

VU Research Portal

Molecular Dynamics

Mouhib, Halima; Gils, Juami H. M. van; Gavaldá-García, Jose; Hou, Qingzhen; May, Ali; Rauh, Arriën Symon; Vreede, Jocelyne; Abeln, Sanne; Feenstra, K. Anton

2023

DOI (link to publisher)

[10.48550/arXiv.2307.02176](https://doi.org/10.48550/arXiv.2307.02176)

[Link to publication in VU Research Portal](#)

citation for published version (APA)

Mouhib, H., Gils, J. H. M. V., Gavaldá-García, J., Hou, Q., May, A., Rauh, A. S., Vreede, J., Abeln, S., & Feenstra, K. A. (2023). *Molecular Dynamics*. <https://doi.org/10.48550/arXiv.2307.02176>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

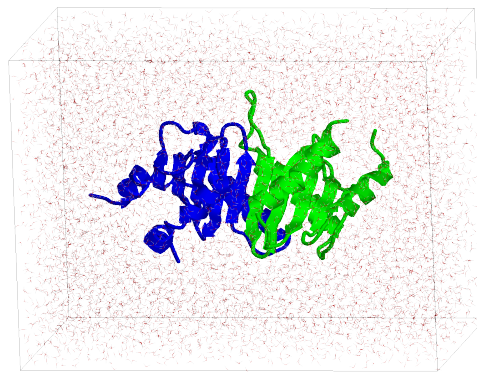
E-mail address:

vuresearchportal.ub@vu.nl




Chapter 14

Molecular Dynamics

Halima Mouhib*  Juami H. M. van Gils 
Jose Gavalda-García  Qingzhen Hou  Ali May 
Arriën Symon Rauh  Jocelyne Vreede 
Sanne Abeln*  K. Anton Feenstra* 



* editorial responsibility

14 Molecular Dynamics **1***Halima Mouhib**  *Juami H. M. van Gils*  *Jose Gavaldá-García* *Qingzhen Hou*  *Ali May*  *Arriën Symon Rauh* *Jocelyne Vreede*  *Sanne Abeln**  *K. Anton Feenstra** 

1 Introduction to molecular dynamics	4
1.1 Simulating a protein by classical physics	4
2 Relevant time and length scales	6
3 Forces & interactions	9
3.1 Force fields	9
3.2 Interactions	10
3.3 Parameters	12
4 Dynamics	15
4.1 Integrating equations of motion	17
The Verlet integration scheme	17
Choosing timesteps	17
4.2 Convergence of state properties	24
Enhanced Sampling techniques	24
Observing sampling – RMSD	25
4.3 Temperature dependence	30
Simulated annealing	30
4.4 Homology model optimization	31
5 Outlook and summary	34
6 Key concepts	34
7 Further reading	35
References	35

1 Introduction to molecular dynamics

We know that many proteins have functional motions, and we already introduced this in Chapter “Structure determination”, Section 6. One famous example presented there in Panel “Allosteric motions and time-resolved crystallography”, is the cooperative binding of oxygen to hemoglobin, where the first oxygen binding induces a conformational change throughout the tetrameric structure, which make subsequent oxygen binding much more favourable. However, experimentally, such motions are hard to observe. Exceptions are when a protein can be crystallized in multiple distinct conformations, when fluorescence experiments are able to capture fast events, or in the very rare cases where time-resolved crystal structures can be recorded. A dramatic manifestation of functional motion occurs for some enzymes that are still active in the crystal form (which is relatively common). Large scale motions of the protein structure may occur during catalysis of the reaction, and sometimes these motions physically break the crystal when the substrate is added to the crystal! (This shows how strong molecular motions can be.) As an alternative to experimentally studying protein motions, molecular dynamics (MD) simulations have been used since the 1960s. These allow us to simulate all atomic motions in detail, but of course within the restrictions of the accuracy of the molecular models used.

In this chapter we will introduce MD simulations, which represent an important method to investigate the dynamic behaviour of proteins and polymers. Basically, in such simulations, the forces and interactions between particles are used to numerically derive the resulting three-dimensional movement of these particles over a certain time-scale. The main emphasis will lie on explaining the basic method of MD simulations, when and how to use them, and provide some examples of how to evaluate the results. More specifically, we will describe the physical interactions between atoms and the algorithms used to perform these simulations. Secondly, we will see how simulation results may be interpreted. For this purpose, some basic knowledge on thermodynamics and statistical mechanics, as introduced in Chapter “Thermodynamics of Protein Folding” is essential. It is particularly important to understand the relation between free energy and probability, in order to analyse simulation results.

Although a wide variation of problems (not necessarily of biological relevance) can be addressed using MD simulations, in the course of this book we will focus mainly on proteins.

1.1 Simulating a protein by classical physics

Molecular dynamics (MD) simulations, using classical or Newtonian physics, is currently the most common tool to investigate the dynamics and dynamic ensembles of proteins and other large biomolecules at a molecular level (e.g.,

Adcock and McCammon, 2006). There are several branches of the technique that can be used to elucidate complicated or ambiguous results from biophysical experiments such as Atomic Force Microscopy (AFM) or Nuclear Magnetic Resonance (NMR) (Kumar and Li, 2010). Most importantly, MD can provide insights into the molecular mechanisms through which proteins perform their functions; this can range from binding to a specific substrate, to proteins that can bind to each other, to the complete cycle of movement of molecular motors (e.g., Karplus and Kuriyan, 2005).

Even though MD is used ubiquitously, it is important to remember that simulation methods and descriptions of interactions used are necessarily an abstraction of the real world (see in particular also the Panel “Limitations of Newtonian physics and force fields”). Therefore, any hypothesis on mechanistic workings obtained from MD simulations should be verified with experimental evidence. This way, the biological, physical and scientific relevance of the results can be guaranteed. Fortunately, Newtonian physics suffices for the vast majority of biologically relevant phenomena, because we are typically interested in macroscopic properties which depend on averages obtained from the ensembles observed in the simulation. Please refer to Chapter “Thermodynamics of Protein Folding” for more detail on the statistical dynamics of ensembles.

In Newtonian physics, interactions are described as forces, and a force applied to an object triggers an effect. Figure 14.1 shows how the interaction (by way of force F) between two particles leads to changes in positions of both. This way, the structural and dynamical properties of large biological systems can be investigated at a molecular level. However, to simulate the large collection of atoms that make up biological systems, we need to resort to numerical approximations, as analytical solutions are not available (this

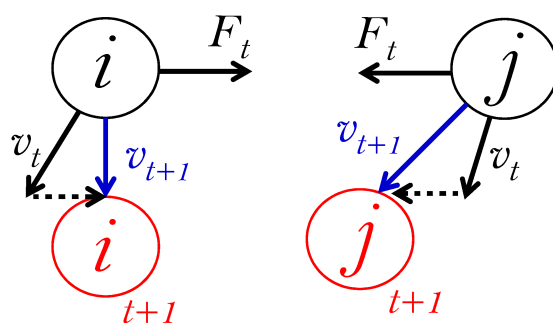


Figure 14.1: Two particles i and j at time t with initial position (indicated by black spheres) and velocities v_t (black arrows), and exerting a force F_t on each other (also black arrows; note that between two particles $F_{t;i,j} = -F_{t;j,i}$). These forces cause the velocities to change at the next time step $t + 1$, as indicated by v_{t+1} in blue arrows, and the new velocities cause the positions to change as well (red spheres).

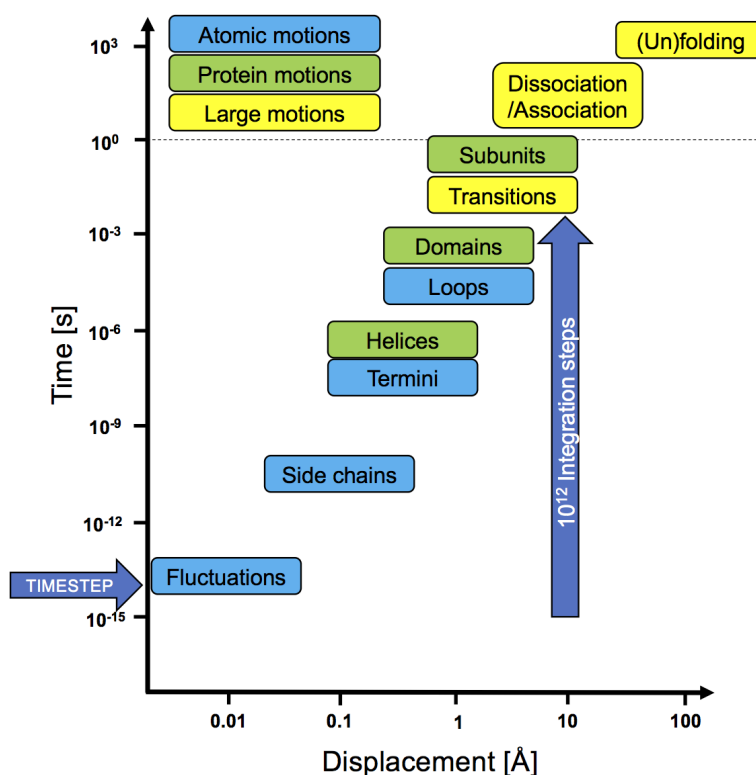


Figure 14.2: A simplified overview of the order-of-magnitude time and length scales of molecular motions that are of biological interest; the boxes indicate atomic motions (blue), protein domain motions (green), and large biological motions (yellow). The x axis corresponds to displacement of atoms or molecules, and the y axis indicates the approximate time scales. Note that both axes are on a logarithmic scale. The size of the MD timestep (2 fs) is indicated by the arrow on the left; the arrow on the right indicates that a thousand billion (10^{12}) timesteps are needed to reach the shortest biologically relevant timescales.

is known as the ‘multi-body’ problem in physics). Moreover, to obtain an accurate approximation, we need very small integration (time) steps, which is the main limit on computational efficiency. This is what much of the current chapter will deal with.

2 Relevant time and length scales

To give some perspective to the phenomena we will be describing in molecular simulations, it is good to realize which time and length scales are involved. Figure 14.2 shows an overview of the relevant time and length scales of several classes of biological processes. Please note that these are order-of-magnitude indications that should not be taken as absolute values.

Summarizing:

- At the lower end of the scale are small-scale fluctuations of individual atoms over fractions of an Ångström (Å , 10^{-10} m) distance and at timescales of tens to hundreds of femtoseconds (10^{-15} s).
- Amino acid side chains that involve several to tens of atoms move tenths of an Ångström per picoseconds (10^{-12} s).
- Rotamer changes, conformational changes arising through rotation around rotatable bonds and sidechains, may produce larger motions, i.e., in the order of Ångströms, and will be slower, tens to hundreds of picoseconds.
- The termini and loop regions, which involve several residues and thus many more atoms than a single sidechain are still slower, moving up to several Ångströms anywhere in the range of nano- (10^{-9}) and microseconds (10^{-6}).
- As helices are relatively stable structural (and dynamical) units, they move around the same scales as termini regions and loops.
- Larger parts, such as domains or subunits are again larger and therefore move slower, around tens of Ångströms or even more and may take up to seconds in the slowest of cases.
- Conformational transitions take place in the same time-range as domain motives. These transitions often involve the motion of domain-size parts of a protein, or for example the association or dissociation of protein binding (e.g. in protein-protein interactions).
- The typical folding time for a protein is on the order of seconds, nevertheless ‘fast folders’ may take only milliseconds.

For accurate simulation, the timestep used should be smaller than the fastest motions. Fastest vibrations involve hydrogen atoms (which are light), and/or particularly stiff angles, with vibrational periods of around 10 femtoseconds. Therefore, typically timesteps of 1 or 2 fs (10^{-15} s) are used, as indicated with an arrow at the bottom left in Figure 14.2. It should be noted that an order of 10^{12} integration steps are then required to start getting into biologically relevant (millisecond) timescales as, indicated with a vertical arrow at the right in Figure 14.2. This computational power is slowly coming within range of being feasible. But, in practice this means we are not routinely able to fold or unfold proteins in MD simulations, simply because we can not simulate for long enough. Note, also, that just simulating for longer may also not simply solve the problem, as the results would still depend on the accuracy of the force fields we use. We will deal with force fields in section 3.1. Moreover some effects, such as quantum effects, are hard to model accurately. In a large-scale survey of the ability of current force fields to improve upon homology models, (Lindorff-Larsen *et al.*, 2012; Raval *et al.*, 2012) observed the accumulation of errors in long simulations, highlighting that this is an as yet unsolved problem.

Nevertheless, numerous interesting problems of biological relevance such

as Amyloid aggregation in Alzheimer's (Lemkul and Bevan, 2013) and T-cell receptor-MHC interactions (Cuendet *et al.*, 2011) have been investigated using simulations that might strictly be considered too short. These simulations have fortunately shown the interesting and relevant dynamical molecular behaviour that helps answering several important biological questions.

Historical background

A historical overview on the evolution and applications of MD simulations is given in (van Gunsteren *et al.*, 2006). The rough estimates of future simulation times are extrapolated from past achievements based on the size of those systems and assuming the continued growth of computer power by Moore's law by doubling every 18 months (Moore, 1965). Assuming that will hold, it may take till the end of this century before we can simulate protein folding at the speed it occurs in nature – but even then only a single protein, and only if the accuracy of our force fields do not remain limiting.

year	system	time-scale	reference
1936	Gelatine balls		Morrell and Hildebrand
1953	MC Simulations		Metropolis <i>et al.</i>
1957	MC of Lennard-Jones spheres		Wood and Parker
1964	MD of liquid Argon	10 ps	Rahman
1970's	Non-equilibrium methods		
1970's	Stochastic dynamics methods		
1974	MD of liquid water		Stillinger and Rahman
1977	MD of protein in vacuum	20 ps	McCammon <i>et al.</i>
1980's	Quantum-mechanical effects		
1983	MD of protein in water	20 ps	van Gunsteren <i>et al.</i>
1998	MD of reversible peptide folding	100 ns	Daura <i>et al.</i>
1998	MD of protein folding	1 μ s	Duan and Kollman
2010	MD of reversible small protein folding	1 ms	Shaw <i>et al.</i>
Today	Large proteins or complexes in water or membrane	up to milliseconds ($\sim 10^{12}$ – 10^{14} slower than nature)	
2029	Protein folding	1 ms	???
2034	E-coli, $\sim 10^{11}$ atoms	1 ns	
2056	Eukaryotic cell, $\sim 10^{15}$ atoms	1 ns	
2080	Protein folding	as fast as in nature	

The field of molecular simulations was initiated around 1930, when Morrell and Hildebrand (Morrell and Hildebrand, 1936) investigated the distribution of gelatine spheres, compared to X-ray experiments on atoms. The subsequent rapidly growing number of applications was enabled through to the continuous development and improvement of computational facilities over the last 50 years. This allowed for more and more detailed theoretical investigations of larger systems over longer timescales. Along with this development, simulations have moved from hard-sphere systems (Alder and Wainwright, 1957) and simple mono-atomic systems (e.g. argon) in the 60's (Rahman, 1964), to proteins and water separately in the 70's (Stillinger and Rahman, 1974; McCammon *et al.*, 1976), to proteins in water in the 80's (van Gunsteren *et al.*, 1983), and finally, to something close to the equilibrium behaviour of peptides and (small) proteins in water in the 90's and early 2000's (Daura *et al.*, 1998; Duan and Kollman, 1998; Shaw *et al.*, 2010). Generally it can be said that MD simulation became important for biophysics in the late 70's and early 80's with the first simulations of a protein in water. More biological relevance perhaps can be said to start with the simulation of reversible, equilibrium peptide folding (late 90's) and protein folding (early 2000's).

3 Forces & interactions

If we want to understand how particles move, we need to understand how they interact with each other, or more precisely, with what forces they repel or attract each other. The basis of all molecular simulations is the so-called *force field*. Force fields consist of a collection of all forces which are considered to occur in a system of interest, for example all atoms within a protein and the surrounding solvent. Forces, as noted before, correspond to interactions, in this case between individual atoms in the system. Note that these interactions and therefore the energies associated with them are dependent on the types of atoms involved, as well as their positions with respect to each other. Formally, the force F on particle i is the derivative of the energy U , depending on the position r_i of particle i , and is given as:

$$F_i = -\frac{\partial U}{\partial r_i}, \quad (1)$$

3.1 Force fields

Force fields can be written as interaction energies; in the next section on Interactions we will see how interaction forces and interaction energies are related. In molecular mechanics, interaction energies between individual

atoms in the system contain the following terms for interaction energies:

$$U_{total} = U_{bonded} + U_{non-bonded} + U_{crossterm} \quad (2)$$

Thus, in molecular systems, a distinction is made between bonded interactions, non-bonded interactions and ‘other’ interactions called ‘crossterm’ in the formula. Bonded interactions act in between atoms close by in the same molecules; most notably two atoms in a bond, three in an angle and four in a dihedral:

$$U_{bonded} = U_{bond} + U_{angle} + U_{dihedral} \quad (3)$$

Non-bonded interactions act in between molecules and also in between atoms in the same molecule: these are Coulomb’s (electrostatic) and Van der Waals’ (atomic contact) interactions.

$$U_{non-bonded} = U_{Coulomb} + U_{VanderWaals} \quad (4)$$

Combining Equations 2-4, the total energy may thus be specified as:

$$U_{total} = U_{bond} + U_{angle} + U_{dihedral} + U_{Coulomb} + U_{VanderWaals} + U_{crossterm}. \quad (5)$$

Each of these energy terms will be specified in more detail in the next section. The main assumption here is that the total interaction (e.g. between molecules) can be described as a sum of pairwise atomic interactions. For some specific interactions, e.g., weak non-covalent interactions between two different bonds, a pairwise description is not a good approximation; these are represented as the ‘crossterm’ interactions in the formula. For protein simulations these terms are, usually, negligible and therefore not included in the most commonly used force fields to save computational resources.

3.2 Interactions

The most commonly used energy functions for the bonded interactions are bond, angle, and dihedral; for the non-bonded these are Coulomb and Van der Waals. Bond and angle interactions are shown in Figure 14.3a and b. Proper dihedrals correspond to rotatable bonds, for the proteins backbone these are the N-C_α and C_α-C=O bonds, as well as the non-aromatic ones in the side chains, Figure 14.3c. Improper dihedrals are used to constrain a fixed geometry, as for example the peptide bond or the tetrahedral arrangements of the hydrogens in an -NH₃ group, see Figure 14.3d. For bonds, angles and improper dihedrals, typically quadratic (harmonic) functions are used:

$$\text{Bonds: } U(r) = \frac{1}{2}k_b(r - r_0)^2 \quad (6)$$

$$\text{Angles: } U(\theta) = \frac{1}{2}k_\theta(\theta - \theta_0)^2 \quad (7)$$

$$\text{Improper dihedrals: } U(\chi) = \frac{1}{2}k_\chi(\chi - \chi_0)^2 \quad (8)$$

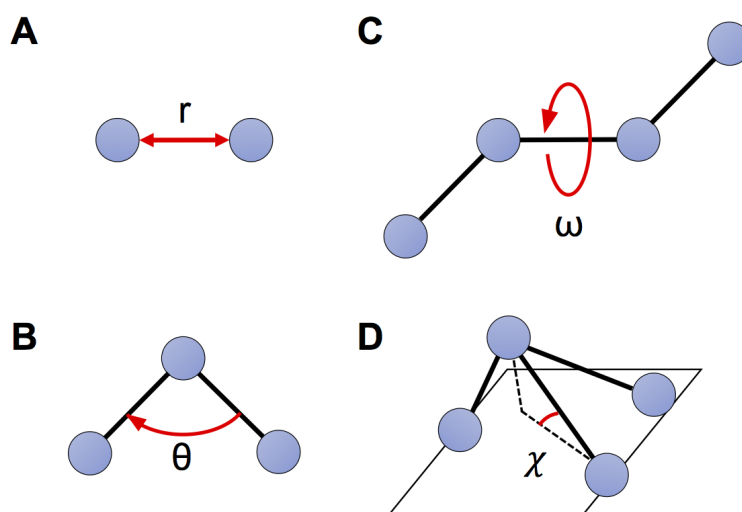


Figure 14.3: Schematic of common bonded interactions: **A** bond between two atoms, **B** angle between three atoms, **C** dihedral angle between four atoms (rotatable bond between the middle two), **D** improper dihedral, used to fix particular orientation, e.g. in-plane, or as in the drawing one atom out of the plane of three other atoms.

For these interactions, r denotes distance, and θ (theta), and χ (chi) denote angles. The subscript 0 refers to the equilibrium value in each case. The k 's are force constants. These parameters depend on the atoms involved.

(Proper) dihedrals are expressed as (periodic) cosines:

$$\text{Dihedrals: } U(\omega) = \frac{1}{2} \sum k_j [1 + (-1)^{j+1} \cos(j\omega + \phi)] \quad (9)$$

Here, ω (omega) denotes the angle. Note that angle (Equation 7) and dihedral (Equations 8 and 9) bonded interactions are not strictly speaking pairwise as they involve three and four atoms, respectively. They are the main non-pairwise contribution that is usually included for protein simulations.

Non-bonded interactions are modelled according to Coulomb's law for electrostatic forces, and the Lennard-Jones potential describing Van der Waals forces, respectively as:

$$\text{Coulomb: } U(r_{ij}) = \frac{\epsilon_0(q_i \cdot q_j)}{r_{ij}} \quad (10)$$

$$\text{Lennard-Jones: } U(r_{ij}) = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right] \quad (11)$$

For the non-bonded interactions, $r_{i,j}$ denotes the distance between atoms i and j , and q_i and q_j the charges on atoms i and j . In Equation 10 ϵ_0 (epsilon)

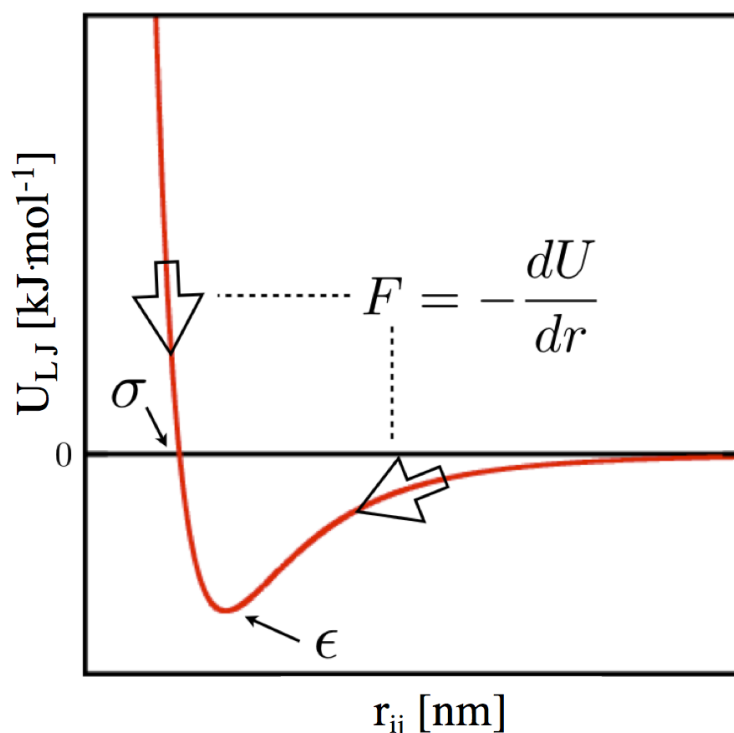


Figure 14.4: The Lennard-Jones potential U_{LJ} as function of r_{ij} . The force F is the derivative of U with respect to r , or in other words the slope of the function U in this plot (indicated by the two arrows).

indicates Coulomb's constant and in Equation 11 σ (sigma) and ϵ indicate the Van der Waals radius and the strength of the interaction, respectively. These parameters depend on the atoms involved. The Lennard-Jones potential is a combination of repulsive interactions originating from the Pauli exclusion principle, and attractive interactions resulting from London dispersion forces. Figure 14.4 shows an example of a Lennard-Jones interaction potential. Generally speaking, an energy is associated with the forces of the system: the slope of the energy determines the direction in which the atom is pushed as shown in Figure 14.4.

3.3 Parameters

The parameters included in force fields are usually derived from experimental techniques and (expensive) quantum chemical calculations (Leach, 2001). For example, bond lengths and angles, i.e. r_0 , θ_0 , χ_0 , as introduced in the previous section “Interactions”, are based on distances in small molecule crystal structures, while force constants are based on a combination of infrared spectroscopy (which measures molecular vibrations) and quantum

chemical calculations. Existing force fields are continuously validated and improved to properly describe the nature of a given system. Many modern force fields like GROMOS (van Gunsteren *et al.*, 1996; Oostenbrink *et al.*, 2004), AMBER (Case *et al.*, 2014), CHARMM (Brooks *et al.*, 1983) and OPLS (Kaminski *et al.*, 2001) are extensively validated on thermodynamic parameters like partition coefficients (affinity for amino acids side chains for either water or oil phase), and peptide and protein folding equilibria (Daura *et al.*, 1999; van Gunsteren *et al.*, 2001; Oostenbrink *et al.*, 2004; Case *et al.*, 2014; Brooks *et al.*, 1983; Shaw *et al.*, 2010; Lindorff-Larsen *et al.*, 2012). This makes these force fields ready to simulate biological phenomena, since we are usually interested in the equilibrium states and transition events which are thermodynamic quantities. See Chapter “Thermodynamics of Protein Folding” for more on thermodynamics.

Limitations of Newtonian physics and force fields

MD simulations take a classical mechanical approach by using Newton’s second law of motion to determine the collective dynamics of a set of particles, such as the atoms in a protein and the surrounding solvent (typically water); this is known as the Molecular Mechanics (MM) approach. Most simulations are performed without considering quantum effects explicitly, such as pi-pi interactions between aromatic rings (e.g. Phenylalanine), electron transfer, proton transfer, bond breakage, and even H-bonds in certain conditions.

For some chemical processes such as enzyme catalysis, proton transfer, and pH effects (typically involving hydrogen atoms) simulations using more advanced models that incorporate quantum physics may be required. Unfortunately, when compared to a classical mechanical approach, quantum mechanical (QM) methods require much longer computation times to investigate the conformational space and energy landscape for large systems such as proteins and polymers, which typically consist of many tens of thousands of atoms. QM methods are currently limited to hundreds of atoms at the very best. Hybrid QM/MM approaches may also be employed, but these also remain strongly limited by the expense on the QM part.

Fortunately, many biological effects that we may be interested in are macroscopic properties, such as protein folding or ligand binding, for which classical or Newtonian physics suffices. Moreover, for such macroscopic properties, our models can be, and typically also have been, validated with experiments.(van Gunsteren *et al.*, 2006)

However, you have to keep in mind that force fields cannot automatically adapt to new conditions, which means that they will work

only for atoms and structures for which parameters have been appropriately determined. If your system includes anything not described in the force field, e.g., a ligand or metal ions, this will typically make the software crash. Therefore, you need to parametrize your ligands and metal coordination centers before launching such a simulation. This is not trivial although several approaches and web tools are available to generate reasonably good topologies for small organic molecules. One of the major bottlenecks is that classical force fields describe electrostatic interactions in physical systems as static atom-centered charges that do not change during your simulation. In other words, the polarization that a real physical system undergoes in a dielectric medium such as water is entirely neglected. To address this problem, polarizable force fields that aim to describe the variations in charge distribution within the dielectric environment are very promising for future applications (Halgren and Damm, 2001). However, these force fields are not easy to set up and therefore currently not available for large systems.

If you need to include metal ions (for example in metalloproteins), things may even get more complicated. Here, transition metals such as copper, zinc and iron are particularly challenging as they exhibit several oxidation states. Often, a subtle equilibrium between two co-existing redox states will be present in a biological systems. Until now, even when using hybrid methods such as QM/MM, it is not possible to get sufficiently realistic parameters for metalloprotein simulations. However, there exist several approaches that approximate these systems as best as possible. For the interested reader, the review by Li and Merz (2017) provides a detailed overview of the advances, state-of-the-art and bottlenecks of modelling metal ions in classical dynamics.

Now, because of the assumption of additive pairwise interactions, you can combine everything (Equations 6-11) by simply adding all energy terms, as was already suggested by Equation 5, which leads to the following result:

$$\begin{aligned}
 U(r) = & \frac{1}{2} \sum_b k_b (r - r_0)^2 + \frac{1}{2} \sum_\theta k_\theta (\theta - \theta_0)^2 + \\
 & \frac{1}{2} \sum_\omega \sum_j k_j [1 + (-1)^{j+1} \cos(j\omega + \phi)] + \\
 & \frac{1}{2} \sum_\chi k_\chi (\chi - \chi_0)^2 + \epsilon \sum_i \sum_j (q_i \cdot q_j) / r_{ij} + \\
 & 4\epsilon \sum_i \sum_j [(\sigma / r_{ij})^{12} - (\sigma / r_{ij})^6]
 \end{aligned} \tag{12}$$

Although not all details of this formula are equally important, you should notice the character of the different summations. The first four are single

Table 1: Overview of important molecular dynamics force fields.

Force fields	Description	Ref.
AMBER	Full-atomistic, used for proteins and DNA	Ponder and Case 2003; Tian <i>et al.</i> 2020
CHARMM	Full-atomistic, used for small molecules and macromolecules	Brooks <i>et al.</i> 1983, 2009
GROMOS	Full-atomistic, bio-molecular systems: solutions of proteins, nucleotides, and sugars	Oostenbrink <i>et al.</i> 2004; Reif <i>et al.</i> 2012
OPLS	Full-atomistic, Optimized Potential for Liquid Simulations	Kaminski <i>et al.</i> 2001; Shivakumar <i>et al.</i> 2012
MARTINI	Coarse grained, molecular dynamics simulations of large bio-molecular systems (~ 4 heavy atoms represented by a single bead)	Marrink <i>et al.</i> 2007; Monticelli <i>et al.</i> 2008; Souza <i>et al.</i> 2021

Note that force fields are constantly improved, so that there are several versions available for each. Therefore, it is recommended to use the latest version of a force field (unless you have a valid reason not to). The choice of the force field itself is not strictly predetermined and may often depend on the system of interest, the program package or the history and experience of the research team. A relatively recent review of force fields can be found in Lindorff-Larsen *et al.* 2012.

sums over all bonds (b), angles (θ) and proper (ω) and improper (χ) dihedrals. Here, as above, we have simply left out cross terms between bonds, angles or dihedrals; this is the ‘other’ class already mentioned which is typically negligible for protein simulations. The last two sums go over all unique pairs of atoms ij for both the Van der Waals (first) and Coulomb (second) interactions. The number of bonds, angles and dihedrals in a system increases approximately linearly with the number of atoms. However, the pairs of atoms is quadratic in the number of atoms. Therefore, the non-bonded interactions are typically the vast majority ($> 95\%$) of the computational effort of any biomolecular simulation (Leach, 2001).

4 Dynamics

Now that we have the forces (and energies) of our system, we can look at the mechanics behind the simulations. We will first look into dynamics in the form of MD simulations. In the next chapter (Chapter “Monte Carlo for Protein Structures”) we will cover Monte Carlo (MC) simulations, which can be seen as a statistical approach to the same problem that MD simulations address.

To describe a dynamic system, the coordinates (positions, r) and mo-

Table 2: Overview of important molecular dynamics software packages.

Software packages	Description	Ref.
AMBER – Assisted Model Building with Energy Refinement	Molecular dynamics simulations of bio-molecules	Case <i>et al.</i> 2014
GROMACS – GRONingen MACHine for Chemical Simulations	Molecular dynamics simulations of proteins, lipids and nucleic acids	Pronk <i>et al.</i> 2013; Abraham <i>et al.</i> 2015
NAMD – NANoscale Molecular Dynamics program	Molecular dynamics simulations of large bio-molecular systems	Phillips <i>et al.</i> 2005, 2020
YASARA – Yet Another Scientific Artificial Reality Application	Molecular dynamics program, including molecular visualization and modeling	Krieger and Vriend 2014, 2015
CP2K – Open Source molecular Dynamics	Atomistic and molecular simulations of solid state, liquid, molecular, and biological systems including QM/MM	Warshel and Levitt 1976

menta (velocities times the mass, $p = m \cdot v$) of all particles are needed. These two sets of variables describe the ‘phase space’. Previously, in Chapter “Thermodynamics of Protein Folding” we have already introduced the *conformational space* which only includes the coordinates (and not the velocities or momenta). In the course of this chapter, we will not go into the subtle difference between these, so for simplicity we will consider the conformational space an adequate description of our molecular system.

In addition to the coordinates and momenta, the energies of the particles are needed. These are split into two parts: the potential and the kinetic energies, U and K respectively. The potential energy derives from the force field description we introduced in the previous section, while the kinetic energy simply derives from the temperature of the system. For example at 0 K (in a frozen system) all velocities are zero, whereas at higher temperature the (average) velocity of all atoms increases.

The final ingredient required to understand the course of MD simulations is the relation between the forces (derivative of the potential energy) and the positions of the atoms, which goes according to Newton’s famous law of motion:

$$F = m \cdot a. \quad (13)$$

Here, F is the force, a the acceleration and m the mass. As we can primarily

calculate the change in velocity, this equation can be rearranged to

$$a = F/m. \quad (14)$$

Actually, Newton wrote his law as a differential equation

$$\frac{\partial a(r)}{\partial r} = \frac{1}{m} \frac{\partial F(r)}{\partial r}, \quad (15)$$

with r the position, F and a as before, and ∂ indicating the (partial) first derivative. Remember that the differential of the energies is the force, so for dynamics we require a differentiable function for the energy.

Now, solving Newton's equation of motion yields a description of the motion of the particles (atoms) involved, which is what we want when we perform an MD simulation. However, this differential equation can only be solved analytically for a system of two particles. Therefore, in a typical MD simulation dealing with large number of particles, a numerical approximation is required to obtain the changes in velocity and position over time. The analytical solution to Newton's equation has nice properties: it is energy-conserving, reversible and deterministic. The numerical approximation, however, may not be so well behaved. Moreover, for a system of more than two particles, we know that this behaves intrinsically non-deterministic or 'chaotic'; meaning that even the tiniest differences in the starting situation will eventually diverge into vastly different behaviour of the simulation. In practice however, for our biomolecular simulations, this is not a problem as average and/or aggregate behaviour should be independent of the details of the integration scheme (as long as the errors are not systematic).

4.1 Integrating equations of motion

The Verlet integration scheme

The most used numerical scheme to integrate the equations of motion in MD simulations is the 'Verlet integration' scheme, which is based on a Taylor expansion of Newton's differential equation (Verlet, 1967). Using a Taylor expansion is one way of writing a differential equation, which uses infinitesimally small increments, as a numerical approximation using finite size time step increments (See Panel "Derivation of the Verlet integration scheme" for more details). Note that there exist a variety of additional molecular dynamics integrators that can be used. However, a detailed discussion of the differences between the methods, their benefits and limitations goes beyond the scope of this chapter and further details may easily be found elsewhere (see for instance Leach (2001) or Frenkel and Smit (2002)).

Choosing timesteps

The time-step Δt needs to be chosen carefully. If a too small timestep is used, precious computational effort is wasted. A too large timestep on the

```

1 def molecular_dynamics(r, num_steps, N, V, T):
2     # generate initial velocities ,
3     # depending on set Temperature
4     v = initial_velocities(T)
5     for i in range(num_steps):
6         # compute forces , from current coordinates r;
7         # depends on (globally defined) force field
8         F = calc_forces(r)
9         # calculate thermostat scaling factor for set
10        # temperature (depends on velocities)
11        λ = temp_scaling(v)
12        # calculate barostat scaling factor for set
13        # pressure (depends on coordinates and forces)
14        μ = pressure_scaling(r, F)
15        # using calc'd forces F, update velocities v
16        v' = λ*(v + F/m*Δt)
17        # using calculated velocities v,
18        # update coordinates r
19        r' = r + v'·Δt
20        # apply constraints – certain atomic distances
21        # and angles that are not allowed to change,
22        # like for hydrogen atoms
23        r'' = apply_constraints(r', r)
24        # correct v for constraints
25        v = (r'' - r)/Δt
26        # apply barostat and obtain final coordinates
27        r = μ*r''
28        # every set number of steps , perform output
29        # of system variables , r, v, or F
30        if (i%nst_log) do_print_log(r, v)
31        if (i%nst_r) do_output_r(r)
32        if (i%nst_v) do_output_v(v)
33        if (i%nst_F) do_output_F(F)
34    # end for loop
35 # end Molecular Dynamics

```

Figure 14.5: Molecular Dynamics algorithm for molecular simulations in pseudo code Python style.

Derivation of the Verlet integration scheme

The trick to deriving the Verlet integration scheme is to take the Taylor expansion of Newton's equation of motion for the forward ($t + \Delta t$) and backward ($t - \Delta t$) step in time. Adding these cancels the third order term (eliminating any approximations there), and yields a formula of positions at the next time point ($t + \Delta t$) as a function of the current (t) and previous ($t - \Delta t$) time points, which contains only zero, second and fourth order terms. If our final equations include only the zeroth and second order terms, we thus make only a fourth order error.

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \mathbf{r}'(t)\Delta t + \frac{1}{2!}\mathbf{r}''(t)\Delta t^2 + \frac{1}{3!}\mathbf{r}'''(t)\Delta t^3 + \dots \quad (16)$$

where the first derivative (the velocity) is $\mathbf{r}'(t) = \mathbf{v}(t)$ and the second derivative (the acceleration) is $\mathbf{r}''(t) = \mathbf{a}(t)$.

$$\begin{array}{r} \mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \mathbf{r}'(t)\Delta t + \frac{1}{2!}\mathbf{r}''(t)\Delta t^2 + \frac{1}{3!}\mathbf{r}'''(t)\Delta t^3 \\ \mathbf{r}(t - \Delta t) = \mathbf{r}(t) - \mathbf{r}'(t)\Delta t + \frac{1}{2!}\mathbf{r}''(t)\Delta t^2 - \frac{1}{3!}\mathbf{r}'''(t)\Delta t^3 \\ \hline \mathbf{r}(t + \Delta t) + \mathbf{r}(t - \Delta t) = 2\mathbf{r}(t) + 2\frac{1}{2!}\mathbf{r}''(t)\Delta t^2 + \end{array} \quad (17)$$

Which, using $2\frac{1}{2!} = 1$ and $\mathbf{r}''(t) = \mathbf{a}(t)$ can then be rearranged to:

$$\mathbf{r}(t + \Delta t) = 2\mathbf{r}(t) - \mathbf{r}(t - \Delta t) + \mathbf{a}(t)\Delta t^2 \quad (18)$$

One can further rewrite this using the velocity $\mathbf{v}(t - \frac{1}{2}\Delta t) = \mathbf{r}(t) - \mathbf{r}(t - \Delta t)$ into:

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \mathbf{v}(t - \frac{1}{2}\Delta t) + \mathbf{a}(t)\Delta t^2 \quad (19)$$

other hand will yield integration errors and prevent the proper dynamics of the system, as shown in Figure 14.6. Strictly, 10 to 20 integration steps are needed per period of a harmonic vibration to integrate accurately. However, due to Verlet's third-order accuracy, 5 integration steps suffice in practice. Typically, the fastest vibrations in biomolecular simulations, are the hydrogen atoms that vibrate at 10 fs (Feenstra *et al.*, 1999). Therefore, 1 or 2 fs timesteps are typically used for MD simulations.

To put these time scales in perspective, the 2 fs timestep is at the bottom of the scale in Figure 14.2. Biologically relevant behaviour starts at the right-hand block; around milliseconds. That means, we need an order of 10^{12} integration steps to start getting into biological timescales. This is typically not feasible, however, fortunately there is still a lot of interesting and

Temperature and pressure

The integration schemes used in MD simulations conserve energy, and the simulations are performed in a box of fixed dimensions. Thermodynamically speaking, these are thus in the NVE ensemble, where the number of particles (N , atoms), volume (V) and energy (E) are constant. As noted in Chapter “Thermodynamics of Protein Folding”, most biological experiments, however, are done in the so-called ‘grand canonical ensemble’ where chemical potential (μ), Pressure and Temperature are constant. This is very hard to achieve in general in a simulation (Pool *et al.*, 2012), so as the next best option, a typical MD simulation is set up in the NPT ensemble where, as before, the number of particles is constant, and pressure (P) and temperature (T) are kept constant on average over some relatively short time period. There are simple solutions for maintaining constant temperature and pressure, called ‘weak coupling’, that have been used for many decades (Berendsen *et al.*, 1984), but currently more recent methods are generally used as they avoid some serious problems in long time-scale simulations (Cheng and Merz, 1996; Mor *et al.*, 2008; Lingenheil *et al.*, 2008).

biologically relevant behaviour that we may observe at achievable timescales of nanoseconds to microseconds.

Biological systems: water needed

When we simulate a protein, the largest part of the investigated system consists of just water. This water is absolutely crucial to the proper behaviour of the protein, as was already introduced in Chapter “Introduction to Protein Structure” Section 3.1, but not something one is actually interested in by itself. There are two different ways to treat water in a simulation: using either an implicit or an explicit water model. In an explicit water model, the water is modelled as discrete H_2O molecules, with a triangular arrangement of the oxygen and two hydrogen atoms. In an implicit model, the water surrounding the protein is described using an average ‘field’ description, which means that water molecules are not treated as actual molecules consisting of particles, but are instead approximated by a dielectric constant which effectively dampens the electrostatic interactions. In the case of explicit water, for two protein molecules of about 3.5 nm diameter, we need a box of 9.3 nm by 6.9 nm by 5.6 nm filled with 10865

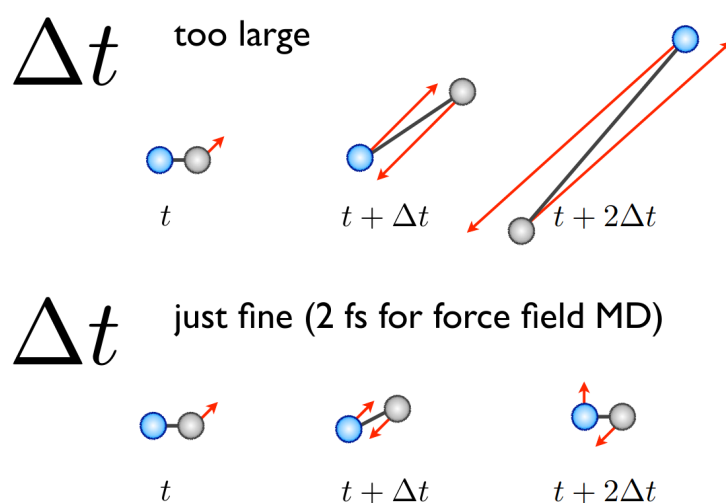


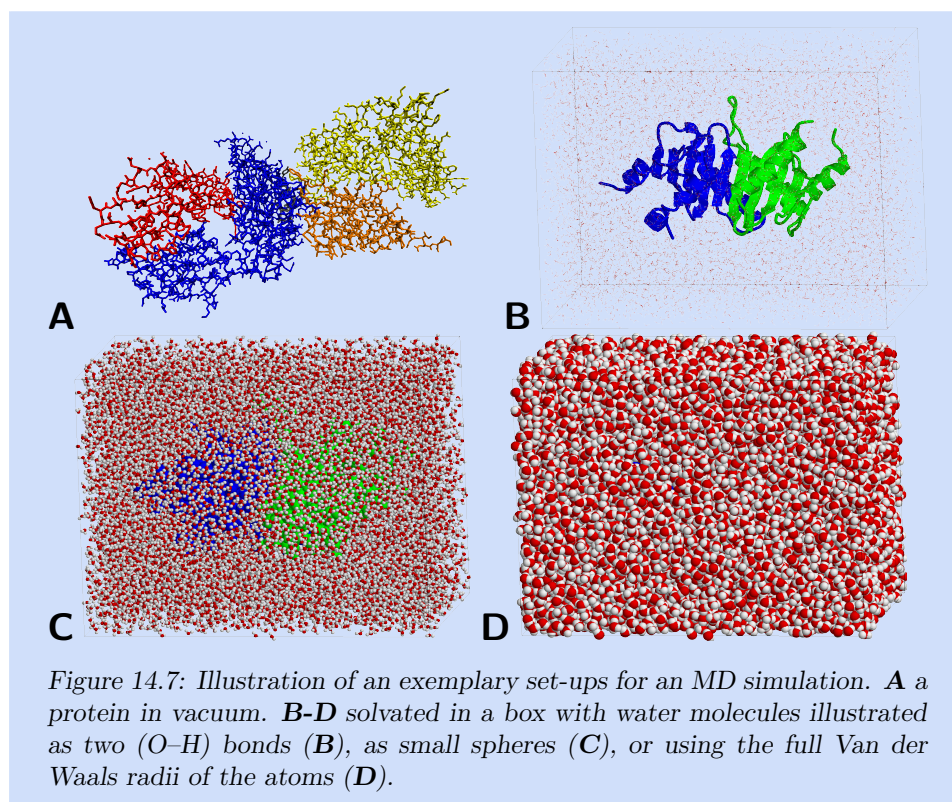
Figure 14.6: Effect of different time steps on the evolution of two particles (atoms) in a molecular dynamics simulation. The forces acting on the atoms are depicted with the red arrows. Note that if the time step is too large (> 2 fs for full-atomistic MD simulations) the forces increase gradually and the positions of the particles are swapped at each step, preventing a physically accurate description of the dynamics.

water molecules. Figure 14.7A shows the two proteins in vacuum, B shows the proteins with the water molecules drawn as lines. In C the water atoms are shown as small spheres where we can still see the protein somewhere in the middle. And in D the full Van der Waals radii of the atoms are drawn which completely hides the protein, thus emphasizing the amount of water needed relative to the size of the protein molecule. This example illustrates why explicit water simulations are much slower and thus computationally more expensive than simulations using an implicit water model. However, depending on the research question and on the experimental counterpart used to validate the simulations, it may be justified to use an implicit model.

Common approaches for performance enhancement

Figure 14.2 highlights the enormous gap between the integration timestep needed, and biologically relevant timescales that we would like to approach in our simulations. So, many tricks have been invented over the years to improve efficiency of the calculations, as well as a wide variety of approximations that enhance performance typically by reducing complexity of the simulated system.

- Trivially: The forces derived from pairwise potentials are related



as follows $F_{ij} = -F_{ji}$, which means the interaction between an atom pair needs to be computed only once.

- The non-bonded potentials (Coulomb: $\sim 1/r$; Lennard-Jones: $\sim 1/r^6$) tend to zero at large distances, so we can use a distance cut-off beyond which interactions are not calculated.
 - Atoms only move a little in one step, so we use a pair-list to keep track of atoms within the cut-off distance which we only re-evaluate every so many steps (calculating distances is relatively expensive because there are quadratically many pairs for the number of atoms).
 - Evaluating r is relatively expensive due to the square-root; so compare the square of the cut-off radius with the square of the distance.
- Large distances change less, which forms the basis for twin-range and multiple time-step methods.
- Many Processor/Compiler/Language specific optimizations:
 - use of Fortran vs. C (and even assembly code) in performance critical parts of the code

- optimize cache performance for very large arrays (positions, velocities, forces, parameters)
- compiler optimizations
- efficient use of multi-core systems
- use of GPU nodes for efficient execution of so-called ‘vector’ and grid operations
- Maximum Time step is limited by vibrational frequencies (see also Figure 14.2):
 - The carbon–hydrogen bond angle vibration is $10^{-14}\text{s} = 10\text{ fs}$. When using 10-20 integration steps per vibrational period, we would need a time step of 0.5 fs, but in practice 1 or 2 fs time steps are used (1.000.000 or 500.000 steps for 1 ns)
 - Bond distances also have high frequencies of vibration, but for protein simulations these are already always constrained.
 - If we also remove other fast vibrations (hydrogen atom bond and angle motion and some out of plane motions of aromatic groups) using constraints we can use a timestep of 2 or 4 fs, and even stretch it to 6 or 7 (Feenstra *et al.*, 1999; Hopkins *et al.*, 2015).
- When using simplified force fields, fewer particles are required, thus reducing the time needed for the force computations.
 - United atoms: aliphatic groups (a carbon bound to only other carbons and hydrogen atoms) are uncharged and nearly spherical, because the hydrogens are so much smaller than the carbon atoms. These can be treated as single particles with a radius slightly larger than just the carbon atom and a larger mass (van Gunsteren *et al.*, 1996; Yang *et al.*, 2006).
 - Coarse-graining: multiple atoms are combined into one particle with so-called effective interaction potentials, resulting in fewer atoms, and also in (much) larger timesteps (May *et al.*, 2014; Singh and Li, 2019).
 - Implicit water: water may also be described not as a collection of molecules, but instead the water molecules are described as an average ‘field’ (Roux and Simonson, 1999; Onufriev and Case, 2019).

In this list we mention only a few strategies to reduce the simulation time. More comprehensive and technical treatises may be found in Hess *et al.* (2008) and Pronk *et al.* (2013), both focused on the GROMACS MD simulation software, and Bowers *et al.* (2006), Dror *et al.* (2012) and Grossman *et al.* (2013) from the D. E. Shaw research group who developed their own simulation software but also their own

supercomputer based on specialized MD hardware called ANTON.

4.2 Convergence of state properties

Viewing the movie resulting from the MD simulation is not an efficient way to analyse simulations, and many details are not registered by the human eye. Moreover, the particular order of events, or even a particular conformation, observed in a simulation, is not of interest in itself. When analysing multiple simulations of the same system, one will observe that the order is not conserved, and many small variations on conformational states exist, due to the chaotic nature of the system, as described in the section on dynamics.

Thus, in this part, we will introduce how to quantify simulation results. An important overall feature of molecular simulations is some measure of convergence. Strictly speaking, convergence indicates how well the sampled conformations represent the system at equilibrium. It can also be seen as a measure of the exploration of the energy landscape (or conformational space) of the system of interest. Moreover, not all parts of the energy landscape are equally relevant to the behavior of the simulated system; (very) high energy parts are (very) unlikely to be visited, in reality and as well as in the simulation, as we saw in the previous Chapter “Thermodynamics of Protein Folding”. The parts that do not have (very) high energies are often called the ‘reachable’ conformational space.

So, sampling can be considered as the extent to which this reachable part of conformational space has been visited during the simulation. Finally, complete sampling then corresponds to (complete) convergence. However, you should always keep in mind that the dynamics of proteins are stochastic. When starting a simulation from a given structure you might get stuck in a local minimum, in which case, you will not be able to explore all the relevant regions of your energy landscape. Now, we will show some examples of how to (not) recognize convergence from the results of a simulation.

Enhanced Sampling techniques

When launching a molecular dynamics simulation from a given starting point (like a protein crystal structure), there is the danger of getting stuck in a local minimum during the simulations. In that case, you won’t be able to sample sufficient parts of the conformational space of your system to study what is going on. To avoid this multiple minima problem, several advanced sampling techniques can be used (Yang *et al.*, 2019). Here, we want to briefly introduce the concepts of umbrella sampling and replica exchange molecular dynamics (see Panel “Replica exchange molecular dynamics (REMD)”); we will also come back to these in Chapter “Monte Carlo for Protein Structures”.

Umbrella sampling requires you to have some prior information of what is going on in your system (Kästner, 2011). This is different from REMD which allows you to freely explore the conformational space. In umbrella sampling, you need to pre-define a specific order parameter (a collective variable) that will properly describe relevant changes in your system. What is a good order parameter depends on your research questions; it may for instance be the distance between a ligand and its protein binding site, or between opposite sides of an opening and closing channel protein, or the end-to-end distance in an unfolding event. Once this is clearly defined, your system will be constrained using harmonic potentials (umbrellas) all along this order parameter to sample the parts of space you are interested in. Similar to REMD, this technique is computationally expensive. However, it does allow you to get a quantitative estimation of the free energy for a given event.

Replica exchange molecular dynamics (REMD)

Replica exchange molecular dynamics (REMD) allows you to sample large portions of your configurational space by running several identical copies (replicas) of your system (e.g., the protein and water) simultaneously at different temperatures (e.g., between 300 K and 700 K). While the higher temperature replicas are provided with sufficient energy to jump over high energy barriers, the low temperature replicas are able to properly sample the local minima. Over time, neighboring replica are allowed to swap their positions which lets you sample different portions of the surface. The mayor bottleneck of this techniques is is the computational costs due to the fact that all the replica need to be simulated in parallel.

Observing sampling – RMSD

Root mean square deviation (RMSD, introduced in Chapter “Structure Alignment”) is one way to chart the progress in sampling the conformational space, by effectively measuring the distance from the starting point. Figure 14.8 shows the RMSD of the conformation of a protein with respect to the starting (crystal) structure during the simulation. It is clear that the RMSD increases over time, as expected (because the atoms move, the structure changes). Importantly, the three panels show this simulation at three different time scales: 0-100 ps (top left), 0-1 ns (top right) and 0-5.5 ns (bottom right). At each of these timescales, the RMSD appears to reach a plateau roughly half-way through the plot. But the plots are all based on the same simulation, and on the next plot with a longer timescale one sees the increase in RMSD continuing. In other words, each time the plateau we see is ‘temporary’, and the impression it gives of reaching a converged

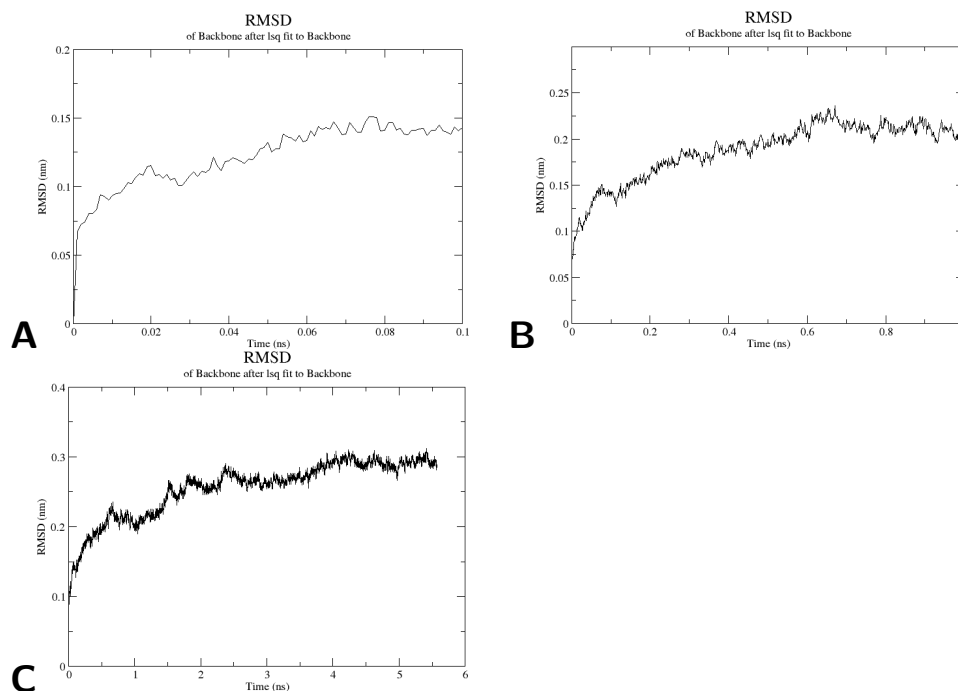


Figure 14.8: Apparent convergence in root-mean-square-deviation (RMSD) may be observed in protein simulations at different timescales. The three plots each derive from the same simulation. At the shortest timescale, up to 0.1 ns or 100 ps **A**, the RMSD appears to reach a plateau, suggesting convergence. However, extending the simulation to 1 ns **B**, and to 5.5 ns **C** shows this convergence is transient.

value (in this case for the RMSD) is false. We can be sure this holds for the plateau observed in the bottom right figure as well. If we would run this simulation for much longer, say 100 ns, we would probably still see similar behaviour. These are all symptoms of diffusive motion in a high-dimensional space; as the distance from the starting point increases, the sampling spreads out more and more at a given RMSD distance from the starting point. This is similar to diffusion of molecules in a solution where the distance reached from the starting point increases with the square root of the time elapsed – but the effect is much stronger in high-dimensional conformational space, as we would take a higher power root of the elapsed time here. Therefore, the change in distance gets slower, but the actual changes in conformation still occur as much as in the beginning.

When sampling of the system reaches timescales at which these conformational transitions start to equilibrate, we should see a change in this behavior. We expect the distances to become smaller again after some time. In fact, Figure 14.9 shows that for a small peptide this is achieved at timescales of a few hundred ns. This seven-residue peptide was studied at the ETH in Zürich in the late 1990's. The NMR predicted an α -helical conformation

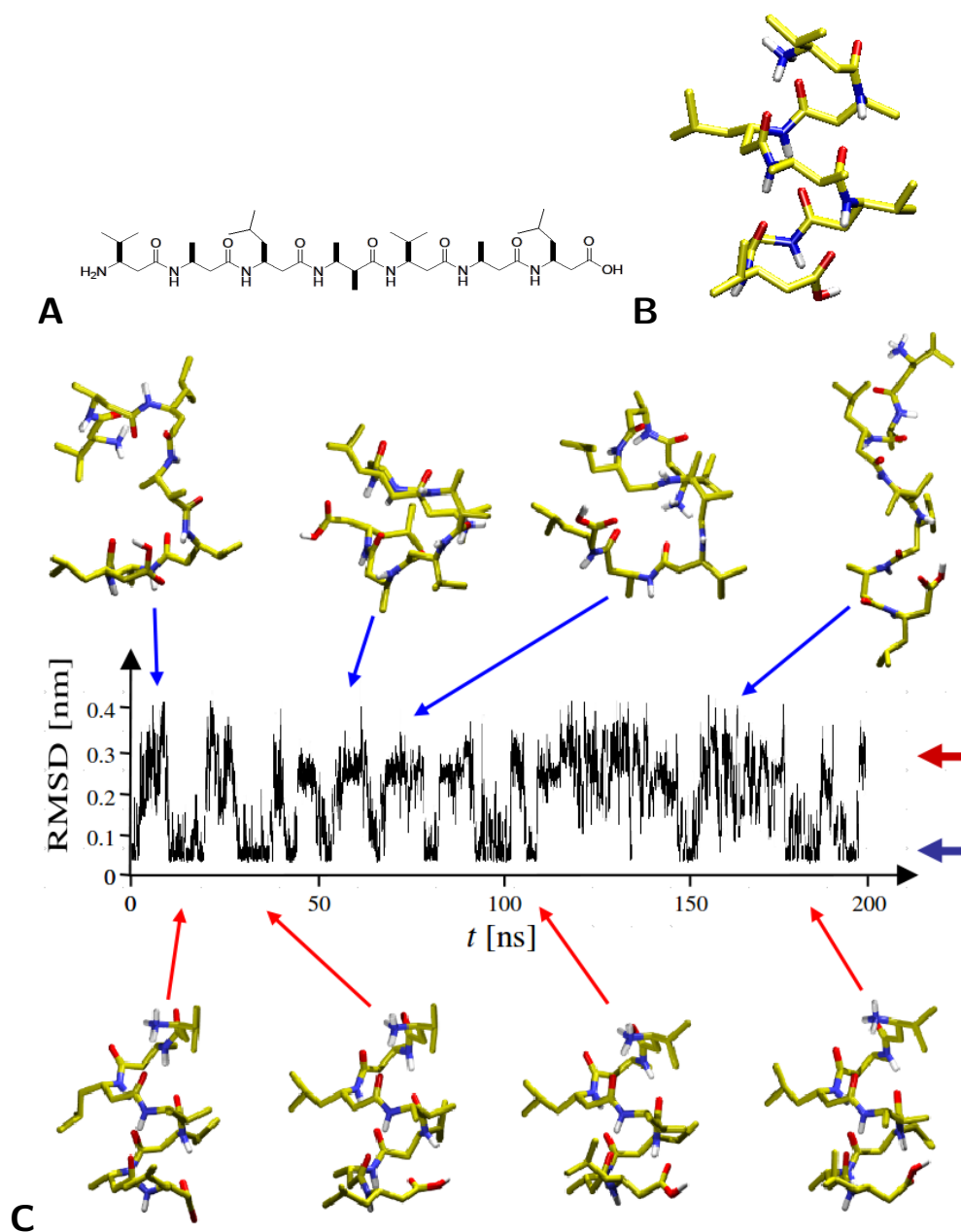


Figure 14.9: **A** Small beta-peptide of seven residues (beta-peptides have an additional carbon atom in the backbone compared to normal (alpha) peptides). **B** The peptide forms a helical structure according to NMR. **C** Simulations show a very dynamical behaviour, where the RMSD to the helical structure increases, but also decreases again repeatedly. This indicates the peptide unfolding (increasing RMSD) and (re-)folding; importantly this is the very first reversible folding simulation ever (Daura et al., 1999; van Gunsteren et al., 2001). Reproduced with permission from Daura & Oostenbrink (pers. comm.).

for the peptide, shown in Figure 14.9b, but the side chains could not be resolved due to lack of data, and some violations of the distances obtained from the experiment.

Very extensive simulations, for those times, of this peptide were performed in the group of Wilfred van Gunsteren (also ETH Zürich) (Daura *et al.*, 1999); note that in those days this was still a major investment of computational power, requiring almost half a year on a large compute cluster. The simulation shows the peptide going through various different conformations, some of them helical (or helix-like), others decidedly different, as can be seen in Figure 14.9c. The RMSD with respect to the helical (NMR) structure shows that this simulation, for the first time in history, reached equilibrium behaviour. The RMSD not only went up, but also went down again. In other words, during this simulation the system visits different states, but also returns to states visited before. It is important to realize that, in this plot, a low RMSD always means the same state (the native, folded helical state, as can be seen in the snapshots below the plot), but a high RMSD can mean many different things. The snapshots above the plot show that these states are all clearly not helical, but they are also not similar to each other. Clustering of these unfolded structure showed that there are (only) about 1000 distinct conformational states in this unfolded ensemble (for details, please refer to van Gunsteren *et al.* 2001).

Ten years later in 2010, Shaw *et al.* showed that now we can obtain reversible folding also for a small (70 residue) protein. The timescale needed is much larger than that for the peptide; here we see the simulation goes up to 200 μs (note, that this is a similarly major investment of computational power as the peptide was a decade earlier, of the order of many months in parallel on a large number of highly specialized CPU's).

Evaluating your MD simulations – Order Parameters

To be able to track the progress of a molecular dynamics (MD) simulation, several powerful analysis tools have been developed over the years. Although it may be fun and also seemingly intuitive to visualize your results as a movie by loading the MD trajectories into programs such as VMD or Chimera, it is not sufficient to understand what is going on and you need a quantitative way of analysing your results. Luckily, MD packages provide a large number of ready-implemented parameters that can be used to track changes between states of interest in your simulations. First, you should check the general properties of your system, such as the average pressure and the temperature, to make sure everything went well. In a next step, tracking the changes of the following *order parameters* may help you pinpoint potential

events during your simulations:

- **Root Mean Square Deviation:** Measure of dissimilarity (distance) between two molecular conformations used to compare the conformation of each frame with respect to, e.g., the starting point of the simulation
- **Root Mean Square Fluctuation:** Measure of structure variation, it calculates the standard deviation of the deviation of an atomic position over time.
- **Radius of gyration:** Measure for the compactness of a structure, powerful to identify unfolding event in your protein structures.
- **Solvent Accessible Surface Area:** Measure to determine the surface of the biomolecules that are accessible to the surrounding solvent (usually water). This is very helpful in protein folding and stability studies.
- **Number of hydrogen bonds:** Compute the numbers of hydrogen bond contacts in your system. Particularly interesting when looking at complexes (ligand-protein, protein-protein) and (un)folding events.
- **Essential Dynamics (ED) analysis** (Amadei *et al.*, 1993; van Aalten *et al.*, 1997): This technique is a principal components analysis (PCA) of the atomic coordinates. It takes the covariance matrix of the coordinates of the atoms. This covariance matrix is diagonalized, yielding eigenvalues and eigenvectors; the eigenvectors describe collective motions of the atoms analyzed. The eigenvalues (or loadings) represent the magnitude of each of these motions. We can now simply focus analysis on the eigenvector(s) with the largest eigenvalues, or several of the largest – these are also referred to as ‘Essential Modes’. Finally, if we choose only the two largest, we can project motions onto a two-dimensional plot for visualization.

For all these measures, the parameter of interest does not have to be calculated for the whole system – often for example only the C α -atoms or the active site or binding region are analyzed.

Depending on the problem at hand, looking at additional order parameters may be interesting. For example to analyse the simulation of a ligand-protein complex, the contact frequency of the ligand with specific amino acid inside the binding site or the distance between the centers of mass. Finally, despite the large number of available tools that come with most simulation packages, in some cases, your simulations may require you to do some self-coding to be able to follow important changes during your simulations. A Python library to analyse your simulations that may come handy can be found

here: <https://www.mdanalysis.org/> (Michaud-Agrawal *et al.*, 2011; Gowers *et al.*, 2016).

4.3 Temperature dependence

In the previous Chapter “Thermodynamics of Protein Folding”, Figure 13.4a,b, we already saw how the folding-unfolding equilibrium that we are sampling in these simulations for this peptide, depends on the temperature of the simulation. The middle plot (at 340 K) is the one also shown in Figure 14.9c, where on average the peptide is folded 50% of the time (and unfolded the other 50%). At lower temperatures in Figure 14.10a, we see the peptide tends to spend a larger fraction of time in the folded state. At higher temperatures we see the opposite; a larger fraction of time spent in the unfolded state(s). Note that the temperature changes are fairly small, only 10-20K up, and 20-40K down, and it is reassuring that simulations are able to capture these relatively subtle effects. Figure 14.10b shows that also high pressure has the effect of unfolding. Both effects of high temperature and high pressure unfolding of proteins are also experimentally well known facts, for by far the majority of proteins.

Simulated annealing

One way to overcome free energy barriers and escape from free energy minima in a molecular system is to increase the temperature. At room temperature a protein system can be trapped in a free energy minimum from which it can escape only after a very long time, as we already discussed previously. For MD simulations, a microsecond is already considered a long time. When performing a simulation at higher temperature, 400 K for example, the system can explore many more different configurations than at room temperature. However, such configurations do not necessarily represent configurations of the system at room temperature, as then they might be very unlikely. To overcome this, one can take a number of snapshots from the high temperature simulation, and use these as a starting point for MD simulations at room temperature. The snapshots will then equilibrate to configurations that are likely to occur at room temperature. This approach is known as simulated annealing and is used in for example resolving NMR structures by letting high temperature conformations anneal to the constraints obtained from various NMR spectra.

One problem with the simulated annealing approach is that the resulting conformations do not represent the equilibrium distribution. The selection of the snapshots chosen to equilibrate to room temperature does not follow the Boltzmann distribution. In principle, the MD simulation at room temperature should sample the equilibrium distribution, but may require a long time to achieve that. To overcome this issue, but still make use of the

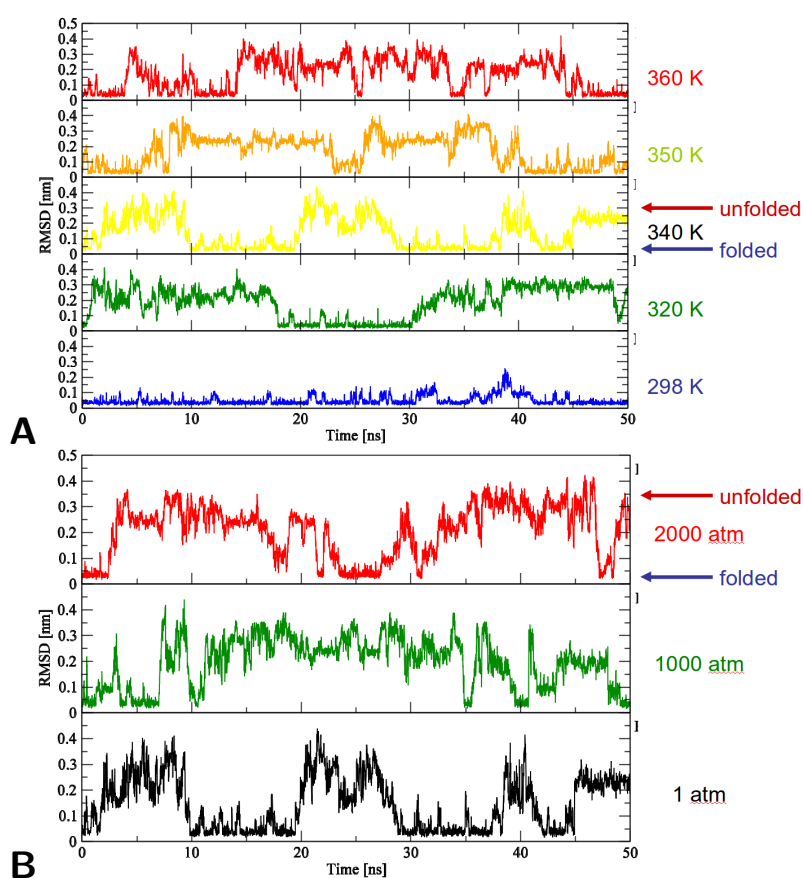


Figure 14.10: **A** The folding equilibrium of the beta-peptide depends on temperature: at lower temperatures a larger fraction of time is spent in the folded (low RMSD) state, at higher temperatures a smaller fraction is folded. **B** Also at higher pressure, the fraction folded decreases (Daura et al., 1999; van Gunsteren et al., 2001). Reproduced with permission from Daura & Oostenbrink (pers. comm.).

enhanced sampling provided by performing MD at high temperature, the procedure to select snapshots for equilibration at lower temperature should follow the Boltzmann distribution. The solution to this is parallel tempering or replica exchange, which we already touched on in Panel “Replica exchange molecular dynamics (REMD)”.

4.4 Homology model optimization

Now we will illustrate simulation, sampling and convergence in practice. Here we have used MD simulations to optimize details of a homology model of an enzyme (Feenstra *et al.*, 2006). Figure 14.11a is an overview picture of the enzyme Styrene mono-oxygenase (SMO) homology model protein structure. If you look closely, you will observe two additional molecules in the

center: the styrene (STY) ligand, and the flavin-adenin-dinucleotide (FAD) co-factor.

The homology model for this structure was built based on two templates of about 23% sequence identity, which is far enough to be a difficult target for homology modelling. Docking of the ligand into the model yielded binding conformations that did not correspond to known products of this enzyme. We therefore used MD simulations to optimize the model, with a main aim to improve the shape of the active site pocket, but to do that we would also need to relax strain due to bad contacts in the initial model. The procedure was therefore aimed to relax this strain, without leading to distortion of the binding site region. We first allowed the water to relax around the protein during a 1-ps MD where the positions of the atoms in the protein were restrained. Subsequently, we released the constraints on parts of the protein, so these were also allowed to relax. First, only the side chains of the residues outside the binding pocket were released, and simulated for 1 ps. Next, also the backbone of these residues were released and simulated for 10 ps. Then, the side chains of the binding residues were released, and simulated for another 10 ps. During this whole procedure the positions of the FAD co-factor and the styrene substrate were also restrained, this allowed the protein structure to relax around the bound co-factor and substrate. Finally, only the backbone atoms of the binding residues were restrained for 100 ps, while the rest of the protein as well as FAD and styrene were free.

To be able to track progress during the optimization simulations for this homology model, we used Essential Dynamics (ED) analysis (Amadei *et al.*, 1993; van Aalten *et al.*, 1997), see also Panel “Evaluating your MD simulations – Order Parameters”. We can now simply focus analysis on the two largest eigenvalues (the essential modes) and project these motions onto a two-dimensional plot for visualization.

To assess stability of the protein, and to see if the optimization procedure was successful in improving stability, we performed two sets of simulations, one started from the non-optimized ‘raw’ homology model, while the second set of simulations was started from the optimized model. We visualized the behaviour in these two sets of simulations using ED analysis on both. Figure 14.11b shows the ED plot representing the motion during these simulations, projected onto the two largest eigenvectors; i.e. the two largest collective motions going on in these simulations. It is clear, that both sets of simulations (‘raw’ vs. ‘optimized’) have different behaviour. The ‘raw’ simulations tend to start out by moving rapidly away from their starting point, and in different directions; this indicates strain in the initial conformation. In contrast the ‘optimized’ simulations move more gradually and more similarly. Finally, it is clear that both sets of simulations have a different starting point: this of course corresponds to the changes in the structure due to the optimization procedure.

The motion of the protein structure corresponding to each of these two

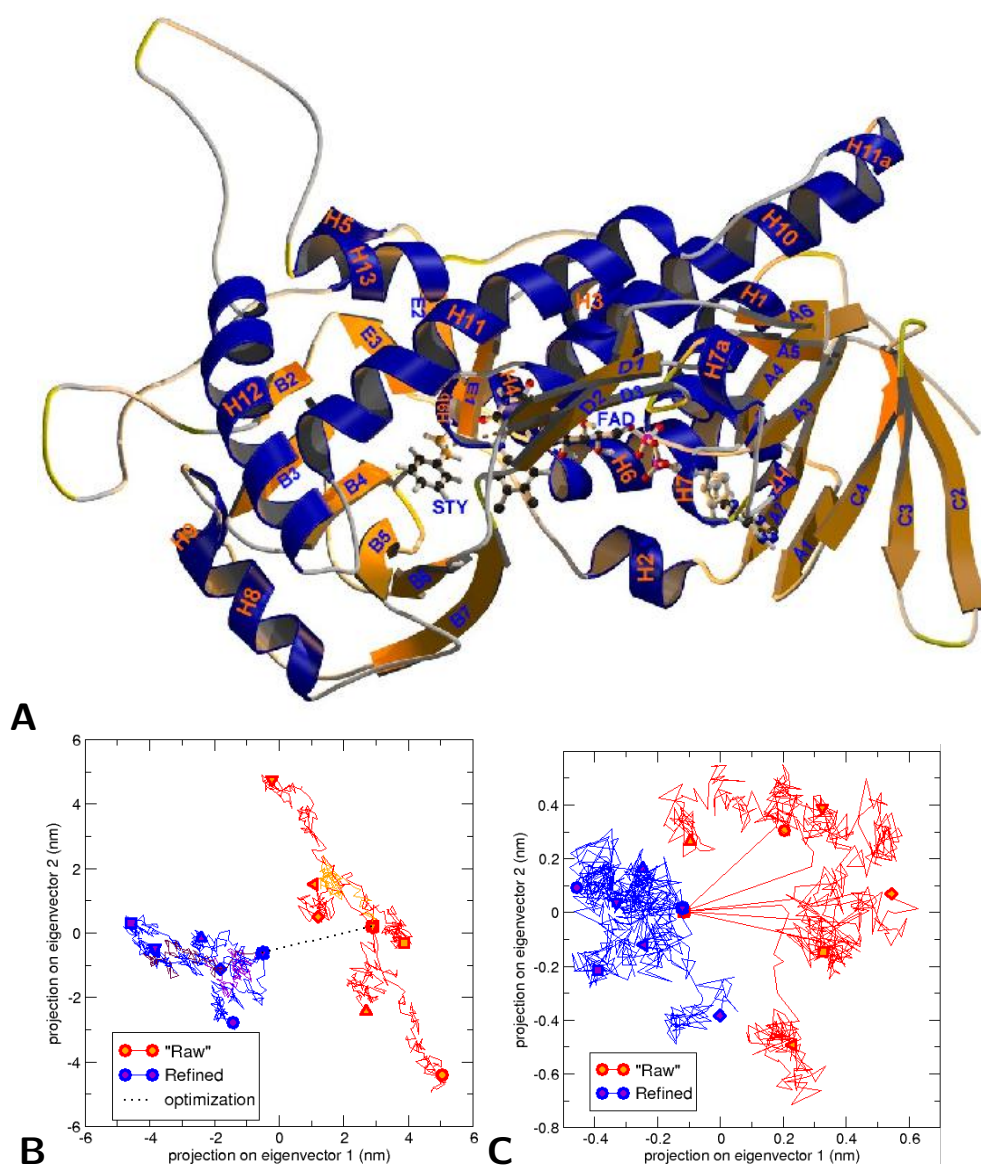


Figure 14.11: A Structure of the Homology Model Styrene Mono-Oxygenase (SMO) Enzyme. **B** Essential Dynamics (ED) analysis of the $C\alpha$ atoms, showing backbone rearrangements during simulations starting from the 'raw' structure (blue), which are distinct from those started from the 'refined' homology model (red). The dotted line indicates the optimization path. **C** The same, but now the ED analysis was performed on the active site region only. Here, structural effects (difference between starting points) are small - the optimization path can not even be seen here. Nevertheless, overall behaviour of the 'refined' simulations is still distinctly different from that of the 'raw' simulations. The long straight lines 'shooting' out from the 'raw' starting point indicate high levels of strain in the 'raw' structure (red), which is relaxed in the refined structure (blue). Figure modified from Feenstra et al. (2006).

eigenvectors can be visualized in a movie. In such a movie, you will be able to see that all atoms contribute to the overall motion, although not all atoms equally. You will also see that the two (eigenvector) motions are distinct; different atoms contribute more to each of them, and the motions are (often) in different directions. Finally, you will also see that different parts, sometimes quite far away in the protein structure, move in and out together in a sort of concerted way.

5 Outlook and summary

Molecular dynamics simulations have shown their usefulness in giving a detailed view on how proteins work; in the words of Shaw *c.s.*, a “Computational Microscope for Molecular Biology” (Dror *et al.*, 2012). With ever increasing computational power, microsecond simulations are within reach and millisecond simulations have already been produced for a few selected systems. Current challenges lie in making available methods accessible, and in handling and analysing the huge amounts of data produced from the simulations. Before starting to run an MD simulation, there are two important things that you need to think about, namely which simulation technique is adequate to solve your scientific question and do you have experimental data to validate your models. As numerous techniques have been developed over the years and are now available to the scientific community, you should make sure to understand the basic concepts, usefulness and limitation of the technique you want to use. Later, it is extremely important to support your results and validate your models to guarantee their physical relevance. Often, you will be able to work in close collaboration with experimental groups or even perform experiments yourself. Alternatively, experimental data can be used from published results in the literature. In any case, make sure to be aware of what is going on in the experiment, the information it is able to provide, its benefits, bottlenecks, errors sources, and limitations.

6 Key concepts

Concepts that should be known after reading this chapter:

- The importance of molecular motions:
 - Proteins are dynamic and understanding their motions is crucial to be able to understand their biological function
 - Always remember and be aware of the relevant length and time scales to observe a biological event
- MD Simulations are based on:
 - Newton’s classical mechanics
 - force fields that contain the description (parametrization) of the bonded and non-bonded interactions

- integrating the equations of motion
- describing thermodynamics ensembles of the system
- Trajectory on Energy Surface
- In protein Dynamics:
 - make sure that you have an adequate experimental counterpart to guide and validate your simulations
 - the level of detail of your forcefield, e.g. atomistic or coarse-grained, should match the problem you are interested in
 - attainable timescales and level of convergence should match the problem you are interested in

7 Further reading

- Understanding Molecular Simulation – Frenkel and Smit (2002)
 - General, in-depth introduction to simulation
- Simulating the Physical World – Berendsen (2007)
 - Technical treatise on physical properties, simulation algorithms and statistical thermodynamics
- Molecular Modelling: Principles and Applications – Leach (2001)
 - Overview of Molecular Modelling and Computational simulations.
- The Art of Molecular Dynamics Simulation – Rapaport (2004)

Author contributions

Wrote the text:	HM, JvG, QH, AM, JV, KAF
Created figures:	HM, ASR, JV, KAF
Review of current literature:	HM, ASR, JV, KAF
Critical proofreading:	HM, JvG, ASR, JV, SA
Non-expert feedback:	QH, AM
Editorial responsibility:	HM, SA, KAF

References

- Abraham, M. J., Murtola, T., Schulz, R., Páll, S., Smith, J. C., Hess, B., and Lindahl, E. (2015). GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, **1-2**.
- Adcock, S. A. and McCammon, J. A. (2006). Molecular dynamics: survey of methods for simulating the activity of proteins. *Chem. Rev.*, **106**(5), 1589–1615.
- Alder, B. J. and Wainwright, T. E. (1957). Phase Transition for a Hard Sphere System. *The Journal of Chemical Physics*, **27**(5), 1208.
- Amadei, A., Linssen, A. B. M., and Berendsen, H. J. C. (1993). Essential Dynamics of Proteins. *PROTEINS: Struct. Funct. Gen.*, **17**, 412–425.
- Berendsen, H. J. C. (2007). *Simulating the Physical World: Hierarchical Modeling from Quantum Mechanics to Fluid Dynamics*. Cambridge Univ. Press, Leiden.
- Berendsen, H. J. C., Postma, J. P. M., van Gunsteren, W. F., DiNola, A., and Haak, J. R. (1984). Molecular dynamics with coupling to an external bath. *J. Chem. Phys.*, **81**(8), 3684.
- Bowers, K. J., Chow, E., Xu, H., Dror, R. O., Eastwood, M. P., Gregersen, B. A., Klepeis, J. L., Kolossvary, I., Moraes, M. A., Sacerdoti, F. D., Salmon, J. K., Shan, Y., and Shaw, D. E. (2006). Scalable Algorithms for Molecular Dynamics Simulations on Commodity Clusters. In *Proceedings of the ACM/IEEE Conference on Supercomputing (SC06)*, New York, NY. IEEE.
- Brooks, B. R., Brucoleri, R. E., Olafson, B. D., States, D. J., Swaminathan, S., and Karplus, M. (1983). CHARMM: a Program for Macromolecular Energy, Minimization, and Dynamics Calculation. *J. Comput. Chem.*, **4**, 187–217.
- Brooks, B. R., Brooks, C. L., Mackerell, A. D., Nilsson, L., Petrella, R. J., Roux, B., Won, Y., Archontis, G., Bartels, C., Boresch, S., Caffisch, A., Caves, L., Cui, Q., Dinner, A. R., Feig, M., Fischer, S., Gao, J., Hodoscek, M., Im, W., Kuczera, K., Lazaridis, T., Ma, J., Ovchinnikov, V., Paci, E., Pastor, R. W., Post, C. B., Pu, J. Z., Schaefer, M., Tidor, B., Venable, R. M., Woodcock, H. L., Wu, X., Yang, W., York, D. M., and Karplus, M. (2009). CHARMM: The biomolecular simulation program. *Journal of Computational Chemistry*, **30**(10).
- Case, D., Babin, V., Berryman, J., Betz, R., Cai, Q., Cerutti, D., Cheatham III, T., Darden, T., Duke, R., Gohlke, H., Goetz, A., Gusarov, S., Homeyer, N., Janowski, P., Kaus, J., Kolossváry, I., Kovalenko, A., Lee, T., LeGrand, S., Luchko, T., Luo, R., Madej, B., Merz, K., Paesani, F., Roe, D., Roitberg, A., Sagui, C., Salomon-Ferrer, R., Seabra, G., Simmerling, C., Smith, W., Swails, J., Walker, R., Wang, J., Wolf, R., Wu, X., and Kollman, P. (2014). AMBER 14. University of California, San Francisco.
- Cheng, A. and Merz, K. M. (1996). Application of the Nosé-Hoover Chain Algorithm to the Study of Protein Dynamics. *The Journal of Physical Chemistry*, **100**(5), 1927–1937.
- Cuendet, M. A., Zoete, V., and Michielin, O. (2011). How T cell receptors interact with peptide-MHCs: a multiple steered molecular dynamics study. *Proteins*, **79**(11), 3007–3024.
- Daura, X., Jaun, B., Seebach, D., van Gunsteren, W. F., and Mark, A. E. (1998). Reversible Peptide Folding in Solution by Molecular Dynamics Simulation. *J. Mol. Biol.*, **280**(5), 925–932.
- Daura, X., Gademann, K., Jaun, B., Seebach, D., van Gunsteren, W. F., and Mark, A. E. (1999). Peptide Folding: When Simulation Meets Experiment. *Angew. Chem. Intl. Ed.*, **38**(1/2), 236–240.
- Dror, R. O., Dirks, R. M., Grossman, J. P., Xu, H., and Shaw, D. E. (2012). Biomolecular Simulation: A Computational Microscope for Molecular Biology. *Annual Review of Biophysics*, **41**(1), 429–452.
- Duan, Y. and Kollman, P. A. (1998). Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution. *Science*, **282**, 740–744.
- Feenstra, K., Hess, B., and Berendsen, H. (1999). Improving efficiency of large time-scale molecular dynamics simulations of hydrogen-rich systems. *Journal of Computational Chemistry*, **20**(8).
- Feenstra, K. A., Hofstetter, K., Bosch, R., Schmid, A., Commandeur, J. N. M., and Vermeulen, N. P. E. (2006). Enantioselective substrate binding in a monooxygenase protein model by molecular dynamics and docking. *Biophysical Journal*, **91**(9), 3206–16.
- Frenkel, D. and Smit, B. (2002). *Understanding Molecular Simulation: From Algorithms to*

- Applications*, volume 1 of *Computational Science Series*. Academic Pr, San Diego, second edition.
- Gowers, R., Linke, M., Barnoud, J., Reddy, T., Melo, M., Seyler, S., Domański, J., Dotson, D., Buchoux, S., Kenney, I., and Beckstein, O. (2016). MDAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations. In *Proceedings of the 15th Python in Science Conference*, number Scipy, pages 98–105.
- Grossman, J. P., Kuskin, J. S., Bank, J. A., Theobald, M., Dror, R. O., Ierardi, D. J., Larson, R. H., Schafer, U. B., Towles, B., Young, C., and Shaw, D. E. (2013). Hardware Support for Fine-grained Event-driven Computation in Anton 2. In *Proceedings of the Eighteenth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '13, pages 549–560, New York, NY, USA. ACM.
- Halgren, T. A. and Damm, W. (2001). Polarizable force fields. *Current Opinion in Structural Biology*, **11**(2).
- Hess, B., Kutzner, C., van der Spoel, D., and Lindahl, E. (2008). Gromacs 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation. *J. Chem. Theory Comput.*, **4**(3), 435–447.
- Hopkins, C. W., Le Grand, S., Walker, R. C., and Roitberg, A. E. (2015). Long-Time-Step Molecular Dynamics through Hydrogen Mass Repartitioning. *Journal of Chemical Theory and Computation*, **11**(4), 1864–1874.
- Kaminski, G. A., Friesner, R. A., Tirado-Rives, J., and Jorgensen, W. L. (2001). Evaluation and Reparametrization of the OPLS-AA Force Field for Proteins via Comparison with Accurate Quantum Chemical Calculations on Peptides. *The Journal of Physical Chemistry B*, **105**(28), 6474–6487.
- Karplus, M. and Kuriyan, J. (2005). Molecular dynamics and protein function. *Proceedings of the National Academy of Sciences*, **102**(19), 6679–6685.
- Kästner, J. (2011). Umbrella sampling. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, **1**(6), 932–942.
- Krieger, E. and Vriend, G. (2014). YASARA View—molecular graphics for all devices—from smartphones to workstations. *Bioinformatics*, **30**(20), 2981–2982.
- Krieger, E. and Vriend, G. (2015). New ways to boost molecular dynamics simulations. *Journal of Computational Chemistry*, **36**(13).
- Kumar, S. and Li, M. S. (2010). Biomolecules under mechanical force. *Phys. Rep.*, **486**, 1–74.
- Leach, A. (2001). *Molecular Modelling: Principles and Applications*. Pearson.
- Lemkul, J. A. and Bevan, D. R. (2013). Aggregation of Alzheimer’s amyloid β -peptide in biological membranes: a molecular dynamics study. *Biochemistry*, **52**(29), 4971–4980.
- Li, P. and Merz, K. M. (2017). Metal Ion Modeling Using Classical Mechanics. *Chemical Reviews*, **117**(3), 1564–1686.
- Lindorff-Larsen, K., Maragakis, P., Piana, S., Eastwood, M. P., Dror, R. O., and Shaw, D. E. (2012). Systematic validation of protein force fields against experimental data. *PLoS ONE*, **7**(2), e32131.
- Lingenheil, M., Denschlag, R., Reichold, R., and Tavan, P. (2008). The “Hot-Solvent/Cold-Solute” Problem Revisited. *Journal of Chemical Theory and Computation*, **4**(8), 1293–1306.
- Marrink, S. J., Risselada, H. J., Yefimov, S., Tieleman, D. P., and de Vries, A. H. (2007). The MARTINI force field: coarse grained model for biomolecular simulations. *J. Phys. Chem. B*, **111**(27), 7812–7824.
- May, A., Pool, R., van Dijk, E., Bijlard, J., Abeln, S., Heringa, J., and Feenstra, K. A. (2014). Coarse-grained versus atomistic simulations: realistic interaction free energies for real proteins. *Bioinformatics (Oxford, England)*, **30**(3), 326–334.
- McCammon, J. A., Gelin, B. R., Karplus, M., and Wolynes, P. G. (1976). Hinge bending mode in lysozyme. *Nature*, **262**, 325–326.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, **21**(6), 1087–1092.
- Michaud-Agrawal, N., Denning, E. J., Woolf, T. B., and Beckstein, O. (2011). MDAnalysis: A toolkit for the analysis of molecular dynamics simulations. *Journal of Computational Chemistry*, **32**(10), 2319–2327.
- Monticelli, L., Kandasamy, S. K., Periole, X., Larson, R. G., Tieleman, D. P., and Marrink, S.-J. (2008). The MARTINI Coarse-Grained Force Field: Extension to Proteins. *J Chem Theory Comput*, **4**(5), 819–834.
- Moore, G. E. (1965). Cramming more components onto integrated circuits. *Electronics Magazine*, **38**(8), 4.
- Mor, A., Ziv, G., and Levy, Y. (2008). Simulations of proteins with inhomogeneous degrees of freedom: The effect of thermostats. *Journal of Computational Chemistry*, **29**(12), 1992–1998.
- Morrell, W. E. and Hildebrand, J. H. (1936). The Distribution of Molecules in a Model Liquid. *The Journal of Chemical Physics*, **4**(3), 224–227.
- Onufriev, A. V. and Case, D. A. (2019). Generalized Born Implicit Solvent Models for Biomolecules. *Annual Review of Biophysics*, **48**(1), 275–296.

- Oostenbrink, C., Villa, A., Mark, A. E., and van Gunsteren, W. F. (2004). A biomolecular force field based on the free enthalpy of hydration and solvation: the GROMOS force-field parameter sets 53A5 and 53A6. *J Comput Chem*, **25**(13), 1656–1676.
- Phillips, J. C., Braun, R., Wang, W., Gumbart, J., Tajkhorshid, E., Villa, E., Chipot, C., Skeel, R. D., Kale, L., and Schulten, K. (2005). Scalable molecular dynamics with NAMD. *J Comput Chem*, **26**(16), 1781–1802.
- Phillips, J. C., Hardy, D. J., Maia, J. D. C., Stone, J. E., Ribeiro, J. V., Bernardi, R. C., Buch, R., Fiorin, G., Hénin, J., Jiang, W., McGreevy, R., Melo, M. C. R., Radak, B. K., Skeel, R. D., Singharoy, A., Wang, Y., Roux, B., Aksimentiev, A., Luthey-Schulten, Z., Kalé, L. V., Schulten, K., Chipot, C., and Tajkhorshid, E. (2020). Scalable molecular dynamics on CPU and GPU architectures with NAMD. *The Journal of Chemical Physics*, **153**(4).
- Ponder, J. W. and Case, D. A. (2003). Force fields for protein simulations. *Adv. Protein Chem.*, **66**, 27–85.
- Pool, R., Heringa, J., Hoeffling, M., Schulz, R., Smith, J., and Feenstra, K. (2012). Enabling grand-canonical Monte Carlo: Extending the flexibility of GROMACS through the grompy python interface module. *Journal of Computational Chemistry*, **33**(12).
- Pronk, S., Pall, S., Schulz, R., Larsson, P., Bjelkmar, P., Apostolov, R., Shirts, M. R., Smith, J. C., Kasson, P. M., van der Spoel, D., Hess, B., and Lindahl, E. (2013). GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. *Bioinformatics*, **29**(7), 845–854.
- Rahman, A. (1964). Correlations in the Motion of Atoms in Liquid Argon. *Phys. Rev.*, **136**(2A), A405–A411.
- Rapaport, D. C. (2004). The Art of Molecular Dynamics Simulation. *The Art of Molecular Dynamics Simulation*.
- Raval, A., Piana, S., Eastwood, M. P., Dror, R. O., and Shaw, D. E. (2012). Refinement of protein structure homology models via long, all-atom molecular dynamics simulations. *Proteins: Structure, Function, and Bioinformatics*, **80**(8), 2071–2079.
- Reif, M. M., Hünenberger, P. H., and Oostenbrink, C. (2012). New Interaction Parameters for Charged Amino Acid Side Chains in the GROMOS Force Field. *Journal of Chemical Theory and Computation*, **8**(10).
- Roux, B. and Simonson, T. (1999). Implicit solvent models. *Biophysical Chemistry*, **78**(1-2).
- Shaw, D. E., Maragakis, P., Lindorff-Larsen, K., Piana, S., Dror, R. O., Eastwood, M. P., Bank, J. A., Jumper, J. M., Salmon, J. K., Shan, Y., and Wriggers, W. (2010). Atomic-level characterization of the structural dynamics of proteins. *Science*, **15**, 341–346.
- Shivakumar, D., Harder, E., Damm, W., Friesner, R. A., and Sherman, W. (2012). Improving the Prediction of Absolute Solvation Free Energies Using the Next Generation OPLS Force Field. *Journal of Chemical Theory and Computation*, **8**(8).
- Singh, N. and Li, W. (2019). Recent advances in coarse-grained models for biomolecules and their applications. *International Journal of Molecular Sciences*, **20**(15).
- Souza, P. C. T., Alessandri, R., Barnoud, J., Thallmair, S., Faustino, I., Grünewald, F., Patmanidis, I., Abdizadeh, H., Bruininks, B. M. H., Wassenaar, T. A., Kroon, P. C., Melcr, J., Nieto, V., Corradi, V., Khan, H. M., Domański, J., Javanainen, M., Martinez-Seara, H., Reuter, N., Best, R. B., Vattulainen, I., Monticelli, L., Periole, X., Tieleman, D. P., de Vries, A. H., and Marrink, S. J. (2021). Martini 3: a general purpose force field for coarse-grained molecular dynamics. *Nature Methods*, **18**(4).
- Stillinger, F. H. and Rahman, A. (1974). Improved simulation of liquid water by molecular dynamics. *The Journal of Chemical Physics*, **60**(4), 1545–57.
- Tian, C., Kasavajhala, K., Belfon, K. A. A., Raguette, L., Huang, H., Miguez, A. N., Bickel, J., Wang, Y., Pincay, J., Wu, Q., and Simmerling, C. (2020). ff19SB: Amino-Acid-Specific Protein Backbone Parameters Trained against Quantum Mechanics Energy Surfaces in Solution. *Journal of Chemical Theory and Computation*, **16**(1).
- van Aalten, D. M. F., de Groot, B. L., Berendsen, H. J. C., Findlay, J. B. C., and Amadei, A. (1997). A Comparison of Techniques for Calculating Protein Essential Dynamics. *J. Comput. Chem.*, **18**(2), 169–181.
- van Gunsteren, W. F., Berendsen, H. J., Hermans, J., Hol, W. G., and Postma, J. P. (1983). Computer simulation of the dynamics of hydrated protein crystals and its comparison with x-ray data. *Proceedings of the National Academy of Sciences*, **80**(14), 4315–4319.
- van Gunsteren, W. F., Billeter, S. R., Eising, A. A., Hünenberger, P. H., Krüger, P. A., Mark, E., Scott, W. R. P., and Tironi, I. G. (1996). *Biomolecular Simulation: The GROMOS96 Manual and User Guide*. Vdf Hochschulverlag AG an der ETH Zürich, Zürich, Switzerland.
- van Gunsteren, W. F., Burgi, R., Peter, C., and Daura, X. (2001). The Key to Solving the Protein-Folding Problem Lies in an Accurate Description of the Denatured State. *Angew. Chem. Int. Ed. Engl.*, **40**(2), 351–355.
- van Gunsteren, W. F., Bakowies, D., Baron, R., Chandrasekhar, I., Christen, M., Daura, X., Gee, P., Geerke, D. P., Glattli, A., Hunenberger, P. H., Kastenholz, M. A., Oostenbrink, C., Schenk, M.,

- Trzesniak, D., van der Vegt, N. F., and Yu, H. B. (2006). Biomolecular modeling: Goals, problems, perspectives. *Angew. Chem. Int. Ed. Engl.*, **45**(25), 4064–4092.
- Verlet, L. (1967). Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. *Physical Review*, **159**, 98–103.
- Warshel, A. and Levitt, M. (1976). Theoretical studies of enzymic reactions: dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme. *J. Mol. Biol.*, **103**(2), 227–249.
- Wood, W. W. and Parker, F. R. (1957). Monte Carlo Equation of State of Molecules Interacting with the Lennard-Jones Potential. I. A Supercritical Isotherm at about Twice the Critical Temperature. *The Journal of Chemical Physics*, **27**(3), 720–733.
- Yang, L., Tan, C.-h., Hsieh, M.-J., Wang, J., Duan, Y., Cieplak, P., Caldwell, J., Kollman, P. A., and Luo, R. (2006). New-Generation Amber United-Atom Force Field. *The Journal of Physical Chemistry B*, **110**(26).
- Yang, Y. I., Shao, Q., Zhang, J., Yang, L., and Gao, Y. Q. (2019). Enhanced sampling in molecular dynamics. **151**(7).

