Molecular dynamics: We have a system composed of particles. How do positions and velocities of the particles vary with time? In order to answer this, molecular dynamics uses Newton's laws of motion:

1) A body continues to move in a straight line at constant velocity unless a force acts upon it.
2) Force equals the rate of change of momentum.
3) To every action there is an equal and opposite reaction.

In order to obtain the trajectory, we use the equation $F = m \cdot a$:

$$a = d^2x/dt \rightarrow d^2x/dt = F/m$$

Being d^2x/dt the second derivative of the position of the particle with respect to time (= acceleration), m the mass of the particle and F the force on the particle in direccion x.

We can face 3 different types of problems:

A) No force acts on each particle between collisions → the position changes by $v \cdot \delta t$.
B) Constant force between collisions → Ex: a charged particle moving in a uniform electric field.
C) The force on a given particle depends on its position with respect to other particles → very difficult to solve analytically.

First MD simulation: Alder and Wanwright, 1957

- **Hard-sphere** model: spheres move at constant velocity in straight lines between collisions.
- All collisions are perfectly elastic: the total kinetic energy of the two bodies after the encounter is equal to their total kinetic energy before the encounter. Kinetic energy is not transformed into other types of energy in the collision.
- Collisions occur when the separation between the centres is equal to the diameter of the spheres (obvious).
- Some other simulations used the **square-well potential**, where the interaction energy is:
  - Too far away (above a cutoff distance σ_2)→ Zero
  - Too close (below a cutoff distance σ_1)→ Infinite
  - Between the two cutoffs → v_0
  -
- Steps: identify the next **two particles** to collide → calculate **positions** of those particles at the collision time → calculate **new velocities** after the collision (principle of conservation of linear momentum)→ repeat
- These simulations were very useful to study fluids at a microscopic level.

Molecular dynamics with continuous potentials

- More realistic
- The force on each particle changes whenever the particle changes its position, or whenever any of the other particles interacting with it change their positions
- The motions of all the particles are coupled together → many-body problem → cannot be solved analytically → we use a **finite difference method**

Finite difference methods

- Integration is broken down into many small stages, each separated in time by a fixed time-step δt
- Total force on a particle at time t: vector sum of its interactions with other particles
- How do we calculate the positions and velocities at time $t + \delta t$?
  From the force we determine acceleration, then we combine accelerations with positions and velocities at time t and we calculate positions and velocities at time $t + \delta t$
- Many algorithms to implement this:

Algorithm	Advantages	Disadvantages
Verlet	- Straightforward implementation - Modest storage requirements	-Loss of precision. Positions are obtained by adding very smal terms to large terms. -Velocities are not explicit. They are difficult to obtain (we need to compute the positions at the next step first). - NOT self-starting. At t=0, we have no “previous position”.
Leap-frog	- Explicitly includes velocity - Does not require the calculation of the differences of large numbers	- Positions and bvelocities are NOT synchronised
Velocity Verlet	- Gives positions, velocities and accelerations at the same time t. - Does not compromise precision.	
Beeman	- More accurate expressions for velocity → better energy conservation	- Expressions are more complex. Computationally more expensive.

Which integration algorithm is most appropriate? Criteria:

- Most important: computational effort. But, consider that an algorithm that is nominally more expensive may nevertheless be more cost-effective if it permits a significantly longer time step.

- Important: energy conservation. We can plot root-mean-square fluctuation against time step.
- Synchronisation of positions and velocities?
- Is the algorithm self-starting?

How to choose the time step?

- Too small → the trajectory will cover only a small proportion of the phase space
- Too large → High energy overlaps → instabilities in the integration → violation of the energy and momentum linear conservation
- Example: two argon atoms under the Lennar-Jones potential. We determine the behaviour of the system analytically and numerically (using different time steps). We compare the interatomic distance in the simulations (with both time steps) to the analytical potential. Result: first the numerical trajectory lags behind the analytical one, and then the atoms move too fast in the numerical simulation.
- Tips: atomic fluid → time step should be small compared to the mean time between collisions. Flexible molecules → time step should be aprox. 1/10 of the shortes period of motion.

<u>Setting up and running a MD simulation</u>

First we have to set the initial configuration of the system. We can obtain the initial configuration in different ways:

- From experimental data
- From a theoretical model
- A combination of both

How can we assign the initial velocities of the atoms? Maxwell-Boltzmann equation: probability that an atom of mass m has a velocitu v in the direction x at a temperature T.

Now the simulation can begin. At each step the force on each atom must be calculated by differentiating the potential action. The force includes the contributions from various terms: bonds, angles, torsional terms and non-bonded interactions.

- The force between to atoms is equal in magnitude and opposite in direction, and it applies along the line connecting the two centres.

- In order to calculate each force just once, we use an algorithm that has two nested loops. We compute the forxe between an atom and those atoms that have a higher index. Since the forces for two atoms are equal but opposite in direction, we only need to store those values once.

1st stage of the simulation: **equilibration phase** → to bring the system to equilibrium from the starting configuration. Various parameters are monitored during this phase, when these parameters achieve stable values, we have reached equilibrium. Parameters: kinetic energy, potential energy, total energy, velocities, temperature, pressure.

2nd stage: **production phase** → thermodynamic properties are calculated.
During the simulation:

- Kinetic energies may fluctuate, but total energy should remain constant.
- The components of the velocities should describe a Maxwell-Boltzmann distribution
- Kinetic energy should be equally distributed amon the three directions x, y and z.

Simulating and inhomogeneous system: we usually need a more detailed equilibration phase. For example, if we simulate a macromolecular solute like a protein. We want the solvent to completely readjust to the potential field of the solute.

- We keep the solute fixed and we minimise the energy of the solvent.
- Then the solvent is allowed to evolve using either a MD or a Monte Carlo simulation, again keeping the solute molecule fixed.

Calculating the temperature
The instantaneous value of the temperature is related to the kinetic energy.

Constraint dynamics
Constraint: a requirement that a system is forced to satisfy. For each constraint force there is an equal and opposite force. Thus, constraint forces do no work.

- **Holonomic constraints**: they can be expressed as a function of the coordinates of the particles.
- **Non-holonomic constraints**: they cannot be expressed this way.

Restraints: it encourages the system to take some particular values.

Time-dependent properties
Also called time correlation coefficients. Since MD simulations yield configurations of the system that are connected with time, we can recover that information at the end of the simulation.

- **Correlation** functions: the value of one property of the system at time t is correlated to the value of another property at the same time.
  - **Cross-correlation function**: when the correlated properties are not the same.
  - **Autocorrelation function**: it indicates the extent to which the system retains memory of its previous values. The time it takes to lose the correlation is often called the **relaxation time.** Velocity autocorrelation function

Things missing here
Molecular dynamics at Constant Temperature and Pressure

Recent progress in the study of G-coupled receptors with molecular dynamics computer simulations

Importance of GPCRs:

- Very numerous
- Very important in biomedicine and drug design

BUT: as they are membrane proteins, they are difficult to study:

- They require a membrane.mimetic environment to remain folded
- Difficult to overexpress and purify them

As a result, their functional aspects are very-well characterised but their structure remains poorly understood. Knowing their structure would be a huge step in drug desing.

First high-resolution GPCR crystal structure: **dark-state bovine rhodopsin**

- Very useful, since it was the first atomic level view of a GPCR.
- BUT: rhodopsin is a somewhat unusual GPCR (its ligand bind to it covalently). Can these finding be applied to other GPCRs?
- Afterwards, many other structures were solved for inactive rhodopsine.

Second GPCR crystal structure: **beta2-adrenergic receptor (B2AR)**

After this, several attempts were made to obtain a structure for a **more active form** of the **rhodopsin**. They tried to crystalize:

- The apo form of the rhodopsin (opsin), that is to say, rhodopsin when it is not bound to its retinal ligand.
- Rhodopsin without a protein analog bound.
- Rhodopsin in the presence of all-trans retinal (which is the agonist form of the ligand).
- Structures of dopamine and chemokine receptors were obtained.

So there was an explosion of new structural information, which created an oportunity for computational methods to make major contributions to our understanding of GPCRs, their functions and their dynamics.

Molecular dynamics simulations have been carried out, using the structures of B2AR, B1AR and A2A as a starting point.

RHODOPSIN

The dimlight receptor in the mammalian eye, rhodopsin, works as the hydrogen atom for GPCRs in many ways. Why do we study rhodopsin?

- GPCRs are usually found in very low concentrations in the cell, while rhodopsin is found in high concentrations
- Rhodopsin is more tolerant to changes in the lipid-protein ratios than other GPCRs. This meakes it easier to study them using biophysical methods. For example, crystalization of rhodopsin was possible because of this.

Rhodopsin has also been used in MD simulations, to study the role of lipid-protein interactions in the modulation of rhodopsin function. For example: what is the role of oligomerization in rhodopsin? What is the link between protein-lipid interactions and oligomerization? What are the conformational changes that rhodopsin suffers druing its activation process?

Lipid-protein interactions in the dark state:

The environment surrounding a biomolecule is critical to its behavior. For membrane proteins, the lipid composition of the bilayer alters the stability and the efficiency of the proteins embedded in it.

The native environment of rhodopsin is called the rod outer segment (ROS), and has very unusual lipid compositions:

- High concentrations of **polyunsaturated omega-3 fatty acids** (even though mammals cannot synthesize them themselves). Plus: the overall abundance of omega-3 fatty acids in an organism is about 5%.

- Quite high concentrations of **cholesterol**. Also, cholesterol is not uniformly distributed: its concentation is very high in immature ROS disks and gradually drops as the disks migrate toward

the top of the stack.

Conclusion: the cell is actively regulating the lipid-composition of the ROS membranes. Almost all of the protein in those membranes is rhodopsin. Indeed, experimental in vitro works have shown that both omega-3 fatty acids and cholesterol have significant effects on rhodopsin's activity:

- Polyunsaturated lipids enhance rhodopsin's function. They push the MetaI/MetaII equilibrium toward the active MetaII state.

- Cholesterol decreases rhodopsin's function.

Specific vs bulk effect
But, is this a specific effect of polyunsaturated lipids and cholesterol on rhodopsin? Or is it rather a consequence of a change in the bulk properties of the membrane? For example, omega-3s increase the disorder of lipid bilayers, while cholesterol increases their order. This could change the properties of the membrane (elasticity, etc) and thus affect rhodopsin activity, in a non-specific fashion. Alternatively, they could havespecific interactions with rhodopsin.

Evidence:

- A molecular dynamics simulation of rhodopsin in a SDPC membrane: there was a clear preference for the omega-3 chains to interact with the rhodopsin and to exclude staturated lipids.

- MD simulations with a realistic lipid composition (SCPC, SDPE, cholesterol). Since the lateral reorganization of the membranes occurs in the microsecond scale or slower, they performed 26 separate simulations of 100ns each. Each time they rebuilt the bilayer to ensure that a number of truly independent bilayer conformations were explored. Results: the density of omega-3s at the surface of rhodopsin was enhanced. Moreover, 8 distinct sites of the rhodopsin surface were identified as putative specific binding sites for docohexanoyl chains. Later, they showed this preference was entropically driven (omega-3s experience a lower entropic penalty than saturaded chains when interacting with the protein surface).

- The headgrup composition can also affect rhodopsin function. Particular PE headgroups enhance it. Problem: the time scale of lateral reorganization is too long for it to be studied through all-atom simulations.

There is also a debate going on about the link between **oligomerization** and GPCR function. Some evidence:
- Homo and heterodimerization happen under some conditions (this has been shown experimentally).
- Rhodopsin forms dimers, and other GPCRs form heterodimers (computationally).
- Oligomerization is modulated by lipid-protein interactions (MD simulations using lipid bilayers with varying degrees of hydrophobic thickness).

Dimerization happens, but we still do not know for sure which the physiologically relevant state (monomer/dimer) of the rhodopsin is .
- It appears that rhodopsin works as a monomer.
- In its native ROS membranes, rhodopsin mainly takes the form of a monomer.
- Rhodopsin is capable of binding G protein in its monomeric form.

Which state should be simulated, the monomeric or the dimeric form of the rhodopsin? They have mostly used the monomeric form so far. But, is it a good model for in vivo behavior?

It is difficult to model dimers. Technical problems:
- The system to be simulated becomes larger → the computational cost is increased and the length of the trajectory that can be obtained from the simulation is reduced.
- A longer trajectory would be required to obtain equivalent statistical sampling (which is already difficult for monomer simulations).
- We do not know which dimer (or oligomer) to use. Dimers constructed might be unaccurate. There is

no way to know if we chose a bad starting dimer, because the mis-packed dimer could persist in the MD simulation for hundreds of nanoseconds.