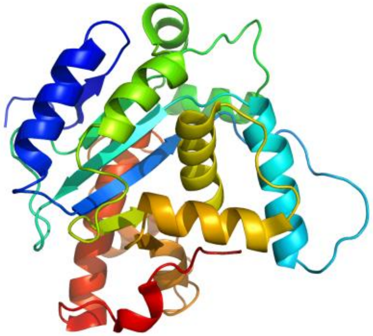
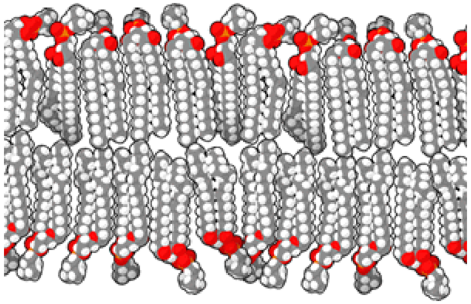
Simulations of Biochemical Systems with GROMACS

Alfredo Carvalho
(ajpalace@uevora.pt)
Universidade de Évora

Paulo Martel
(pmartel@ualg.pt)
Universidade do Algarve

António do Canto
(ammc@uevora.pt)
Universidade de Évora





High Performance Computing Chair
https://catedrahpc.uevora.pt/

Simulations of Biochemical Systems with GROMACS

Course programme

Session 1 – Today, June 17th

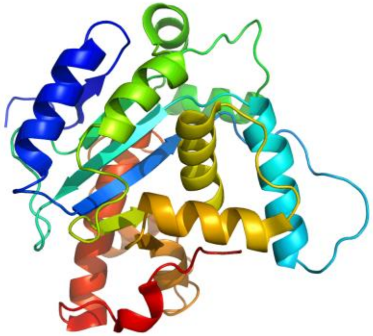
Lecturer: Alfredo Carvalho (***Me!***)

TOPIC: Fundamentals of Molecular Dynamics and GROMACS

- Introduction to Molecular Dynamics
- GROMACS overview
- Force fields in MD simulations

Simulations of Biochemical Systems with GROMACS

Course programme



Session 2 – Tomorrow, June 18th

Lecturer: Paulo Martel

TOPIC: Proteins

- Introduction to Protein Disulphide Isomerase (PDI)
- Preparation of WT and mutant PDIs and simulation setup
- Analysis of simulation results and discussion

ADVICE: You should install the PyMOL software (https://pymol.org/) on your computer in advance to this session so you can get the most out of it!

Simulations of Biochemical Systems with GROMACS

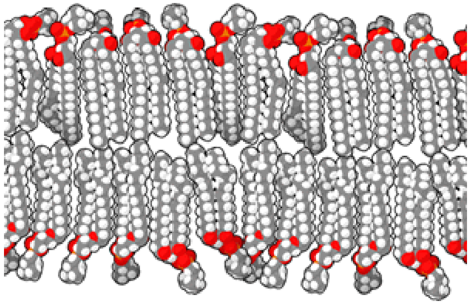
Course programme

Session 3 – Friday, June 19th

Lecturer: António do Canto

TOPIC: Biomembranes

- System setup and equilibration
- Running MD simulations with GROMACS
- Analysis of MD trajectories



Simulations of Biochemical Systems with GROMACS

Fundamentals of MD and GROMACS

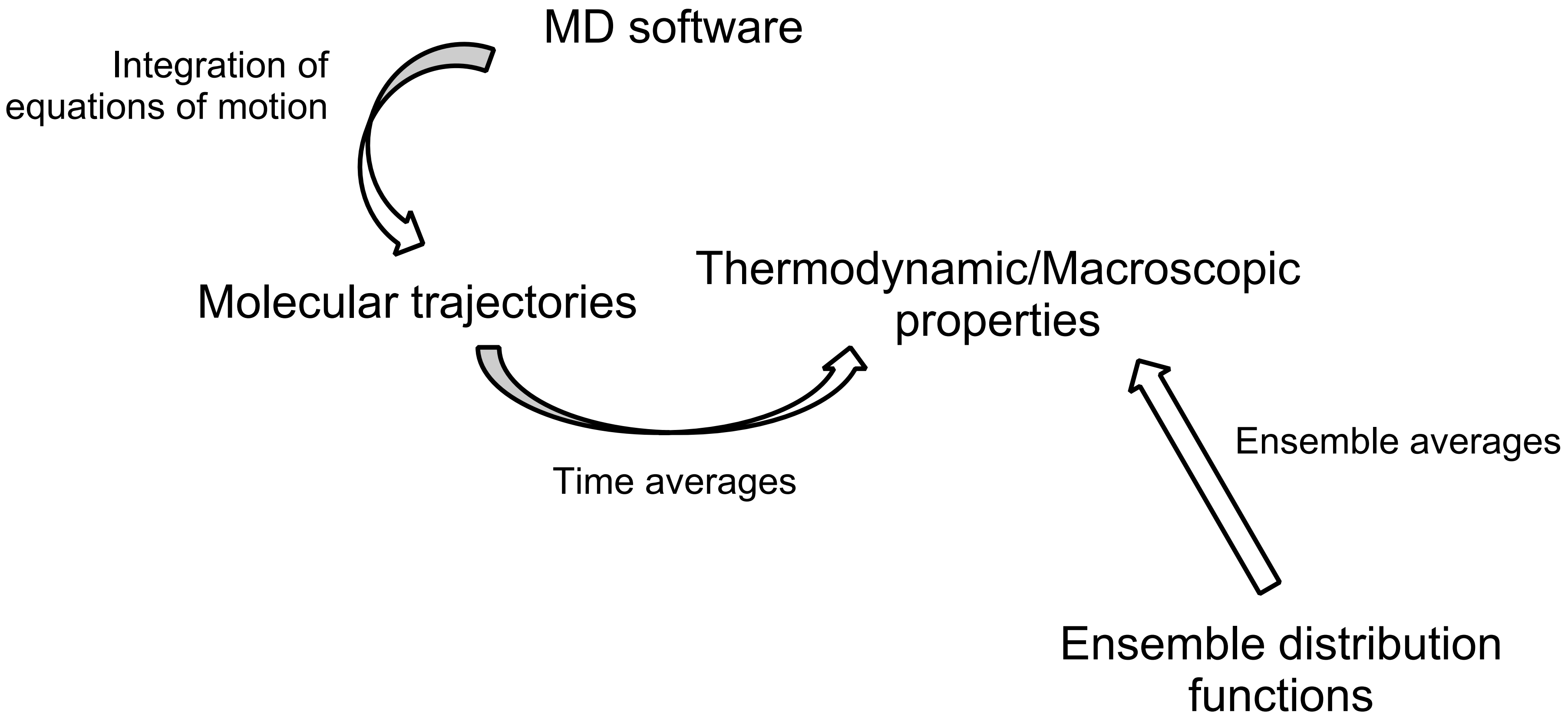
Alfredo Carvalho
High Performance Computing Chair
Universidade de Évora
ajpalace@uevora.pt

Introduction to Molecular Dynamics

Overview of Molecular Dynamics simulations

Definition and Scope

Molecular Dynamics simulations are essentially a sampling method for Statistical Mechanics ensembles



MD software

Integration of equations of motion

Molecular trajectories

Time averages

Thermodynamic/Macroscopic properties

Ensemble averages

Ensemble distribution functions

Overview of Molecular Dynamics simulations

Definition and Scope

Molecular Dynamics method

Limitations

Restricted set of ensembles (*NVE*, *NVT*, *NPT*)

Common FFs cannot describe chemical transformations

Computationally intensive and less parallelizable than MC, limiting the accessible time- and length-scale

Capabilities

Provides time-dependent information, can be used to predict dynamical properties

Hybrid QM/MM methods allow simulation of chemical events

Significant improvements at parallel computing performance in recent years

Coarse-grained methods allow increase in time- and length-scale

Dynamic development, significant enhancements over time

Overview of Molecular Dynamics simulations

Historical Background

Some landmarks

“Classical antiquity” – the paper and pencil age

Long history of the work on generic forms of equations of motion (Newton, Lagrange, Hamilton, etc.) and methods of integration (Verlet, Runge-Kutta, etc.) as a central subject of Classical Mechanics

Nowadays – computer age

Early work by Alder and Wainwright (1957) on the simulation of perfectly elastic collisions between hard spheres, using an IBM 704 computer, is generally considered to be the first reported MD simulation

The first realistic simulation of matter is probably the work of J.B. Gibson et al. (1960) who studied the dynamics of radiation damage in a model representation of solid copper, using a Born–Mayer repulsive interaction along with a cohesive surface force

In 1964, A. Rahman used a Lennard-Jones potential in the simulation of liquid argon; the calculated properties, such as the coefficient of self-diffusion, compared well with experimental data

Overview of Molecular Dynamics simulations

Historical Background

Rahman and Stillinger reported for the first time, in 1971, an MD study on a model of liquid water

The first MD simulation of a simple protein was due to Karplus and collaborators, and appeared shortly after, in 1977

In 1980, Andersen described how to extend MD to enable it to sample the isoenthalpic (constant pressure) ensemble

Following in the steps of Andersen, Nosé incorporated, in 1984, a new dynamical variable coupling the kinetic energy of the atoms to the external (bath) temperature, in such a way that the resulting dynamics can be shown to sample the canonical ensemble

In 1985, Car and Parrinello offered a scheme to perform first-principles MD simulations

In 1991, Berne and co-workers developed the multiple time scale MD algorithm

And, then, the story continues …

Overview of Molecular Dynamics simulations

Basic Workflow

Setup

- Defining the system
- Preparing initial coordinates
- Choosing the force field
- Defining simulation parameters (*T*, *P*, algorithms, etc.)

Run

- Energy minimization → Equilibration → Production

Analyze

- Getting basic simulation data from output
- Trimming trajectories and performing further calculations from them
- Visualizing data

Publish

Theoretical foundations: Newton's equations and integration algorithms

Newton's equations of motion

The basis of the MD method is the **production of a trajectory** by **integrating the equations of motion** of the system

The equations of motion can be defined by application of **Newton's second law** (note that there are other alternative formalisms for Classical Mechanics):

$$\vec{F}_i = m_i \frac{d^2\vec{q}_i}{dt^2}$$

(assuming constant masses)

The forces F_i acting on particle i are the result of the **interactions** of that particle with all other particles

Therefore, forces F_i depend on the **separation distances** between all pairs of particles *(more on the calculation of interactions/forces later)*

Theoretical foundations: Newton's equations and integration algorithms

Numerical integration algorithms

Equations of motion are differential equations, because they depend on the particles' coordinates (in the forces) as well as their derivatives (velocities) and second derivatives (accelerations)

Equations for more than two particles **cannot be solved analytically**, they must be **solved numerically**

The numerical solution is discrete (time divided in steps) as opposed to the continuous analytical solution

There are several different algorithms, but they all essentially involve calculations of forces based on particles' coordinates at a previous time step and use them to calculate particles' velocities and coordinates for the following time step

An integration algorithms, to be used in MD, must be stable (satisfy conservation laws, such as energy), satisfy time-reversibility, and should be fast

Theoretical foundations: Newton's equations and integration algorithms

Numerical integration algorithms

One of the most popular algorithms for integrating the equations of motion is the **Verlet algorithm**, because of its simplicity, stability and efficiency

Consider a Taylor expansion backward and forward in time:

$$r(t+\delta t) = r(t) + v(t)\delta t + \frac{1}{2}a(t)\delta t^2 + \frac{1}{3!}\dddot{r}(t)\delta t^3 + O(\delta t^4)$$

$$r(t-\delta t) = r(t) - v(t)\delta t + \frac{1}{2}a(t)\delta t^2 - \frac{1}{3!}\dddot{r}(t)\delta t^3 + O(\delta t^4)$$

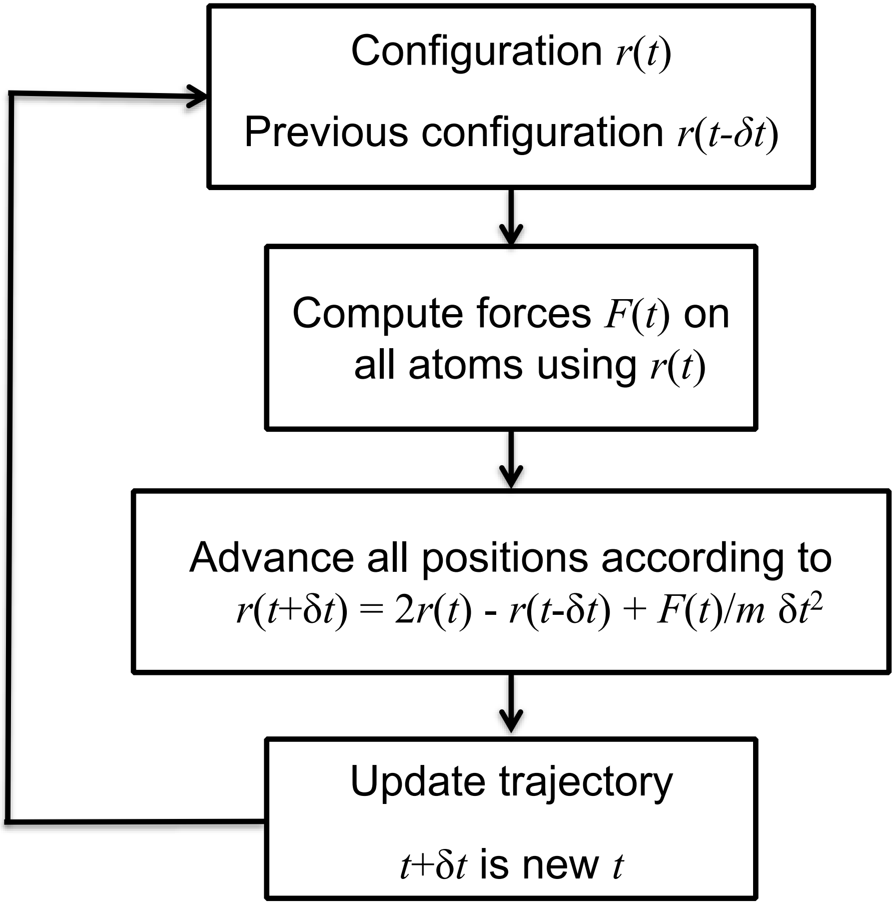
Adding the two expressions and rearranging:

$$r(t+\delta t) = 2r(t) - r(t-\delta t) + a(t)\delta t^2 + O(\delta t^4)$$

Theoretical foundations: Newton's equations and integration algorithms

Numerical integration algorithms

Flow diagram for Verlet algorithm



Configuration $r(t)$

Previous configuration $r(t-\delta t)$

Compute forces $F(t)$ on all atoms using $r(t)$

Advance all positions according to $r(t+\delta t) = 2r(t) - r(t-\delta t) + F(t)/m\ \delta t^2$

Update trajectory

$t+\delta t$ is new t

Theoretical foundations: Newton's equations and integration algorithms

Numerical integration algorithms

Properties of Verlet's method:

- Errors in trajectories within $O(\delta t^4)$
- Preserves time reversal symmetry
- Excellent energy conservation

Modifications and alternative schemes exist, always within the second order approximation:

- **Leapfrog**
- **Velocity-Verlet**

providing better stability or computational efficiency

Theoretical foundations: Newton's equations and integration algorithms

Time integration and stability

The choice of an adequate time step δt is a critical point

- must be **not too short** or the simulation will be too slow
- must be **not too long** or the integration may diverge (energy will not be conserved and, ultimately, the system may ‘blow-up’)

An order of magnitude less than the fastest motion is generally considered ideal

The fastest motion is usually the bond stretching vibrations (especially those involving the lightest H atoms):

- the period of C-H bond vibration is ~10 fs, so use a time step of 1 fs

If these motions are not of interest:

- you can constrain these bonds and increase the time step (2 fs or 4 fs)

Theoretical foundations: Newton's equations and integration algorithms

Potential Energy Functions

Forces which determine the dynamics of the system result from the interactions between the particles

Force is the symmetrical of the potential energy gradient $F = -\nabla U$

The most rigorous theoretical treatment would be to obtain the quantum mechanical energy (*ab initio* molecular dynamics) but that isn’t feasible for large systems and/or long simulation times

Based on the Born-Oppenheimer approximation, the motion of the nuclei can be separated from the motion of electrons:

- nuclei motion assumed to be well described (in almost all cases) by Classical Mechanics (MD simulations)
- electronic energies are calculated for frozen nuclei configurations and certain analytical functions are adjusted to these data; nuclei move under potentials defined by these functions (or, alternatively, potentials may be defined empirically, to reproduce certain experimental properties)

Theoretical foundations: Newton's equations and integration algorithms

Potential Energy Functions

The analytical form of these potential functions is usually a decomposition of terms which have a chemical interpretation, i.e. they model certain types of intra- or inter-molecular interactions

Interactions are usually divided in two classes:

- Bonded interactions: exclusively intra-molecular, higher energy terms, namely bond stretching (harmonic potential), angle bending, torsions
- Non-bonded (or weak) interactions: lower energy, most inter-molecular (but can also occur intra-molecularly) interactions, namely van der Waals (usually modeled by Lennard-Jones potentials) and electrostatics (ion-ion, ion-dipole and dipole-dipole, calculated by the Coulomb potential)

The van der Waals interactions are **short-ranged** and, therefore, a **cut-off** around each particle can be applied for their calculation; however, electrostatic interactions are **long-ranged** and have to be estimated for a large system, as is done by the Ewald summation method for an infinite periodic system

Theoretical foundations: Newton's equations and integration algorithms

Potential Energy Functions

The coherent set of functional forms and their parameterizations, which are developed to model completely the interactions of a set of molecules/chemical compounds (sometimes for a special purpose, such as reproducing with high accuracy certain types of properties) is called a **Force Field**, which we will address next

Simulation components: Force Fields, coordinates, and time steps

What is a Force Field?

In Molecular Dynamics a molecule is described as a set of charged points (atoms) linked, according to the molecule's topology, by springs (bonds)

To describe the time evolution of molecule conformations and aggregation, one uses a Force Field to describe the bond lengths and force constants, bond angles and torsions, as well as the non-bonded van der Waals and electrostatic interactions between atoms

The Force Field is a collection of equations and associated constants which have been designed to reproduce molecular geometry and selected properties of tested structures

Simulation components: Force Fields, coordinates, and time steps

What is a Force Field?

We may begin by considering the decomposition of the potential, that depends on the coordinates of all the atoms, as sum of terms of individual atoms, pairs, triplets, etc.

$$U(r_1,r_2,\dots,r_N)=\underbrace{\sum_i u^{(1)}(r_i)}_{\text{External potential}}+\underbrace{\sum_i\sum_{j>i}u^{(2)}(r_i,r_j)}_{\text{Pairs term}}+\underbrace{\sum_i\sum_{j>i}\sum_{k>j}u^{(3)}(r_i,r_j,r_k)}_{\text{Triplets term}}+\dots$$

The first term, depending on the coordinates of individual atoms, represents the effect of an external potential and corresponds to an additive constant on energy; it does not represent molecule interactions

The second term, the **pair potential**, is the **most important**; it depends only on the magnitude of the pair separation $r_{ij} = |r_i - r_j|$, so it may be written $u^{(2)}(r_{ij})$

Higher order terms are usually neglected, although $u^{(3)}$ contributions (**three-body effects**) may still be significant at high densities; in some Force Fields, pair terms partially include average three-body effects in effective potentials, which thus become density dependent

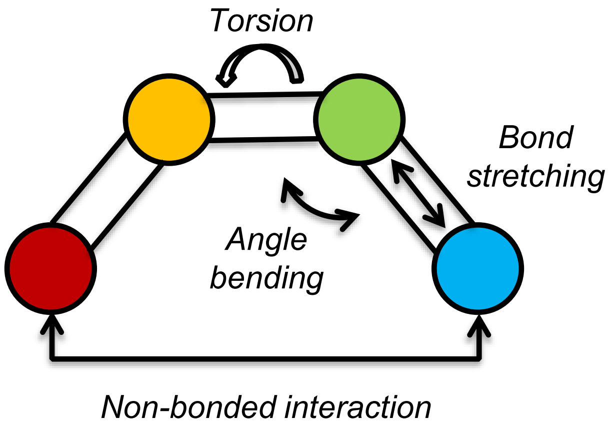
Simulation components: Force Fields, coordinates, and time steps

What is a Force Field?

Taking the interaction potential as the **pair potential approximation**, it can be decomposed as a sum of contributions of different kinds, as already discussed:

$$U(r_1, r_2, \dots, r_N) = \underbrace{U_{str} + U_{bend} + U_{tors} + U_{cross}}_{\text{Intra-molecular terms}} + U_{vdW} + U_{el} + U_{pol}$$

This decomposition was proposed by Linus Pauling in the 1930s



Simulation components: Force Fields, coordinates, and time steps

What is a Force Field?

Bond stretching energy

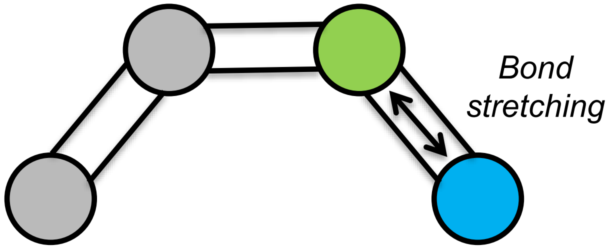
$$U_{str}(r_{ij}) = k_b(r_{ij} - r_0)^2$$

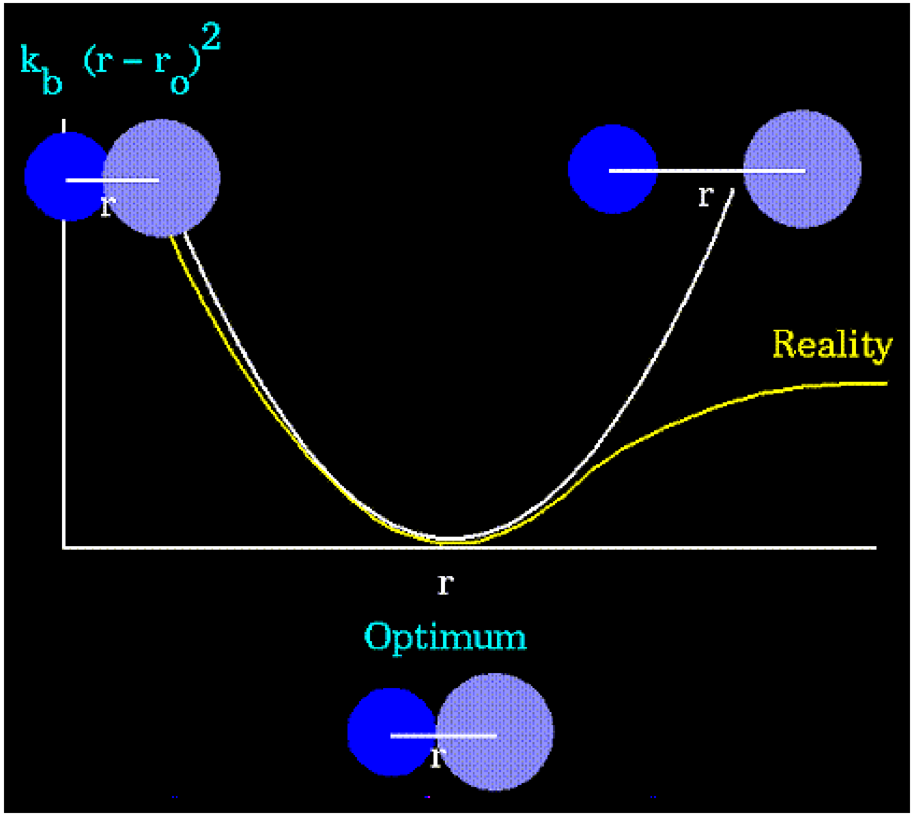
Harmonic approximation

k_b is the spring constant of the bond

r_0 is the bond length at equilibrium

Unique k_b and r_0 assigned for each bond pair, i.e. C-C, O-H





Harmonic approximation vs. Reality

Under the harmonic approximation, bond breaking is impossible

Simulation components: Force Fields, coordinates, and time steps

What is a Force Field?

Bond stretching energy

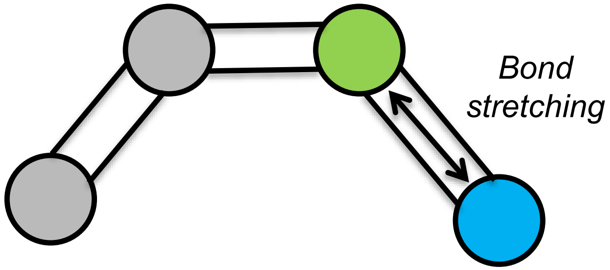
$$U_{str}(r_{ij}) = k_b(r_{ij} - r_0)^2$$

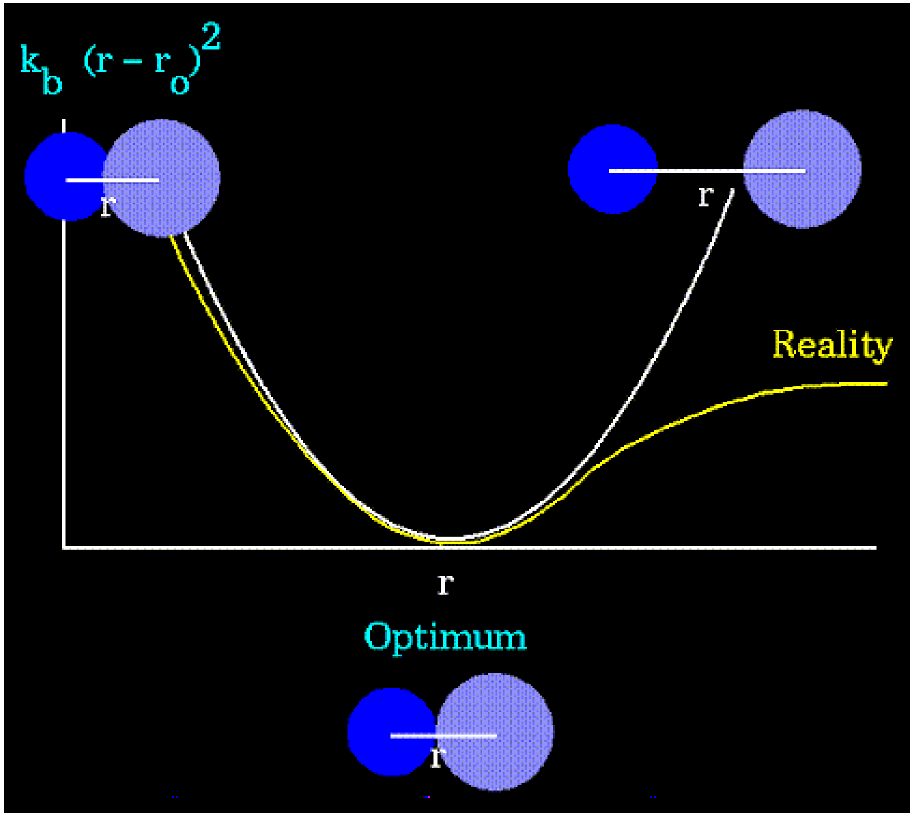
Harmonic approximation

k_b is the spring constant of the bond

r_0 is the bond length at equilibrium

Unique k_b and r_0 assigned for each bond pair, i.e. C-C, O-H





Harmonic approximation vs. Reality

Under the harmonic approximation, bond breaking is impossible

Simulation components: Force Fields, coordinates, and time steps

What is a Force Field?

Bond stretching energy

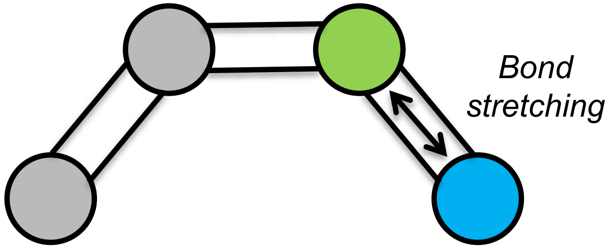
$$U_{str}(r_{ij}) = k_b (r_{ij} - r_0)^2$$

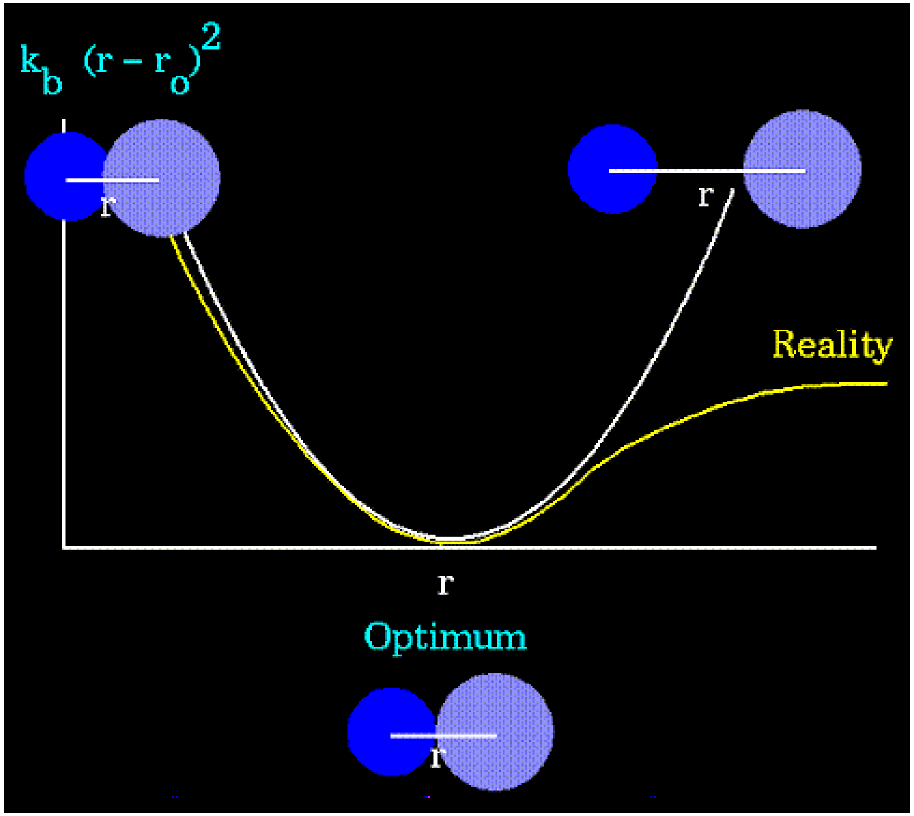
Harmonic approximation

k_b is the spring constant of the bond

r_0 is the bond length at equilibrium

Unique k_b and r_0 assigned for each bond pair, i.e. C-C, O-H





Harmonic approximation vs. Reality

Under the harmonic approximation, bond breaking is impossible

Simulation components: Force Fields, coordinates, and time steps

What is a Force Field?

Bending energy

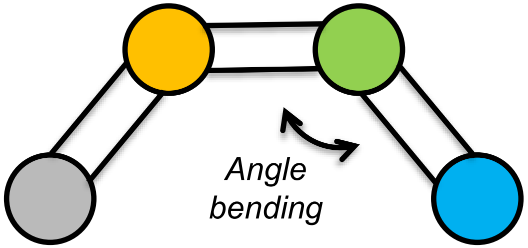
$$U_{bend}(\theta_{ijk}) = k_\theta(\theta_{ijk} - \theta_0)^2$$

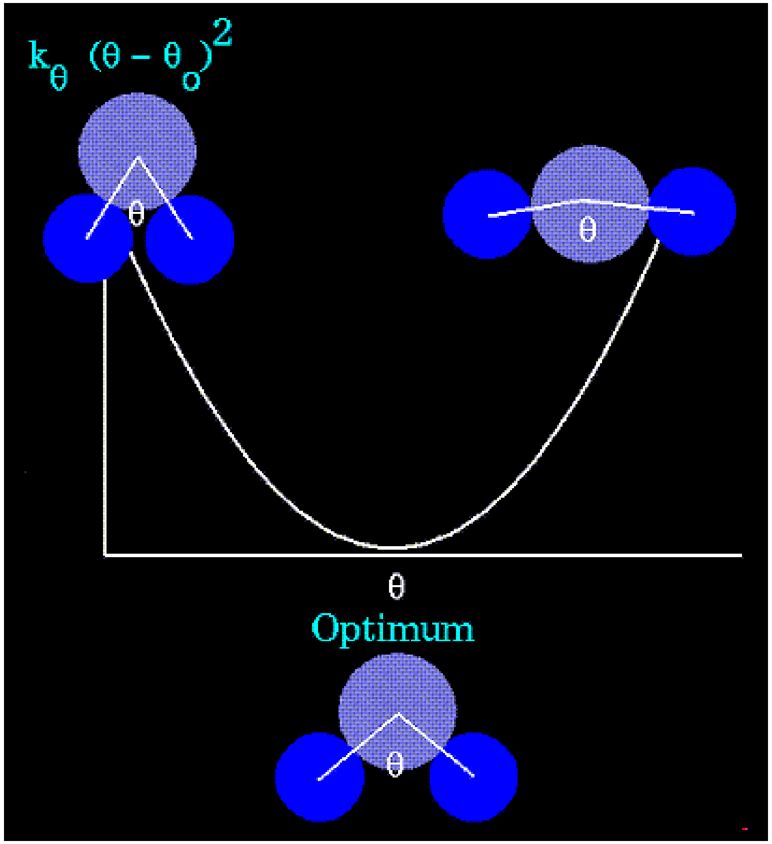
Also the harmonic approximation

k_θ is the spring constant of the angle bending

θ_0 is the bonds angle at equilibrium

Unique parameters for angle bending are assigned to each bonded triplet of atoms based on their types (e.g. C-C-C, C-O-C, C-C-H, etc.)





Simulation components: Force Fields, coordinates, and time steps

What is a Force Field?

Bending energy

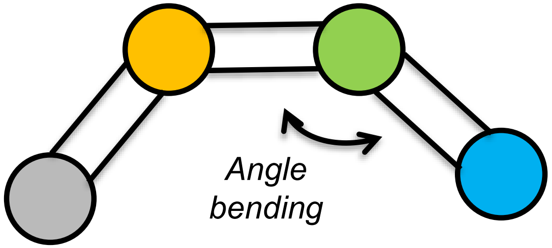
$$U_{bend}(\theta_{ijk}) = k_\theta(\theta_{ijk} - \theta_0)^2$$

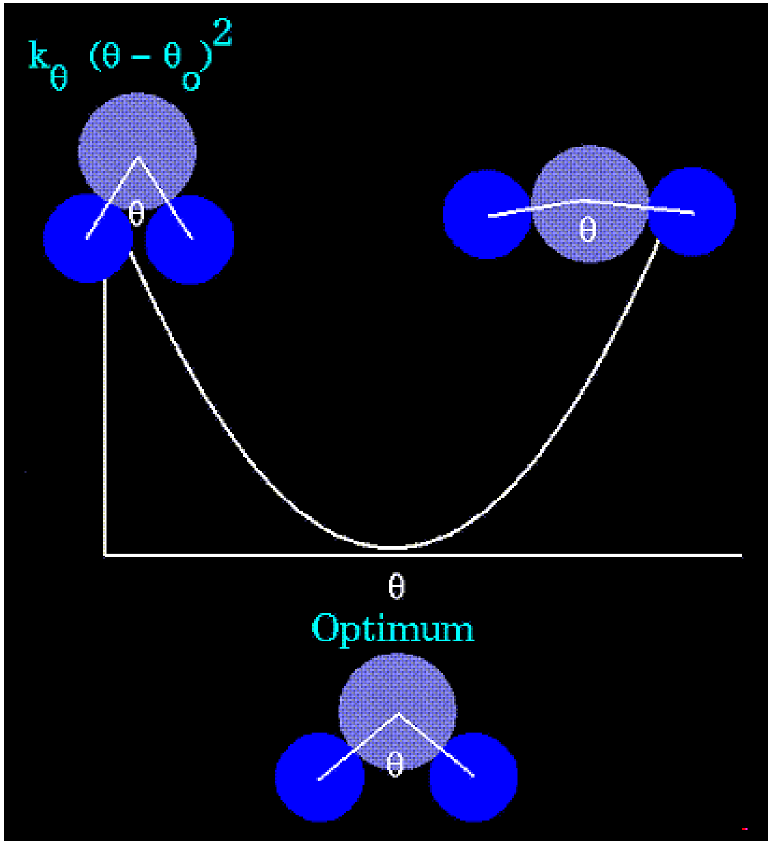
Also the harmonic approximation

k_θ is the spring constant of the angle bending

θ_0 is the bonds angle at equilibrium

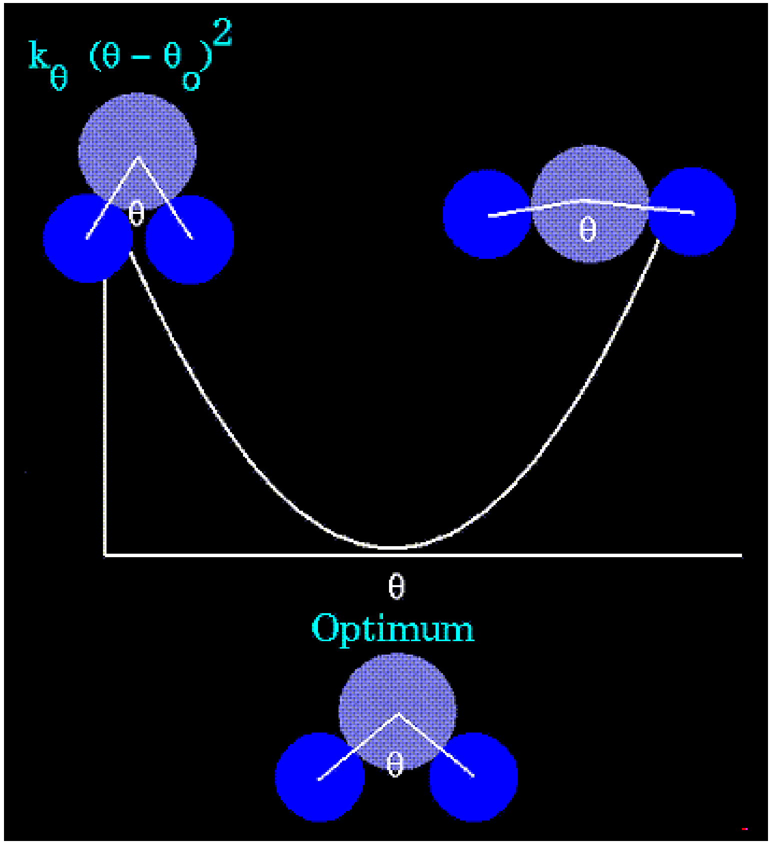
Unique parameters for angle bending are assigned to each bonded triplet of atoms based on their types (e.g. C-C-C, C-O-C, C-C-H, etc.)





Simulation components: Force Fields, coordinates, and time steps

What is a Force Field?



Bending energy

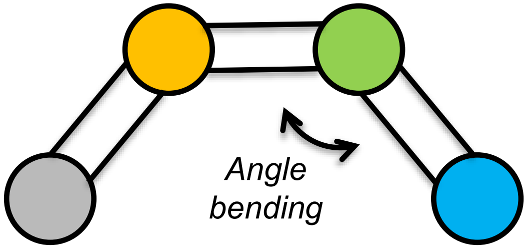
$$U_{bend}(\theta_{ijk}) = k_\theta(\theta_{ijk} - \theta_0)^2$$

Also the harmonic approximation

k_θ is the spring constant of the angle bending

θ_0 is the bonds angle at equilibrium

Unique parameters for angle bending are assigned to each bonded triplet of atoms based on their types (e.g. C-C-C, C-O-C, C-C-H, etc.)



Simulation components: Force Fields, coordinates, and time steps

What is a Force Field?

Torsion energy

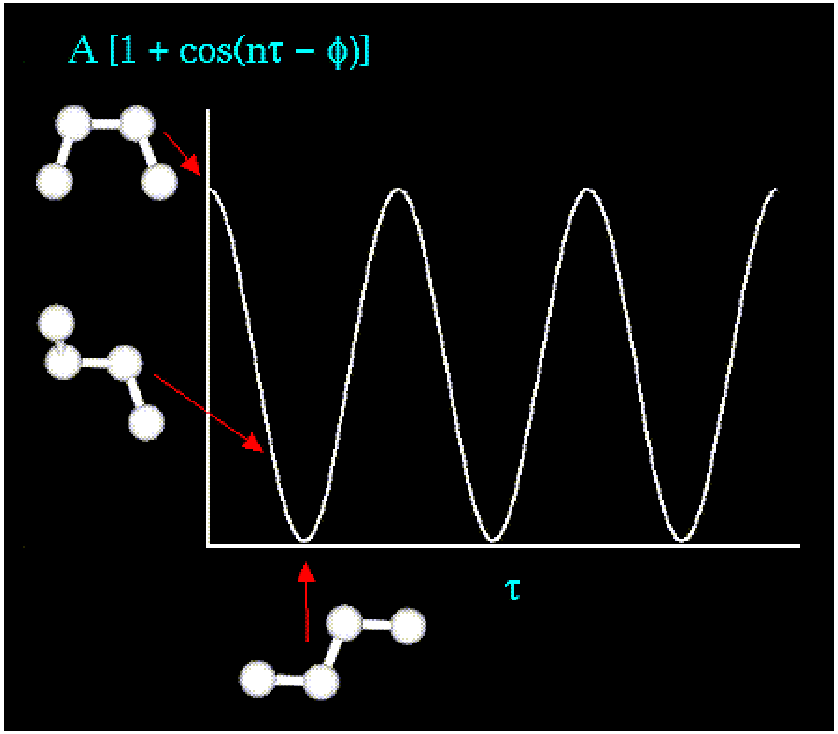
$$U_{tors}(\varphi_{ijkl}) = A\left[1 + \cos(n\tau - \varphi_{ijkl})\right]$$

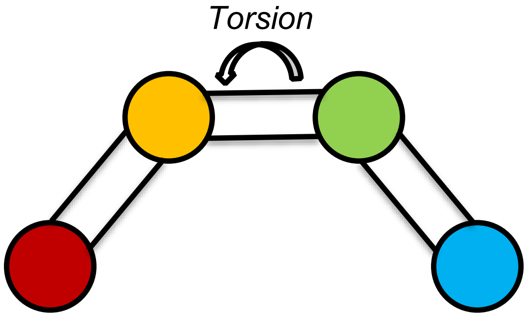
Periodic function

A controls the amplitude of the curve

n controls its periodicity

ϕ shifts the entire curve along the rotation angle axis (τ)





The parameters are determined from curve fitting.

Unique parameters for torsional rotation are assigned to each bonded quartet of atoms based on their types (e.g. C-C-C-C, C-O-C-N, H-C-C-H, etc.)

Simulation components: Force Fields, coordinates, and time steps

What is a Force Field?

Torsion energy

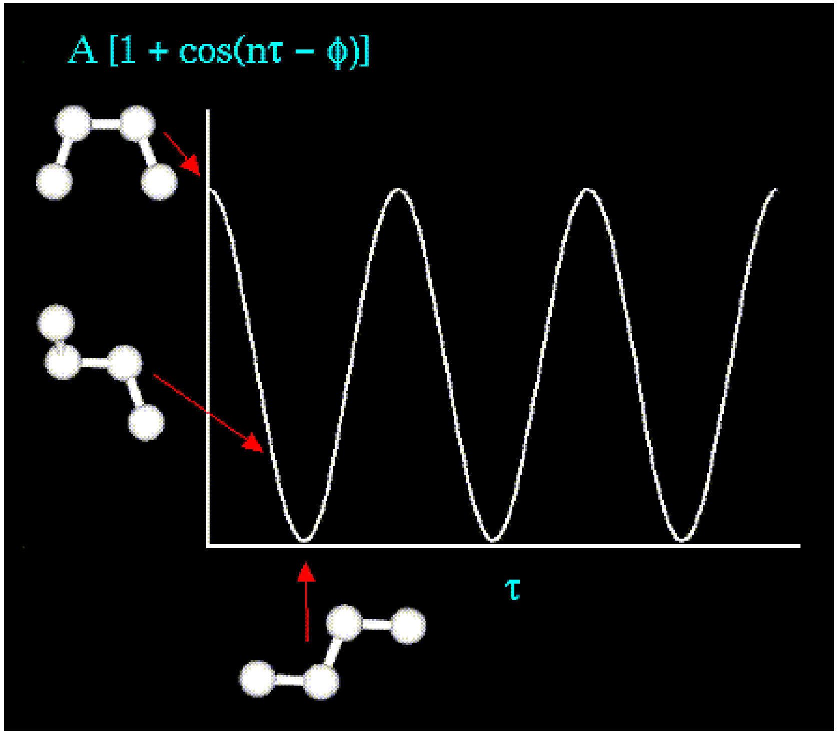
$$U_{tors}(\varphi_{ijkl}) = A\left[1 + \cos(n\tau - \varphi_{ijkl})\right]$$

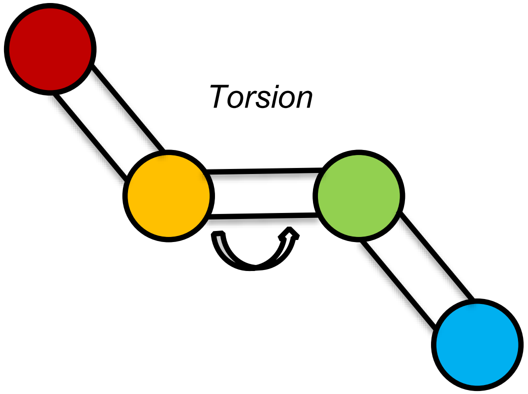
Periodic function

A controls the amplitude of the curve

n controls its periodicity

ϕ shifts the entire curve along the rotation angle axis (τ)





The parameters are determined from curve fitting.

Unique parameters for torsional rotation are assigned to each bonded quartet of atoms based on their types (e.g. C-C-C-C, C-O-C-N, H-C-C-H, etc.)

Simulation components: Force Fields, coordinates, and time steps

What is a Force Field?

Torsion energy

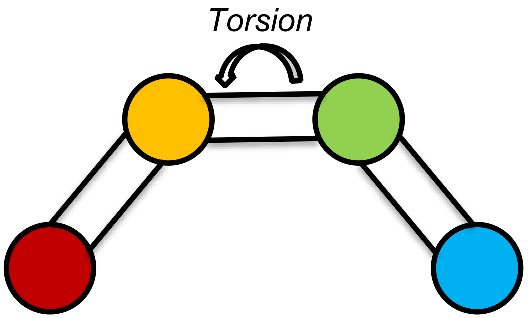
$$U_{tors}(\varphi_{ijkl}) = A\left[1 + \cos(n\tau - \varphi_{ijkl})\right]$$

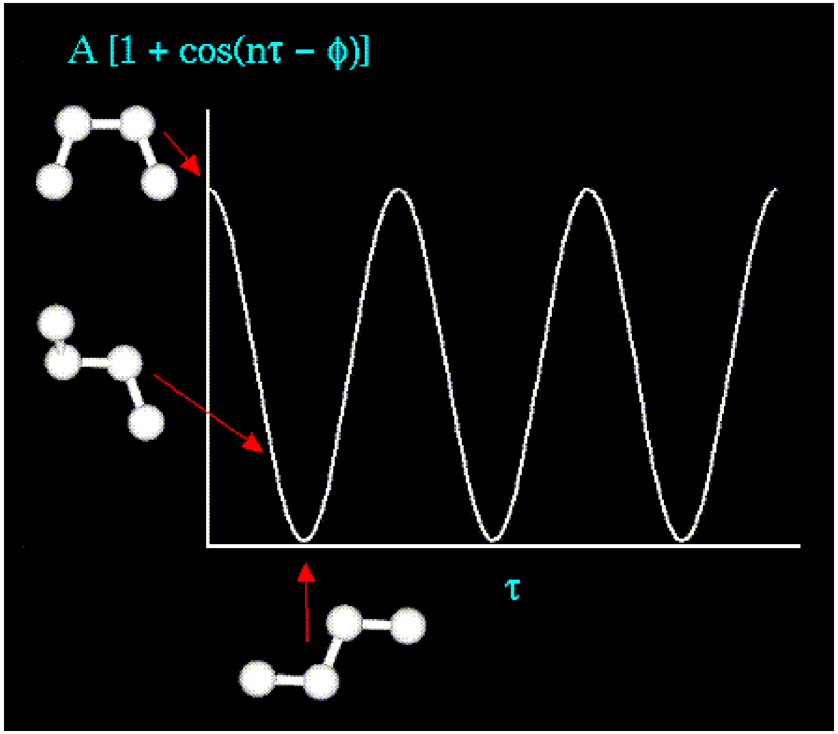
Periodic function

A controls the amplitude of the curve

n controls its periodicity

ϕ shifts the entire curve along the rotation angle axis (τ)





The parameters are determined from curve fitting.

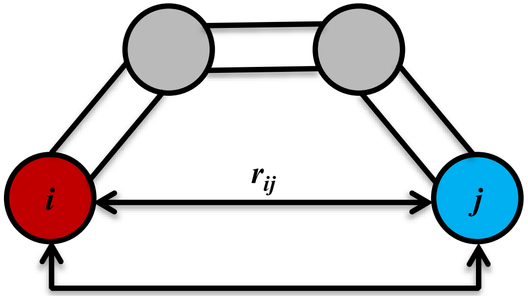
Unique parameters for torsional rotation are assigned to each bonded quartet of atoms based on their types (e.g. C-C-C-C, C-O-C-N, H-C-C-H, etc.)

Simulation components: Force Fields, coordinates, and time steps

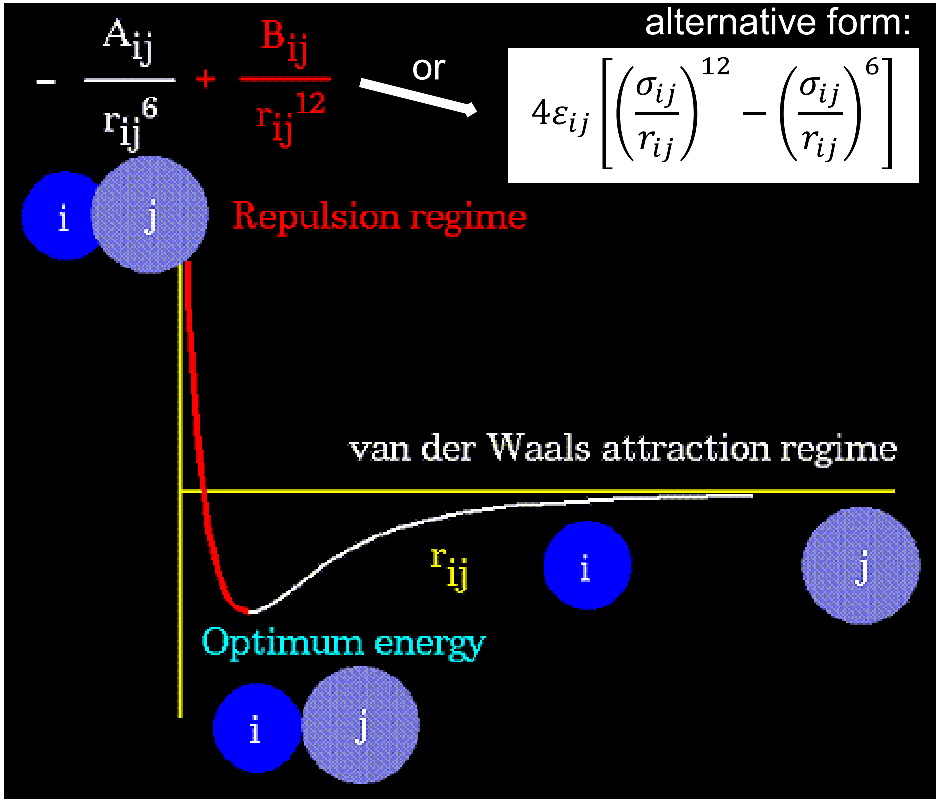
What is a Force Field?

Non-bonded energy

$$U_{nb}(r_{ij}) = \underbrace{-\frac{A_{ij}}{r_{ij}^{6}} + \frac{B_{ij}}{r_{ij}^{12}}}_{\text{van der Waals term (Lennard-Jones potential)}} + \underbrace{\frac{q_i q_j}{r_{ij}}}_{\text{electrostatic term}}$$



Non-bonded interaction



A determines the degree the attractiveness

B determines the degree of repulsion

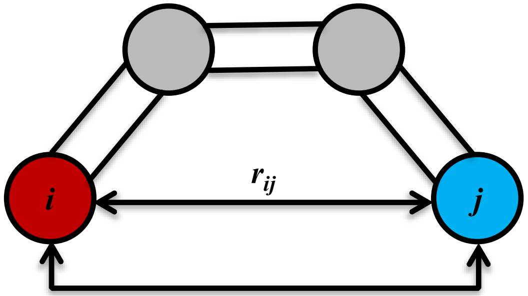
q is the charge

Simulation components: Force Fields, coordinates, and time steps

What is a Force Field?

Non-bonded energy

$$U_{nb}(r_{ij}) = \underbrace{-\frac{A_{ij}}{r_{ij}^{6}} + \frac{B_{ij}}{r_{ij}^{12}}}_{\text{van der Waals term (Lennard-Jones potential)}} + \underbrace{\frac{q_i q_j}{r_{ij}}}_{\text{electrostatic term}}$$

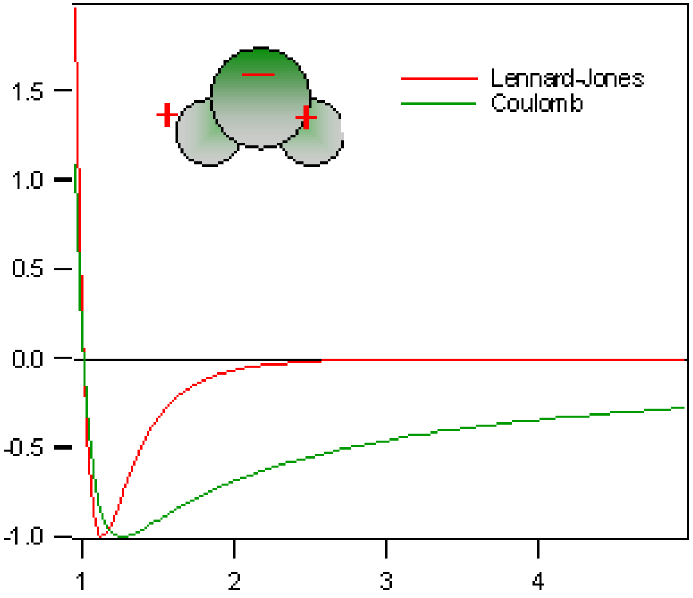


Non-bonded interaction

Coulomb potential

Point charges

Multipoles: dipoles, quadropoles, …



Electrostatic interactions are **long-ranged**

Simulation components: Force Fields, coordinates, and time steps

What is a Force Field?

Refinements

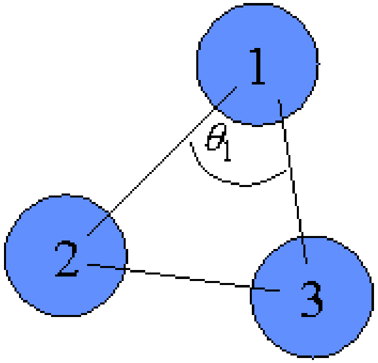
Polarization

Charge redistribution due to the presence of neighboring molecules (involves iterative calculations)

Multi-body potentials

Axilrod-Teller potential

$$u(r_1, r_2, r_3) = \frac{3}{2} \frac{E\alpha_1\alpha_2\alpha_3}{r_{12}^3 r_{23}^3 r_{13}^3} (3\cos\theta_1 \cos\theta_2 \cos\theta_3 + 1)$$



Simulation components: Force Fields, coordinates, and time steps

What is a Force Field?

United-atom force fields

A simplification that can be introduced in the atomic description of molecules is the **omission of some atoms** in certain groups and substitution of those atoms by a single **pseudo-atom** which is attributed interaction parameters that should describe the effects of the group of atoms collectively

This approach is called a **united-atom force field**

Simulation components: Force Fields, coordinates, and time steps

What is a Force Field?

United-atom force fields

Earliest cases of united-atom potentials were the representation of diatomic gas molecules by single pseudo-atoms

More relevant (and common) cases are the united-atom force fields that omit, in organic molecules, the hydrogen atoms in weakly polar bonds with carbon, specifically substituting methyl and methylene groups by pseudo-atoms

The beneficial effects of this approach, besides reducing the total number of atoms, is that it also eliminates the vibrations with highest frequencies, thereby allowing the increase of the time step

Simulation components: Force Fields, coordinates, and time steps

What is a Force Field?

United-atom force fields

Force fields that provide parameters for every type of atom in a system are called **all-atom force fields**

The omission of the explicit representation of some atoms isn’t necessarily a compromise between speed and accuracy

In fact, depending on the properties to be calculated, a careful parameterization of the pseudo-atoms may introduce the effects of all the atoms of the group effectively and yield results of good quality for properties where the omitted atoms are not critical

Commonly used Force Fields

AMBER (Assisted model building with energy refinement) – Used for peptides, proteins and nucleic acids. Can be extended with GLYCAM parameter set for carbohydrates as well as Lipid14 and GAFFLipid for lipids

CHARMM (Chemistry at Harvard Macromolecular Mechanics) - Originally developed for peptides and proteins, but later extended for other types of biomolecules and small organic molecules

OPLS (Optimized Potential for Liquid Simulations) - Developed for the simulation of organic liquids

GROMOS (GROningen MOlecular Simulation) - Force field associated with the GROMOS software (the predecessor to GROMACS) also aimed at proteins originally, with extensions for nucleic acids, carbohydrates and lipids

GROMACS (GROningen MAchine for Chemical Simulations) - GROMACS also has its own Force Field, originally aimed for biomolecules but with a much wider range of applications these days

GAFF (General AMBER force field) - For small organic molecules to facilitate simulations of drugs and small molecule ligands in conjunction with biomolecules

CGenFF (CHARMM General force field) - A force field for drug-like molecules compatible with the CHARMM all-atom additive force fields (an analogue to GAFF for AMBER)

MMFF (Merck Molecular Force Field) - Has been aimed more at drug-like organic compounds than at proteins

MM2/MM3/... - used for conformational analysis of hydrocarbons and other small organic molecules (designed to reproduce the equilibrium geometries precisely)

UFF (Universal Force Field) – A general force field with parameters for the full periodic table up to and including the actinoids

BUFF - Biopolymer counterpart to UFF

Others: Coarse-grained (MARTINI, etc.), Polarizable (AMBER, CHARMM, AMOEBA, etc.), Machine Learning, etc.

Several water models: SPC, SPC/E, flexible SPC, TIP3P, TIP4P, etc.

Simulation components: Force Fields, coordinates, and time steps

Coordinates, Periodic Boundary Conditions, Minimum Image, …

Although a coordinate change is often useful to theoretically approach a solution for certain physical problems, in practice it is more convenient in the computations for a molecular simulation to treat all the positions with Cartesian coordinates

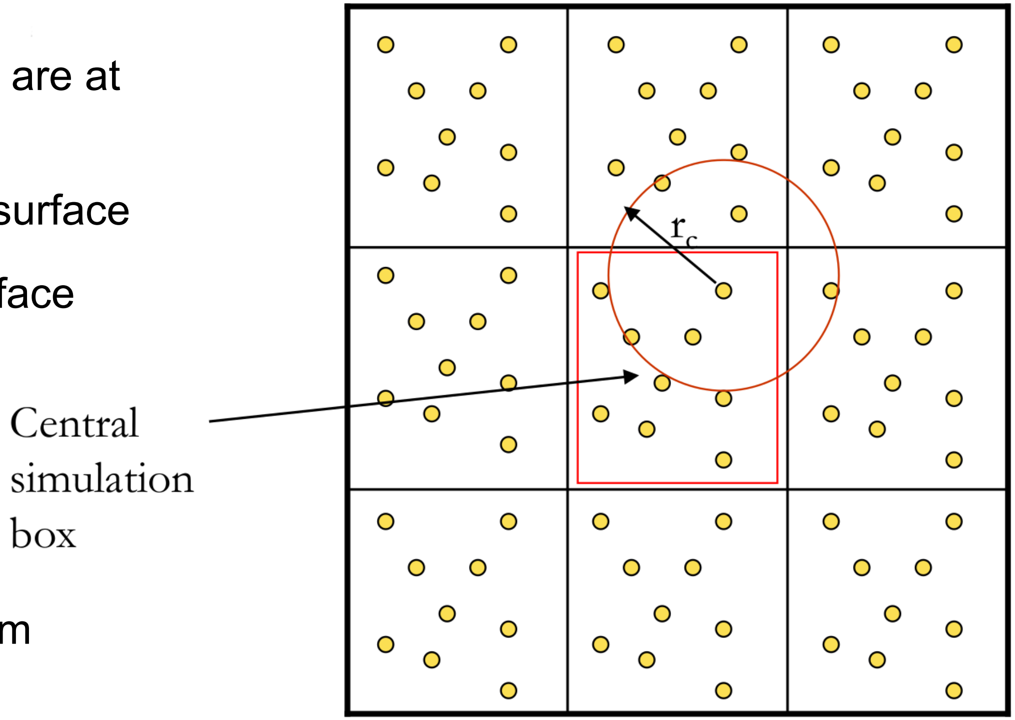
Sizes and distances at a molecular scale are within the ångström – nanometer range

The smaller the system, the more particles are at the surface:

- 1000 atom cubic crystal, 49% on the surface
- 10^6 atom cubic crystal, 6% on the surface

Would like to simulate infinite bulk surrounding N-particle system

- eg. periodic boundary
- usually accompanied with the minimum image convention and cut-off radius



Simulation components: Force Fields, coordinates, and time steps

Coordinates, Periodic Boundary Conditions, Minimum Image, …

Application of a cut-off for short-ranged interactions is reasonable

But in electrostatic interactions, it leads to incorrect results

There are several solutions, but the most popular is Ewald sums, or a derivative form: Particle Mesh Ewald

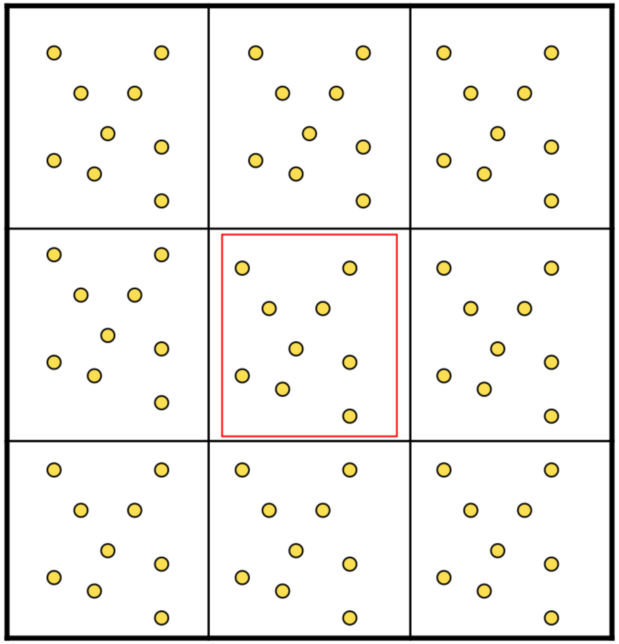
Ewald sums: electrostatics split into two terms, real space and k-space sum

$$U_{elec} = \sum_{\mathbf{n}} \sum_{i<j} \frac{q_i q_j}{\left|\vec{r}_i - \vec{r}_j + \vec{\mathbf{n}}\right|}$$

$$= \sum_{i<j} q_i q_j \frac{erfc(\alpha r_{ij})}{r_{ij}}$$

$$+ \sum_{i<j} \sum_{k} \frac{1}{\pi V} \frac{4\pi^2}{k^2} q_i q_j \exp(-k^2 / 4\alpha^2) \cos(\vec{k} \cdot \vec{r}_{ij})$$

$$- \frac{\alpha}{\sqrt{\pi}} \sum_i q_i^2$$



Simulation components: Force Fields, coordinates, and time steps

Time Steps and Simulation Length

As said before the size of the time step determines the speed of the simulation, but it cannot be chosen arbitrarily

The size of the time step is determined by the shortest vibration period occurring in the simulated system. Therefore, it typically needs to be 1-2 fs in order to capture all atomic vibrations.

Total simulation time can range from nanoseconds to microseconds or longer depending on the study. That corresponds to 10^6-10^9 integration steps.

The size of the time step may be increased by employing united-atom force fields and/or freezing some bond lengths by applying constraints algorithms

However, for addressing very large systems or very long simulation times, other approaches probably have to be taken, such as performing multiple time scale dynamics or employing coarse-grained force fields

GROMACS Overview

Introduction to GROMACS software

What is GROMACS?

GROMACS =
GROningen **MA**chine for **C**hemical **S**imulations

GROMACS is a **versatile package to perform molecular dynamics**, i.e. simulate the Newtonian equations of motion for systems with hundreds to millions of particles and is a **community-driven project**.

(*From GROMACS webpage: https://www.gromacs.org/ about.html*)

It was initially developed at the University of Groningen (as is evidenced in the acronym) and is an open-source software package, which allows continuing improvements from a large base of contributors.

Introduction to GROMACS software

Key Features of GROMACS

Some of the most relevant aspects of GROMACS include:

- High performance and efficiency
- Scalability to many processors, using threads, MPI or GPUs
- Extensive library of molecular topologies and force fields
- Extensive set of analysis tools
- User-friendly with extensive documentation

Introduction to GROMACS software

System Requirements and Installation

Supported platforms

Linux, Windows, macOS

Basic hardware requirements

GROMACS can be installed in various different operating systems and architectures, with very basic hardware requirements, if you will be only analyzing results
However, for performing typical simulations, use of heavy computing power is generally necessary; access to HPC resources is sometimes crucial for large projects

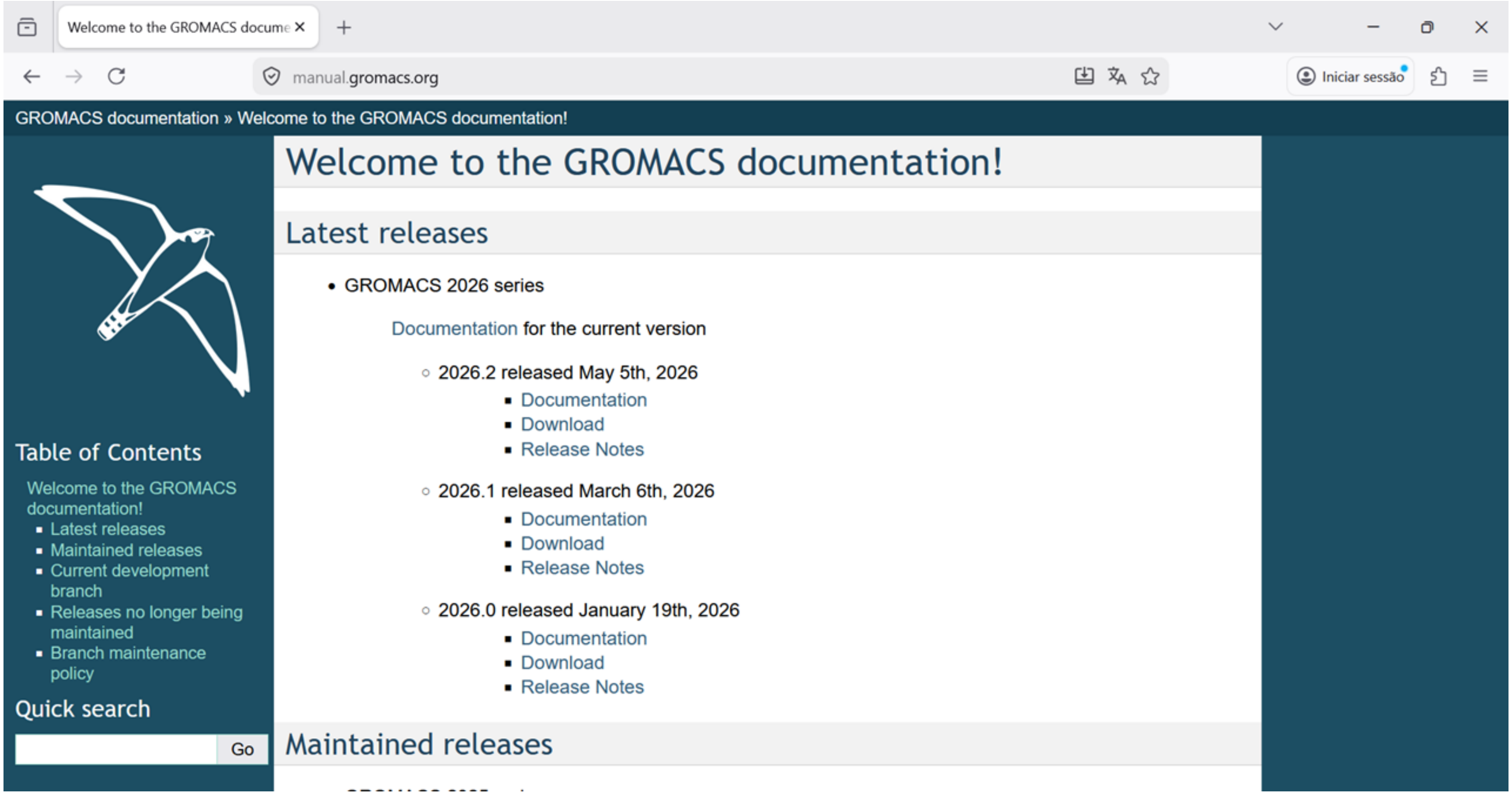
Steps for downloading and installing GROMACS

We won't be addressing private installations of GROMACS in this seminar as we will work on Oblivion, but instructions for installation are available in GROMACS website (**https://manual.gromacs.org/current/install-guide/**)

As a note: installation requires compilation of sources; if you have Linux and are a regular user, it will be easier, but installation in Windows is also possible. There may be binary versions around in the Internet, but beware of viruses!

Introduction to GROMACS software

System Requirements and Installation



GROMACS documentation » Welcome to the GROMACS documentation!

Welcome to the GROMACS documentation!

Latest releases

- GROMACS 2026 series

  Documentation for the current version

  - 2026.2 released May 5th, 2026
    - Documentation
    - Download
    - Release Notes
  - 2026.1 released March 6th, 2026
    - Documentation
    - Download
    - Release Notes
  - 2026.0 released January 19th, 2026
    - Documentation
    - Download
    - Release Notes

Maintained releases

Table of Contents

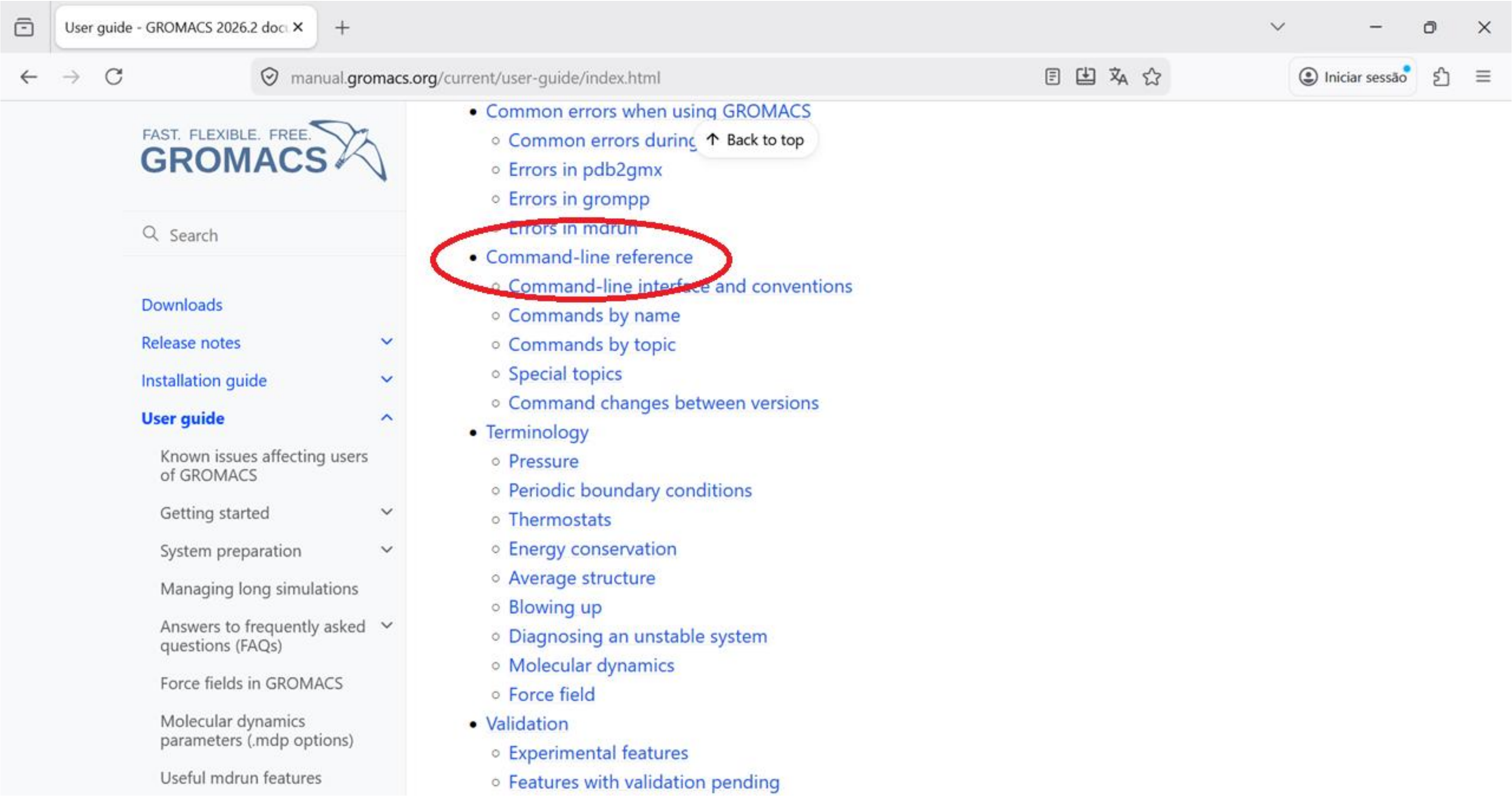
- Welcome to the GROMACS documentation!
  - Latest releases
  - Maintained releases
  - Current development branch
  - Releases no longer being maintained
  - Branch maintenance policy

Quick search

GROMACS’ website provides much important information…

Introduction to GROMACS software

System Requirements and Installation



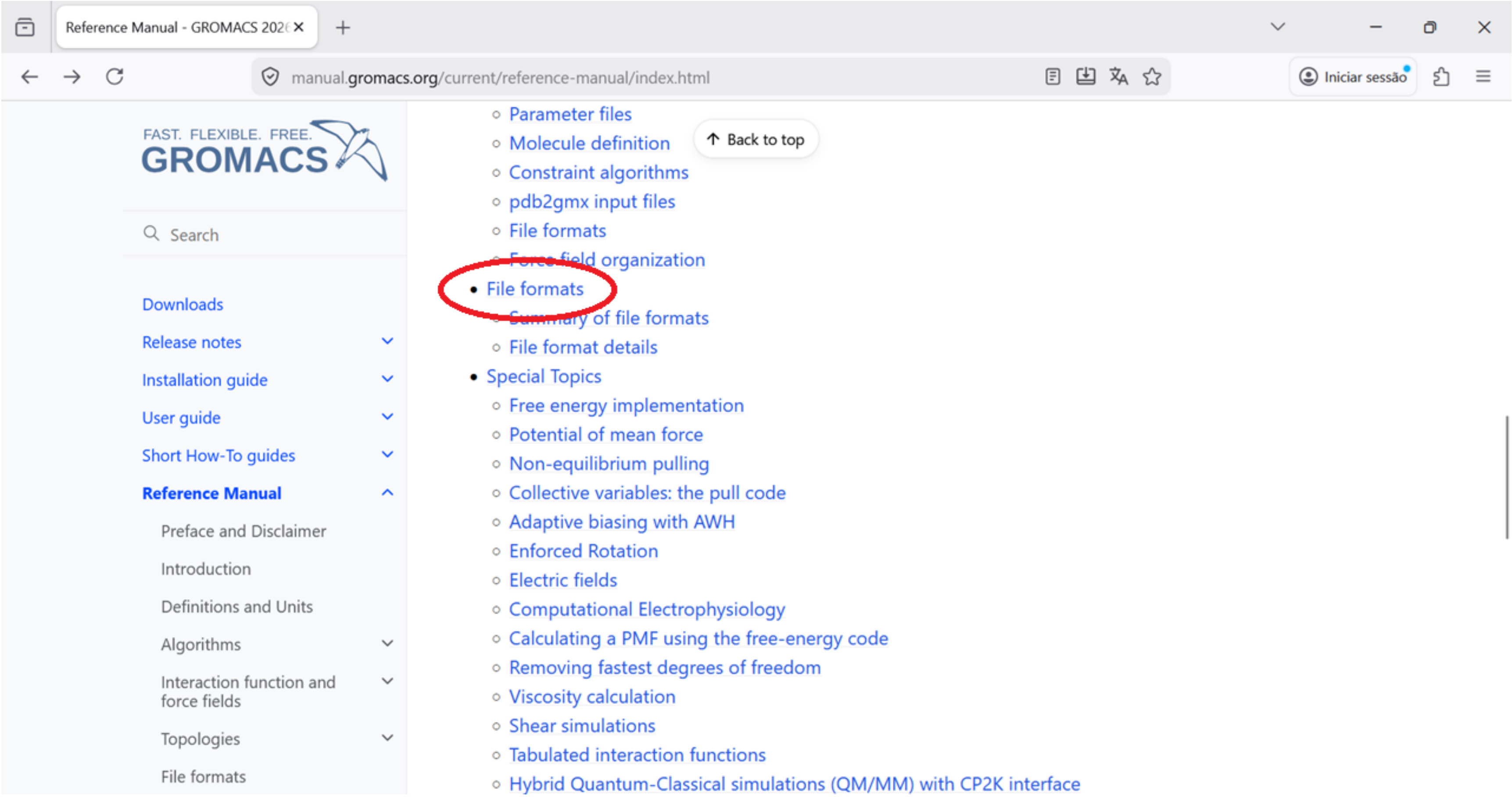
- Downloads
- Release notes
- Installation guide
- User guide
  - Known issues affecting users of GROMACS
  - Getting started
  - System preparation
  - Managing long simulations
  - Answers to frequently asked questions (FAQs)
  - Force fields in GROMACS
  - Molecular dynamics parameters (.mdp options)
  - Useful mdrun features

- Common errors when using GROMACS
  - Common errors during
  - Errors in pdb2gmx
  - Errors in grompp
  - Errors in mdrun
- Command-line reference
  - Command-line interface and conventions
  - Commands by name
  - Commands by topic
  - Special topics
  - Command changes between versions
- Terminology
  - Pressure
  - Periodic boundary conditions
  - Thermostats
  - Energy conservation
  - Average structure
  - Blowing up
  - Diagnosing an unstable system
  - Molecular dynamics
  - Force field
- Validation
  - Experimental features
  - Features with validation pending

… like the User Guide, in particular the section about the Command-line tools, ...

Introduction to GROMACS software

System Requirements and Installation



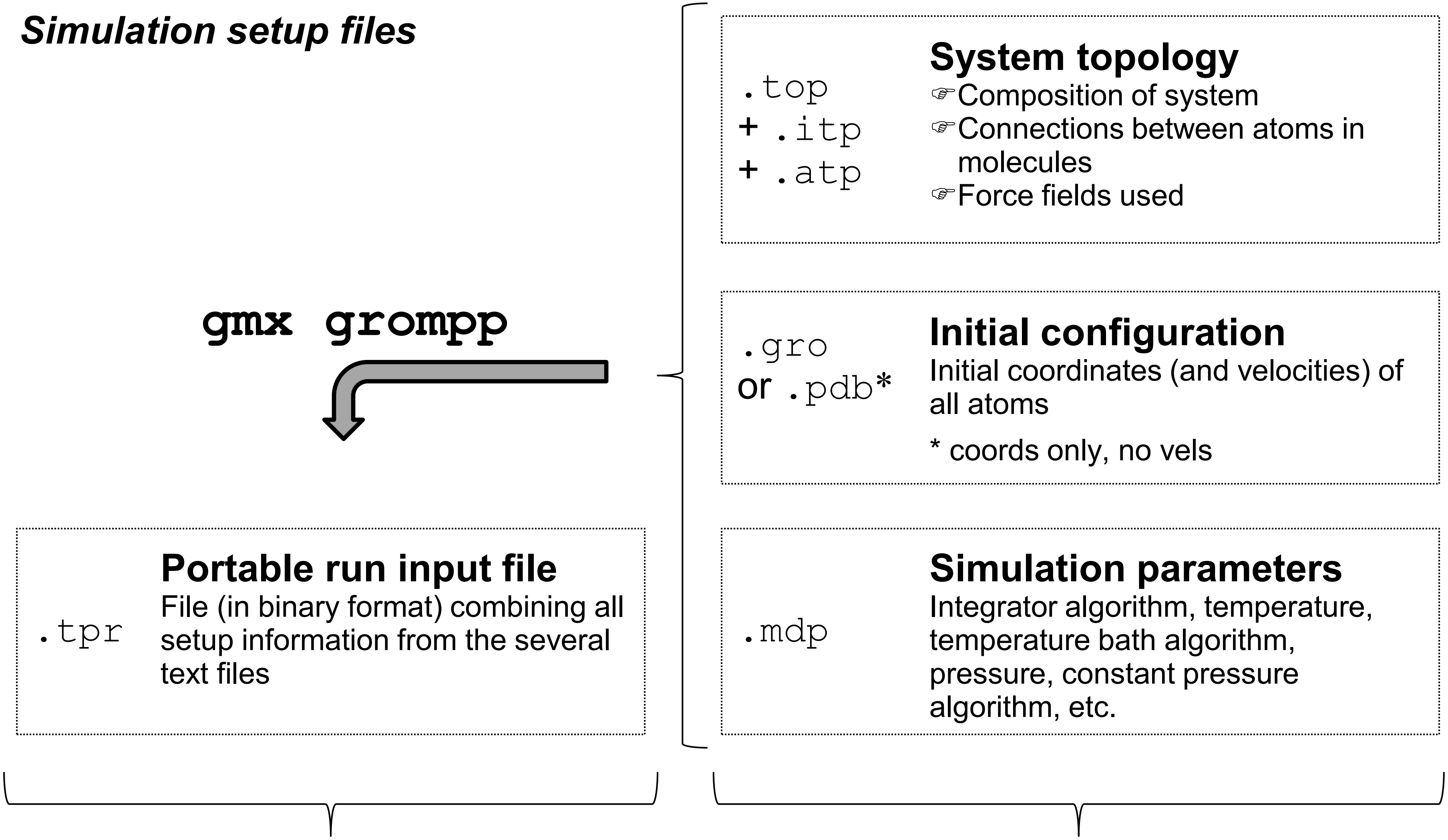
- Downloads
- Release notes
- Installation guide
- User guide
- Short How-To guides
- **Reference Manual**
  - Preface and Disclaimer
  - Introduction
  - Definitions and Units
  - Algorithms
  - Interaction function and force fields
  - Topologies
  - File formats

    - Parameter files
    - Molecule definition
    - Constraint algorithms
    - pdb2gmx input files
    - File formats
    - Force field organization
- File formats
    - Summary of file formats
    - File format details
- Special Topics
    - Free energy implementation
    - Potential of mean force
    - Non-equilibrium pulling
    - Collective variables: the pull code
    - Adaptive biasing with AWH
    - Enforced Rotation
    - Electric fields
    - Computational Electrophysiology
    - Calculating a PMF using the free-energy code
    - Removing fastest degrees of freedom
    - Viscosity calculation
    - Shear simulations
    - Tabulated interaction functions
    - Hybrid Quantum-Classical simulations (QM/MM) with CP2K interface

… and the Reference Manual, in particular the section about the file formats.

Introduction to GROMACS software

Simulation setup files



`.top`
`+ .itp`
`+ .atp`

System topology
- ☞Composition of system
- ☞Connections between atoms in molecules
- ☞Force fields used

`.gro`
or `.pdb*`

Initial configuration
Initial coordinates (and velocities) of all atoms

* coords only, no vels

`.mdp`

Simulation parameters
Integrator algorithm, temperature, temperature bath algorithm, pressure, constant pressure algorithm, etc.

`gmx grompp`

`.tpr`

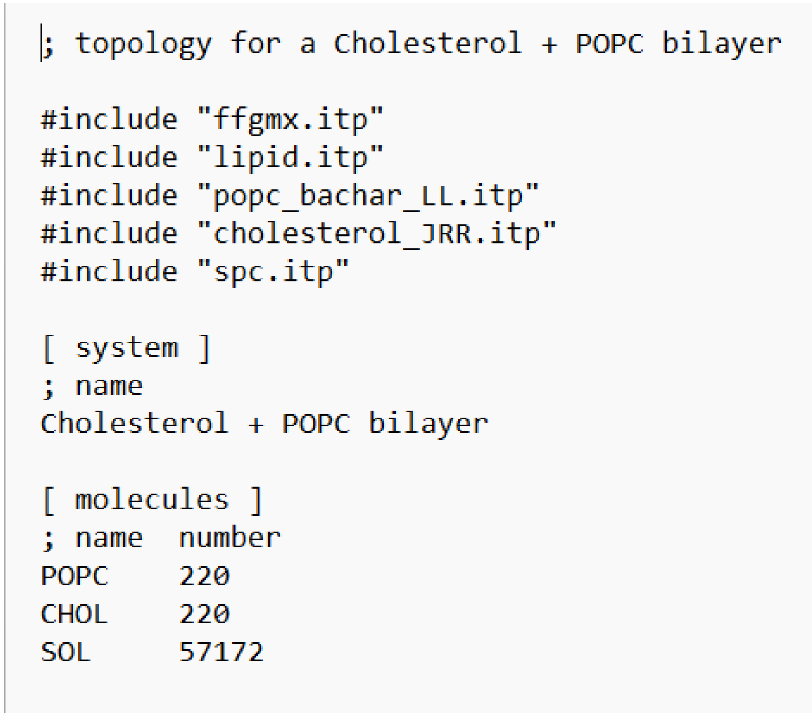
Portable run input file
File (in binary format) combining all setup information from the several text files

Single binary file

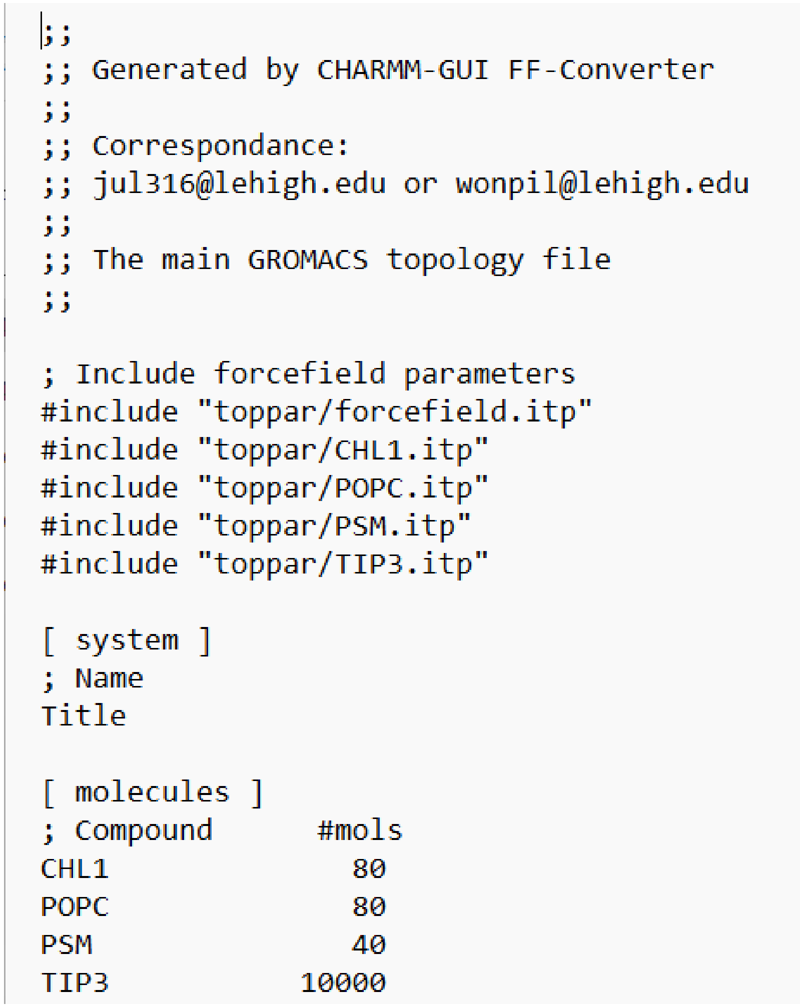
Several text files

Introduction to GROMACS software

Simulation setup files (`.top` files)



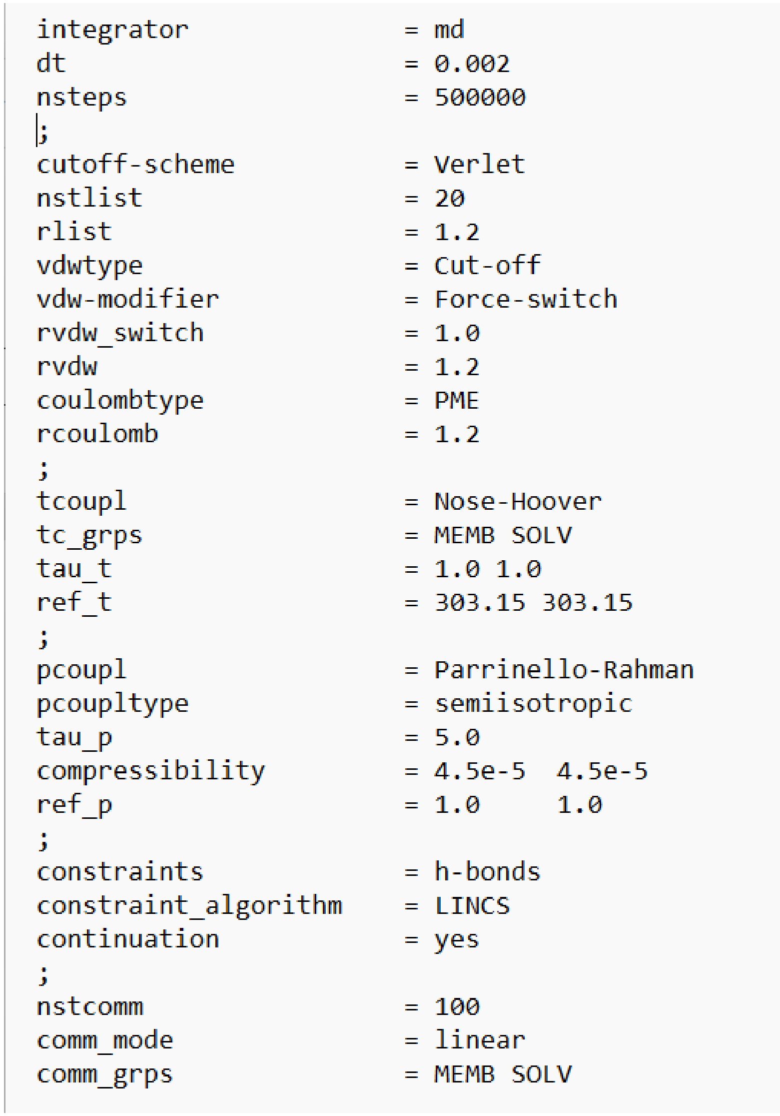
```
; topology for a Cholesterol + POPC bilayer

#include "ffgmx.itp"
#include "lipid.itp"
#include "popc_bachar_LL.itp"
#include "cholesterol_JRR.itp"
#include "spc.itp"

[ system ]
; name
Cholesterol + POPC bilayer

[ molecules ]
; name   number
POPC     220
CHOL     220
SOL      57172
```



```
;;
;; Generated by CHARMM-GUI FF-Converter
;;
;; Correspondance:
;; jul316@lehigh.edu or wonpil@lehigh.edu
;;
;; The main GROMACS topology file
;;

; Include forcefield parameters
#include "toppar/forcefield.itp"
#include "toppar/CHL1.itp"
#include "toppar/POPC.itp"
#include "toppar/PSM.itp"
#include "toppar/TIP3.itp"

[ system ]
; Name
Title

[ molecules ]
; Compound      #mols
CHL1              80
POPC              80
PSM               40
TIP3           10000
```

Introduction to GROMACS software

Simulation setup files (`.mdp` file)

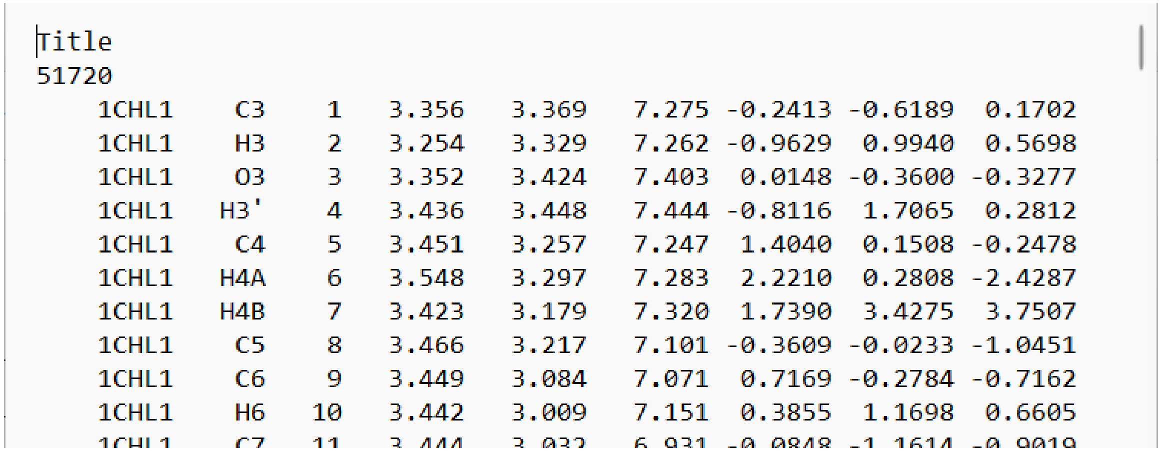


```
integrator               = md
dt                       = 0.002
nsteps                   = 500000
;
cutoff-scheme            = Verlet
nstlist                  = 20
rlist                    = 1.2
vdwtype                  = Cut-off
vdw-modifier             = Force-switch
rvdw_switch              = 1.0
rvdw                     = 1.2
coulombtype              = PME
rcoulomb                 = 1.2
;
tcoupl                   = Nose-Hoover
tc_grps                  = MEMB SOLV
tau_t                    = 1.0 1.0
ref_t                    = 303.15 303.15
;
pcoupl                   = Parrinello-Rahman
pcoupltype               = semiisotropic
tau_p                    = 5.0
compressibility          = 4.5e-5  4.5e-5
ref_p                    = 1.0     1.0
;
constraints              = h-bonds
constraint_algorithm     = LINCS
continuation             = yes
;
nstcomm                  = 100
comm_mode                = linear
comm_grps                = MEMB SOLV
```

Introduction to GROMACS software

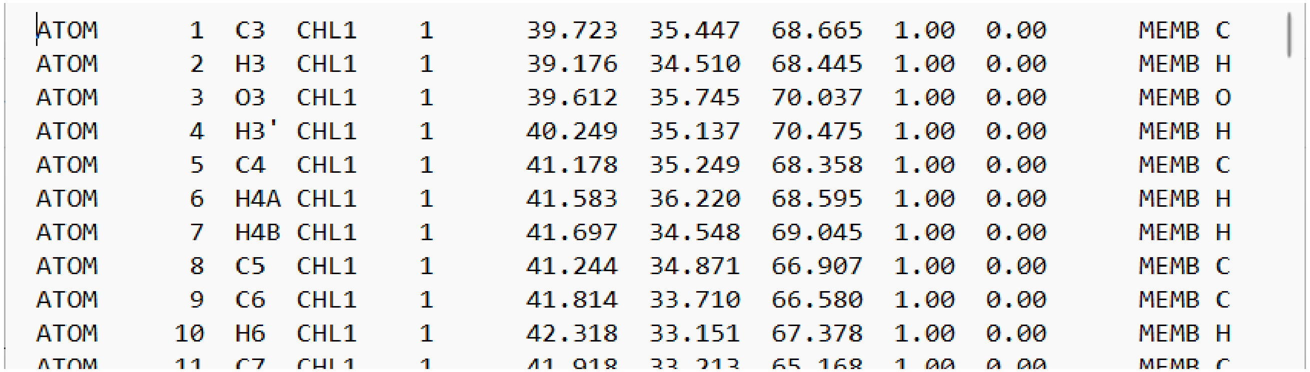
Simulation setup files (coords files – `.gro` or `.pdb`)

`.gro` file



```
Title
51720
    1CHL1     C3    1   3.356   3.369   7.275 -0.2413 -0.6189  0.1702
    1CHL1     H3    2   3.254   3.329   7.262 -0.9629  0.9940  0.5698
    1CHL1     O3    3   3.352   3.424   7.403  0.0148 -0.3600 -0.3277
    1CHL1    H3'    4   3.436   3.448   7.444 -0.8116  1.7065  0.2812
    1CHL1     C4    5   3.451   3.257   7.247  1.4040  0.1508 -0.2478
    1CHL1    H4A    6   3.548   3.297   7.283  2.2210  0.2808 -2.4287
    1CHL1    H4B    7   3.423   3.179   7.320  1.7390  3.4275  3.7507
    1CHL1     C5    8   3.466   3.217   7.101 -0.3609 -0.0233 -1.0451
    1CHL1     C6    9   3.449   3.084   7.071  0.7169 -0.2784 -0.7162
    1CHL1     H6   10   3.442   3.009   7.151  0.3855  1.1698  0.6605
    1CHL1     C7   11   3.444   3.032   6.931 -0.0848 -1.1614 -0.9019
```

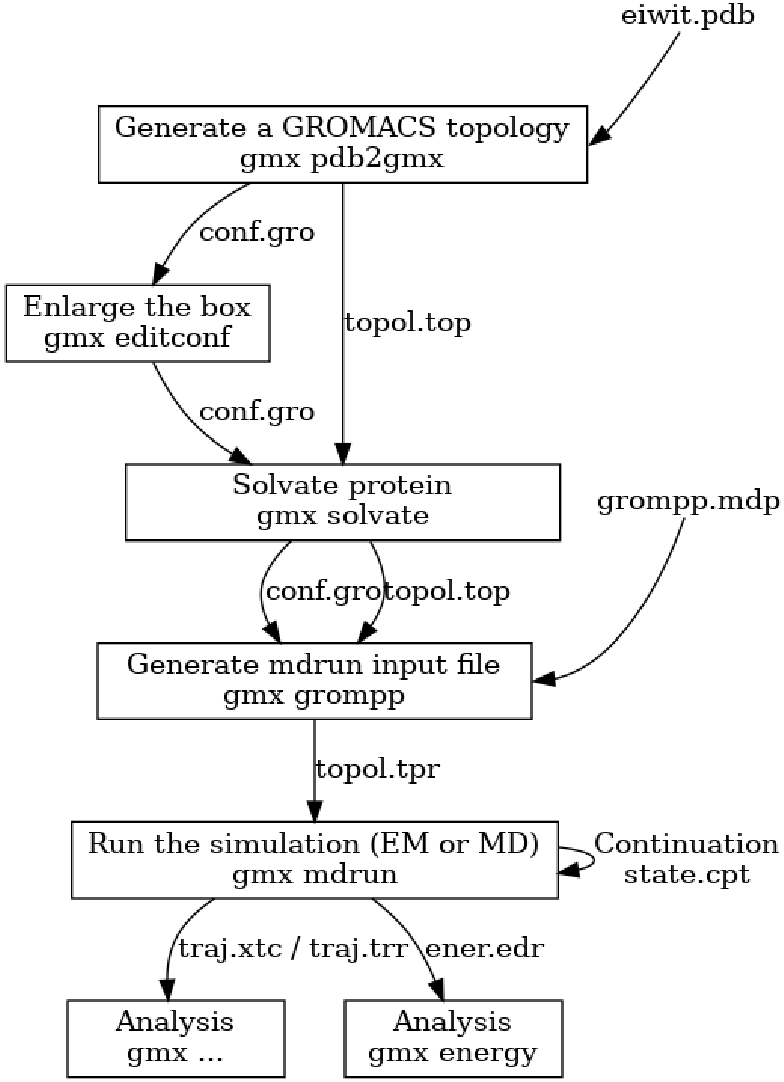
`.pdb` file



```
ATOM      1  C3   CHL1    1      39.723  35.447  68.665  1.00  0.00      MEMB C
ATOM      2  H3   CHL1    1      39.176  34.510  68.445  1.00  0.00      MEMB H
ATOM      3  O3   CHL1    1      39.612  35.745  70.037  1.00  0.00      MEMB O
ATOM      4  H3' CHL1    1      40.249  35.137  70.475  1.00  0.00      MEMB H
ATOM      5  C4   CHL1    1      41.178  35.249  68.358  1.00  0.00      MEMB C
ATOM      6  H4A CHL1    1      41.583  36.220  68.595  1.00  0.00      MEMB H
ATOM      7  H4B CHL1    1      41.697  34.548  69.045  1.00  0.00      MEMB H
ATOM      8  C5   CHL1    1      41.244  34.871  66.907  1.00  0.00      MEMB C
ATOM      9  C6   CHL1    1      41.814  33.710  66.580  1.00  0.00      MEMB C
ATOM     10  H6   CHL1    1      42.318  33.151  67.378  1.00  0.00      MEMB H
ATOM     11  C7   CHL1    1      41.918  33.213  65.168  1.00  0.00      MEMB C
```

Introduction to GROMACS software

Preparing a Simulation (using GROMACS setup tools)



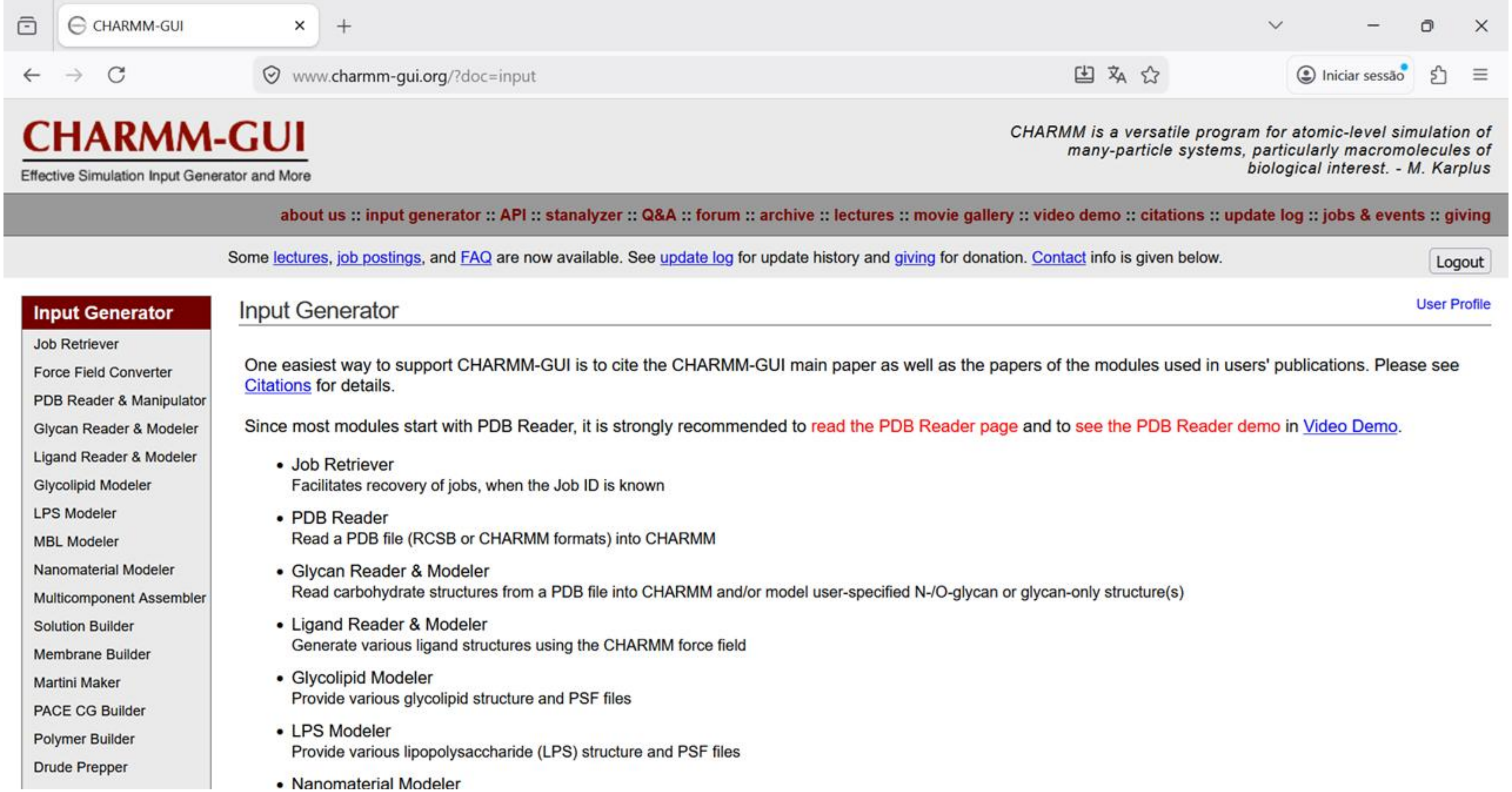
Steps to set up a molecular dynamics simulation

- Preparing the input files: structure file, topology file, parameter file
- Using GROMACS tools for preparation (e.g., pdb2gmx, editconf)

Introduction to GROMACS software

Preparing a Simulation (with the help of other tools)

You can also use other tools to set up a GROMACS MD simulation:



CHARMM-GUI
Effective Simulation Input Generator and More

CHARMM is a versatile program for atomic-level simulation of many-particle systems, particularly macromolecules of biological interest. - M. Karplus

Some lectures, job postings, and FAQ are now available. See update log for update history and giving for donation. Contact info is given below.

Input Generator

One easiest way to support CHARMM-GUI is to cite the CHARMM-GUI main paper as well as the papers of the modules used in users' publications. Please see Citations for details.

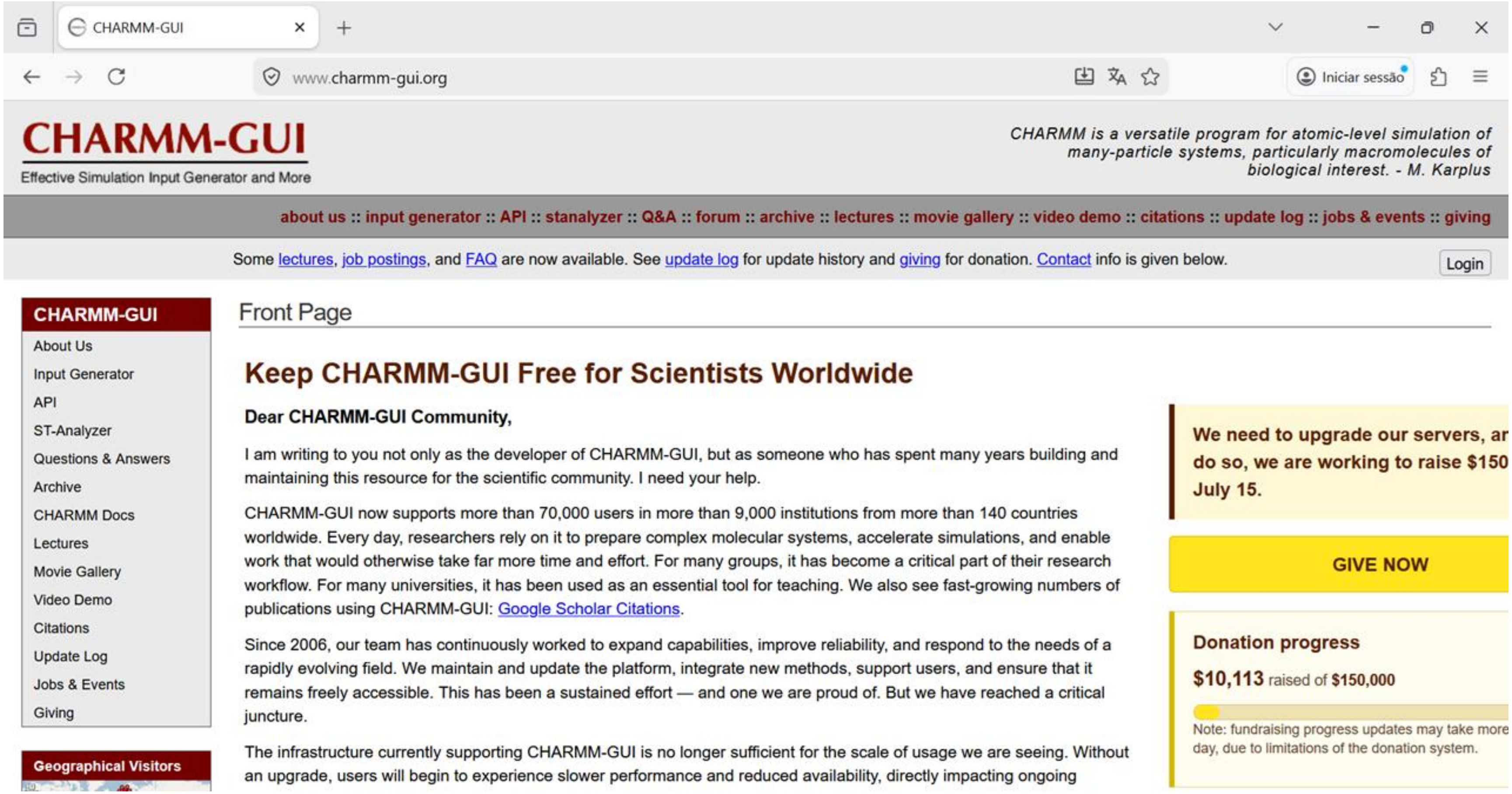
Since most modules start with PDB Reader, it is strongly recommended to read the PDB Reader page and to see the PDB Reader demo in Video Demo.

- Job Retriever
  Facilitates recovery of jobs, when the Job ID is known
- PDB Reader
  Read a PDB file (RCSB or CHARMM formats) into CHARMM
- Glycan Reader & Modeler
  Read carbohydrate structures from a PDB file into CHARMM and/or model user-specified N-/O-glycan or glycan-only structure(s)
- Ligand Reader & Modeler
  Generate various ligand structures using the CHARMM force field
- Glycolipid Modeler
  Provide various glycolipid structure and PSF files
- LPS Modeler
  Provide various lipopolysaccharide (LPS) structure and PSF files
- Nanomaterial Modeler

Introduction to GROMACS software

Preparing a Simulation (with the help of other tools)

You can also use other tools to set up a GROMACS MD simulation:



CHARMM-GUI

Effective Simulation Input Generator and More

CHARMM is a versatile program for atomic-level simulation of many-particle systems, particularly macromolecules of biological interest. - M. Karplus

Some lectures, job postings, and FAQ are now available. See update log for update history and giving for donation. Contact info is given below.

Front Page

Keep CHARMM-GUI Free for Scientists Worldwide

Dear CHARMM-GUI Community,

I am writing to you not only as the developer of CHARMM-GUI, but as someone who has spent many years building and maintaining this resource for the scientific community. I need your help.

CHARMM-GUI now supports more than 70,000 users in more than 9,000 institutions from more than 140 countries worldwide. Every day, researchers rely on it to prepare complex molecular systems, accelerate simulations, and enable work that would otherwise take far more time and effort. For many groups, it has become a critical part of their research workflow. For many universities, it has been used as an essential tool for teaching. We also see fast-growing numbers of publications using CHARMM-GUI: Google Scholar Citations.

Since 2006, our team has continuously worked to expand capabilities, improve reliability, and respond to the needs of a rapidly evolving field. We maintain and update the platform, integrate new methods, support users, and ensure that it remains freely accessible. This has been a sustained effort — and one we are proud of. But we have reached a critical juncture.

The infrastructure currently supporting CHARMM-GUI is no longer sufficient for the scale of usage we are seeing. Without an upgrade, users will begin to experience slower performance and reduced availability, directly impacting ongoing

We need to upgrade our servers, ar do so, we are working to raise $150 July 15.

Donation progress

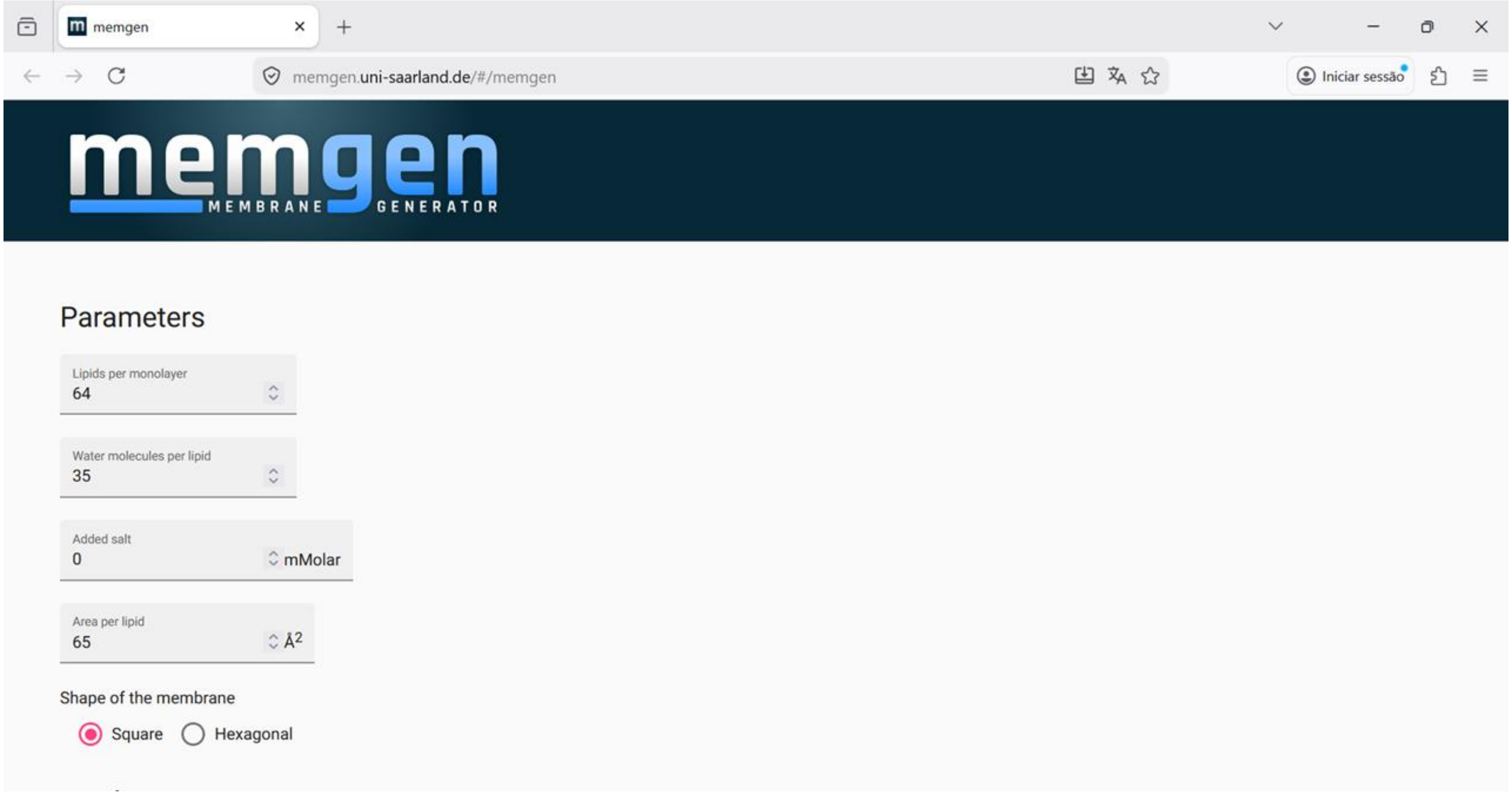
$10,113 raised of **$150,000**

Note: fundraising progress updates may take more day, due to limitations of the donation system.

Introduction to GROMACS software

Preparing a Simulation (with the help of other tools)

You can also use other tools to set up a GROMACS MD simulation:



Parameters

Lipids per monolayer: 64

Water molecules per lipid: 35

Added salt: 0 mMolar

Area per lipid: 65 Å2

Shape of the membrane

● Square ○ Hexagonal

Introduction to GROMACS software

Running simulations

Use of GROMACS involves interaction through a command-line interface

Running GROMACS simulations uses only two commands: `gmx mdrun`, `gmx grompp`

- `gmx grompp`: for preparing the simulation, i.e. reading input files (topology, coordinates, etc.), checking their validity, add additional files and gather it all in just one single file
- `gmx mdrun`: the program that actually performs the simulation

Development of parallel computing capabilities of GROMACS has been quite successful, GROMACS has a very efficient parallel code; it also uses GPU acceleration very efficiently. GROMACS is a good option to perform MD simulations in HPC systems

- details of running simulations in an HPC system vary
- it usually involves submitting simulation jobs to queues
- in Oblivion computer, SLURM queues are used, details later

Introduction to GROMACS software

Analyzing results

After the simulation is performed, the most interesting part is the analysis of the results, which involves several tasks (which mostly can be performed in your own desktop/laptop computer):

- Using GROMACS command-line tools for analysis: `gmx rms`, `gmx trjconv`, `gmx cluster`
- Visualizing results using software like VMD or PyMOL
- Interpreting output data: energy, temperature, pressure, etc.

Running MD Simulations with GROMACS in Oblivion (HPC-UÉvora)

Running MD Simulations with GROMACS

Running in Oblivion

In order to run MD simulations in Oblivion you need to consider a few things:

- SLUMR is used for job submission and management in Oblivion (check SLURM's documentation in **https://slurm.schedmd.com/documentation.html**)
- There's a lot of software installed in Oblivion; for properly setting up the environment for the software you want to use, you must load the appropriate environment modules before using the software
- For information on loading environment modules, using SLURM on Oblivion's queues, and lots of other useful things, you can check **https://oblivion-docs.readthedocs.io/**)

Webpage for Oblivion at **https://oblivion.hpc.uevora.pt/**