (ABNR): This is an adaptation of the NR method that is especially suitable for large systems. Instead of using a full matrix, it uses a basis that represents the subspace in which the system made the most progress in the past. Advantage: Second derivative information, convergence, faster than the regular NR method. Disadvantages: Still quite expensive, less accurate than NR.

5.1.3 Assignment of Initial Velocities

At the beginning the only information available is the desired temperature. Initial velocities are assigned randomly according to the Maxwell-Bolzmann distribution which associate velocities with temperature:

$$P(v)dv = 4\pi(\frac{m}{2\pi k_B T})^{\frac{3}{2}} v^2 e^{\frac{-mv^2}{2k_B T}}$$

where P(v) – the probability of finding a molecule with velocity between v and dv.
Note that:

1. The velocity has x,y,z components.
2. The velocities exhibit a gaussian distribution

Bond and Angle Constraints (SHAKE Algorithm)

Some bond lengths and/or angles can be constrained to fixed values using a restraining force G_i.

$$m_i a_i = F_i + G_i$$

First, we solve the equations once with no constraint force. Then we determine the magnitude of the force (using lagrange multipliers) and correct the positions accordingly. Finally, we iteratively adjust the positions of the atoms until the constraints are satisfied. This is especially useful in constraining bonds including hydrogens, thus saving time.

5.1.4 Equilibrating the System

Velocity distribution may change during simulation, especially if the system is far from equilibrium. An equilibration period is necessary, usually lasting a few thousand time steps. The equilibration

stage is used to equilibrate kinetic and potential energies, i.e., to distribute the kinetic energy ?pumped? into the system during heating among all degrees of freedom. During this period the system is coaxed towards the desired thermodynamic state point (defined by temperature and density) by a technique known as temperature scaling: Perform a simulation, scaling the velocities occasionally to reach the desired temperature.

The system is at equilibrium if: The quantities – energy, temperature etc. fluctuate around an average value. and the averages remains constant over time.

5.1.5 Production Run

After the system has reached equilibrium, the production stage can start. During this stage, the MD simulation is run for as long as needed to collect the desired information about the system.

The Verlet Integration Method Taylor expansion about r(t):

$$r(t+\delta t) = r(t) + v(t)\delta t + \frac{1}{2}a(t)\delta t^2 + \dots$$
$$r(t-\delta t) = r(t) - v(t)\delta t + \frac{1}{2}a(t)]\delta t^2 + \dots$$

Adding the two terms gives a velocity independent term:

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

The odd terms go, so the error is the order of magnitude of δt^4, the next term Velocities can be calculated via the derivation method:

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

Here the error is of order δt^2.
Note – the time interval δt is in the order of 1fs. (10^{-15}s)

1. Start with $r(t)$ and $r(t-\delta t)$
2. Calculate a(t) from the Newton equation: $a(t) = f_i(t)/m_i$.
3. Calculate $r(t+\delta t)$ according to the aforementioned equation.
4. Calculate $v(t)$.
5. Replace $r(t-\delta t)$ with $r(t)$ and $r(t)$ with $r(t+\delta t)$.
6. Repeat as desired.

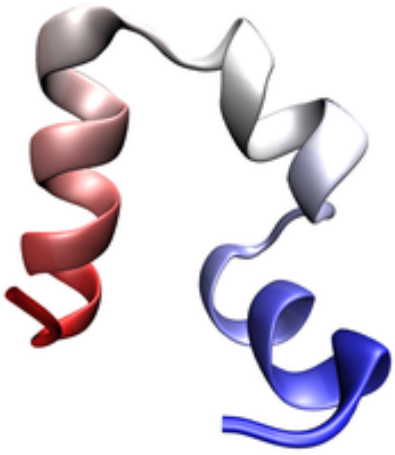
6 QM/MM

Hybrid quantum mechanics/molecular mechanics (QM/MM) simulations have become a popular tool for investigating chemical reactions at multiple scales of detail. In QM/MM methods, the region of the system in which the chemical process takes place is treated at an appropriate level of quantum chemistry theory, while the remainder is described by a molecular mechanics force field. Within this approach, chemical reactivity can be studied in large systems, such as enzymes. The region around the active site can be modeled by QM, while the rest of the protein can be modeled with classical molecular mechanics.

7 Coarse Grained MD

7.0.1 Case Study – Vilin Headpiece Simulation

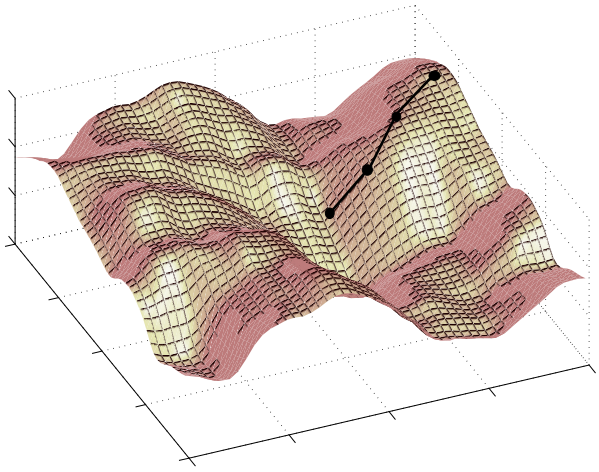
This protein folds very fast – 4-5ms It has a mutant folds in under 1ms. This allows us to characterize the folding process in all-atom explicit solvent simulation.

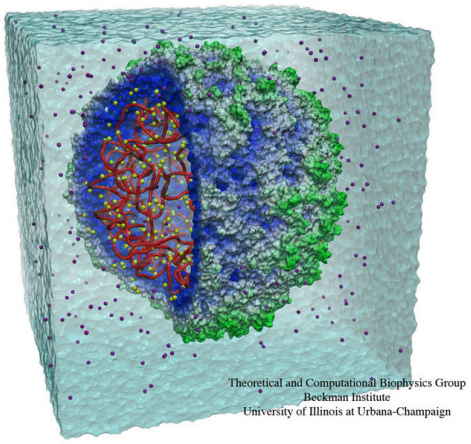


`http://www.ks.uiuc.edu`

Case Study – Simulating an Entire Virus

Viruses contain two components: the capsid (a protein shell), and a genome. MD shows the assembly and disassembly of several viruses as part of the virus life cycle. STMV (Satellite tobacco mosaic virus) particle consists of 60 identical copies of a single protein that make up the viral capsid (coating), and a 1063 nucleotide single stranded RNA genome which codes for the capsid and one other protein of unknown function.





`http://www.ks.uiuc.edu`

MD Shortcomings – Expensive!

Small integration time step (10^{-15} sec). Complex interactions between atoms in the molecule. Simulating 1ns of a medium sized protein (300+ amino acids, approx. 100,000 atoms incl. solvent) requires millions of calculations per step X 1,000,000 steps. Must use distributed computing to scale up to reasonable sized systems.

Replica Exchange MD (REMD)

Many properties of a molecular system, such as barriers between energy minima, can be difficult to cross at room temperatures over accessible simulation time scales. Since MD simulations usually sample just one path of the conformational space, the results depend a lot on the choice of initial conditions, because these determine the region of space that is explored by the simulation. Replica

exchange simulations (also known as parallel tempering) seek to enhance the sampling relative to standard MD by running multiple independent replicas in slightly different conditions (usually different temperatures), and periodically exchanging the coordinates of replicas between the ensembles. This allows to overcome energy barriers on the potential energy surface. Every period of time (at least 1ps) replicas are exchanged among close-by temperatures. The exchange relies on doing Monte Carlo moves of replicas between ensembles. The probability of observing a replica in a particular state depends on the potential energy and the temperature: $P(x) \propto \exp -\frac{U(x)}{k_B T}$. The range of probabilities that are observed follow a normal distribution (because of the Central Limit Theorem). If two ensembles are chosen so that their distribution of states have significant overlap, then when we observe a state in one there is appreciable chance that it could have been observed in the other.

The move suggests that we exchange the coordinates found in two replicas based upon the probabilities that they would have been observed in the two ensembles. The exchange is accepted only if a random number has a suitable value. When done correctly, detailed balance is achieved, and the sampling in both ensembles is correct whether or not an exchange took place. How do we select the temperature range? The highest temperature is chosen so that the energy barriers can be crossed. Replicas wander up and down through temperature space as exchanges are accepted. Barriers are crossed probabilistically at all of the temperatures, but at a higher rate at the higher temperatures. Those states filter down to the lower temperatures if they "belong" to the corresponding ensembles. As a rule of thumb, a 0.2 exchange probability is a good choice.

Targeted MD (TMD)

A subset of the atoms is guided towards a final target structure using a steering force. The steering force is assigned for each atom using the gradient of the following potential: $U_{TMD} = \frac{1}{2}\frac{k}{N}\left[RMSD(t) - RMSD^*(t)\right]^2$ $RMSD(t)$ is the least RMSD of the current coordinates with the target coordinates at time t. $RMSD^*(t)$ evolves linearly from the initial RMSD at the first TMD step to the final RMSD at the last TMD step. The spring constant k is scaled down by the number N of targeted atoms.

Steered MD (SMD)

SMD is one of the methods used to calculate free energy changes. The basic idea behind SMD is to apply an external force to one or more atoms, which are referred to as SMD atoms. During the run of the simulation runs the atoms adjust to the forced change, so that conformations may be sampled along a particular pathway, while another group of atoms may be held fixed. As the force is executed and motion occurs along a coordinate, the potential energy (or *Potential Mean Force*, PMF) of the system is calculated The PMF is related to the free energy change for the process.

The NAMD [**?**] guide for SMD states that the center of mass of the SMD atoms is harmonically constrained with force constant k to move with velocity v in the direction $\vec{n}$. SMD thus results in the following potential being applied to the system:

$$U(\vec{r}_1, \vec{r}_2, ..., t) \;=\; \frac{1}{2}k\left[vt - (\vec{R}(t) - \vec{R}_0)\cdot\vec{n}\right]^2.$$

Here, $t \equiv N_{ts}dt$ where N_{ts} is the number of elapsed timesteps in the simulation and dt is the size of the timestep in femtoseconds. Also, $\vec{R}(t)$ is the current center of mass of the SMD atoms and R_0 is the initial center of mass as defined by the coordinates. Vector $\vec{n}$ is normalized before being used.

This enables us to study the behavior of your protein under various conditions in accelerated time, if the process were to take place anyway, but in a much longer time frame. An example is – (un)folding and binding events that do not happen under MD time scales.