NTNU – Trondheim
Norwegian University of
Science and Technology

Molecular Dynamics Simulations in GROMACS

presented by

Jan Schulze

Outline

1. Basic principles of MD simulations

2. MD simulations in GROMACS

   - **Main steps** in performing MD simulations
   - Important **computational aspects**
   - Different **types of input files** and what they mean
   - Different **types of output files** from simulation

   - Some colorful visualizations

Focus (technology)

Remark: Case study not discussed today.

- Same procedure, several times in a row
- Today: Focus on GROMACS technology behind MD simulations

Basic principles of MD simulations

Newton's equations of motion for every atom:

$$\frac{d}{dt}\mathbf{r}_\mathrm{i} = \mathbf{u}_\mathrm{i} \qquad \text{and} \qquad m_i \cdot \frac{d}{dt}\mathbf{u}_\mathrm{i} = \mathbf{f}_\mathrm{i}$$

The force $\mathbf{f}_\mathrm{i}$ is the **gradient** of the potential V :

$$\mathbf{f}_\mathrm{i} = -\frac{\partial}{\partial \mathbf{r}_\mathrm{i}} V\left(\mathbf{r}^\mathrm{N}\right)$$

Contributions to the potential V :

- Intermolecular interactions
  - Lennard-Jones potential (repulsion, van-der-Waals)
  - Coulomb potential (charges)
- Intramolecular interactions
  - Stretching vibrations
  - Bending vibrations
  - Internal rotation

This is how we get molecules!

Force field

- Functional terms (models)
- Model parameters

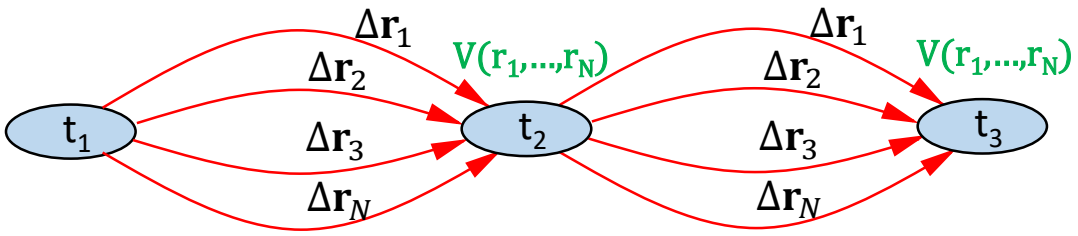
e.g. **O**ptimized **P**otentials for **L**iquid **S**ystems force field

Computational aspects (1)

Parallelization:

Equations of motion for every atom can be solved independently from one time step to the next.

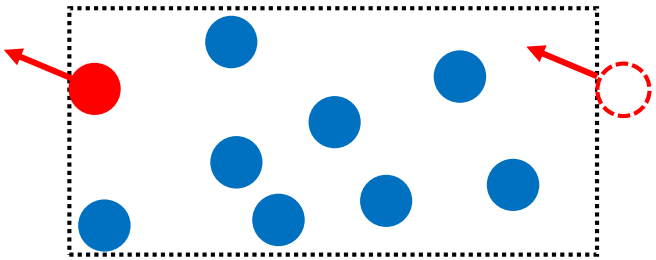
- But: Need all positions together to calculate forces for next step.



Periodic boundary conditions:

Avoid simulation of wall effects in bulk systems.

- Allow atom to “pass” the wall, but relocate it to the opposite wall.
- Atoms can interact through periodic walls as well.
- Leads to bulk-like behavior, even at the walls.



Computational aspects (2)

Controls:

Simulations of practical relevance:

- **NVT simulation** (closed system, heat transfer)
- **NpT simulation** (closed system, heat and work transfer)

Control structures to keep T, p constant:

Barostat: Rescales **atom positions** (box volume) to keep pressure constant (adjust potential energy).

Thermostat: Rescales **atom velocities** to keep temperature constant (adjust kinetic energy).

Equilibration:

Usually, simulations start in non-equilibrium state. Molecular positions and velocities do not correspond with equilibrium.

- First, perform **equilibration simulation** until thermodynamic equilibrium reached.
- Then extend simulation time to simulate actual equilibrium state and apply time-averaging for analysis.

Computational aspects (3)

Time and length scales:

Interactions on a molecular level are very fast and occur on a very small length scale

- Characteristic length scale: Nanometers
- Characteristic time scale: Femtoseconds (10^{-6} ns)

Number of molecules:

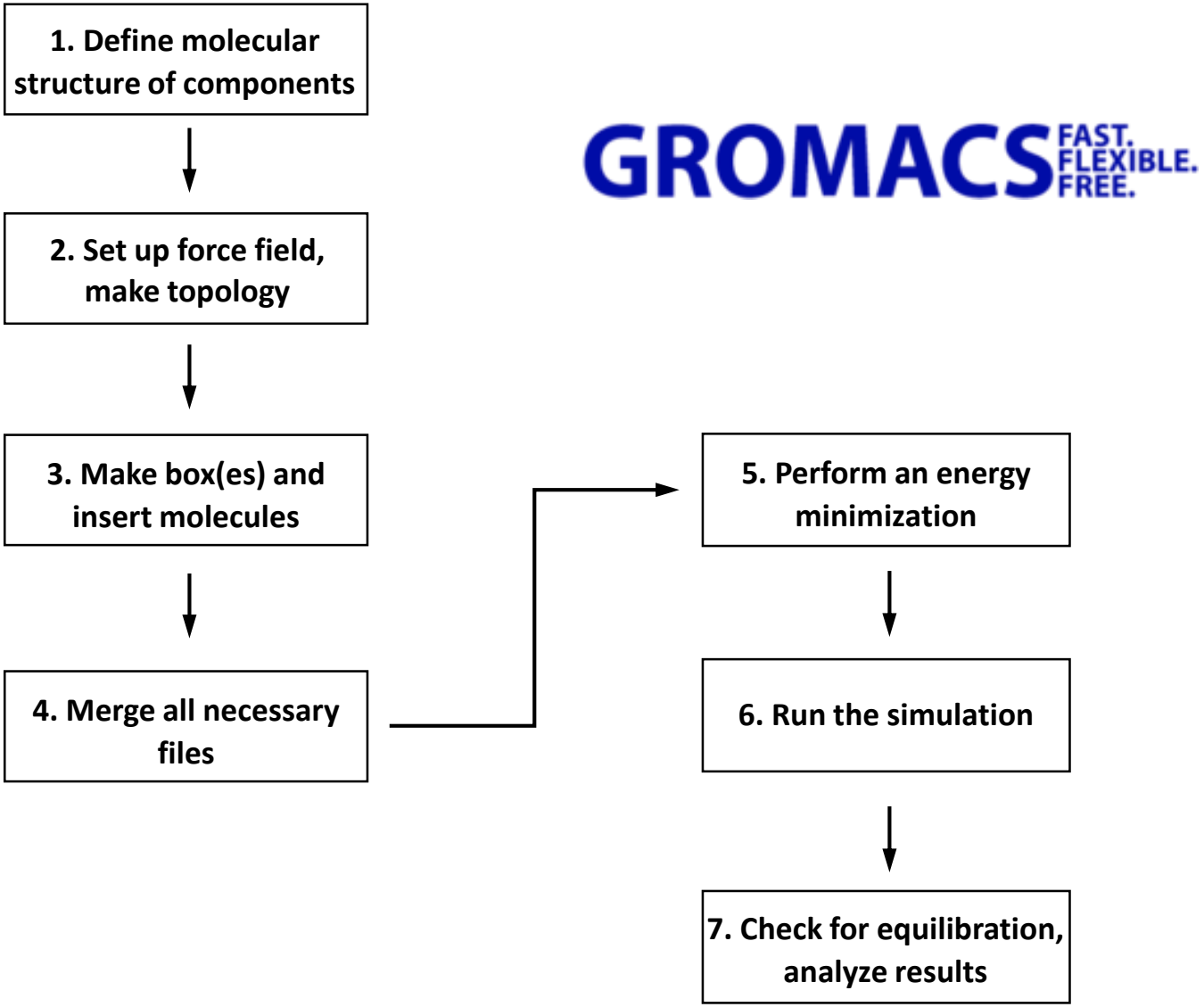
Affects **computational effort** but also the **quality of data**

- **Too small**: High statistical influence in simulation results, low-quality data, results loose on generality
- **Too high**: Simulation takes “forever”

Typical number for pure component: ~ **1000 molecules**

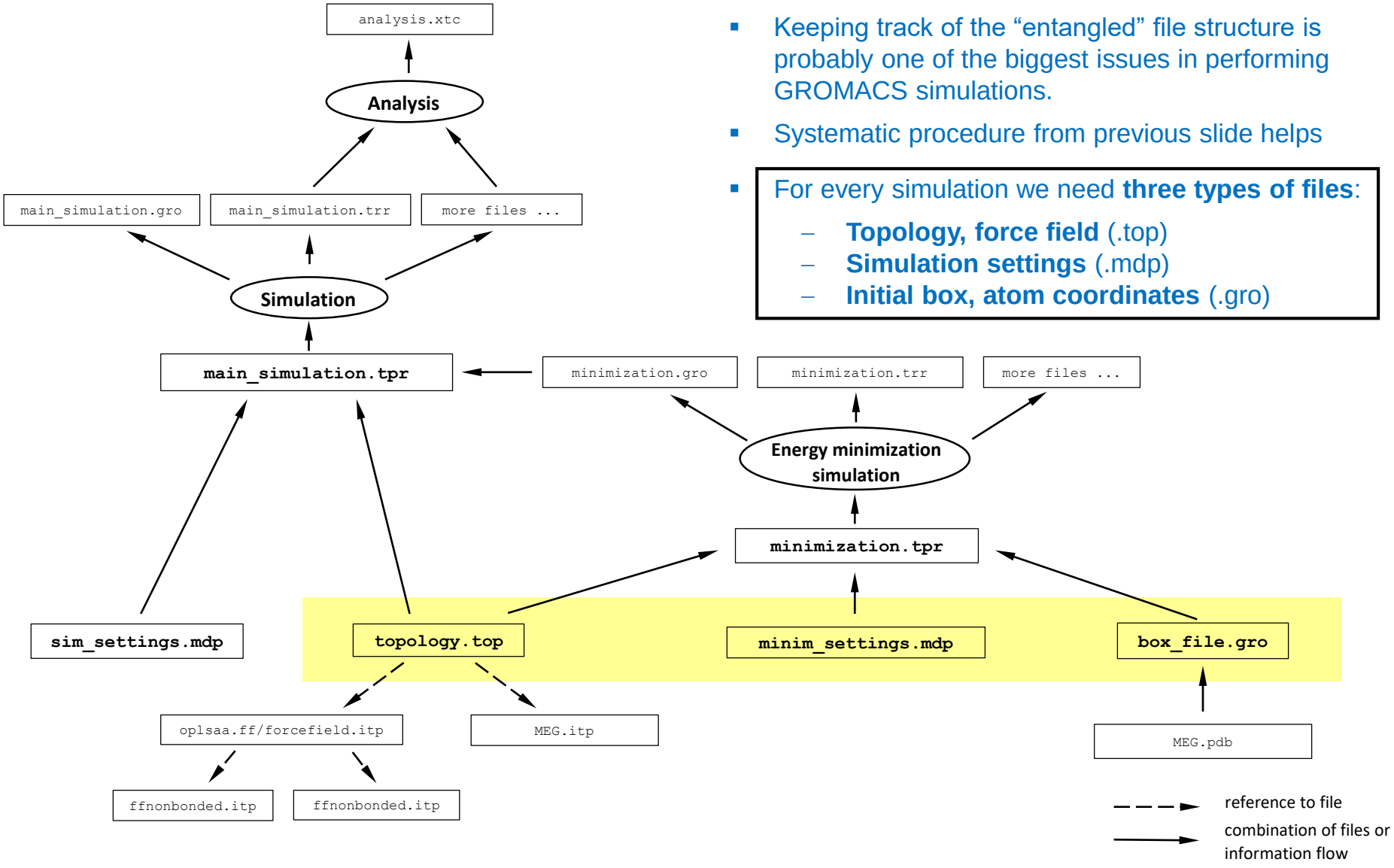
Need **refinement study** to justify choice.

Main steps in a GROMACS simulation



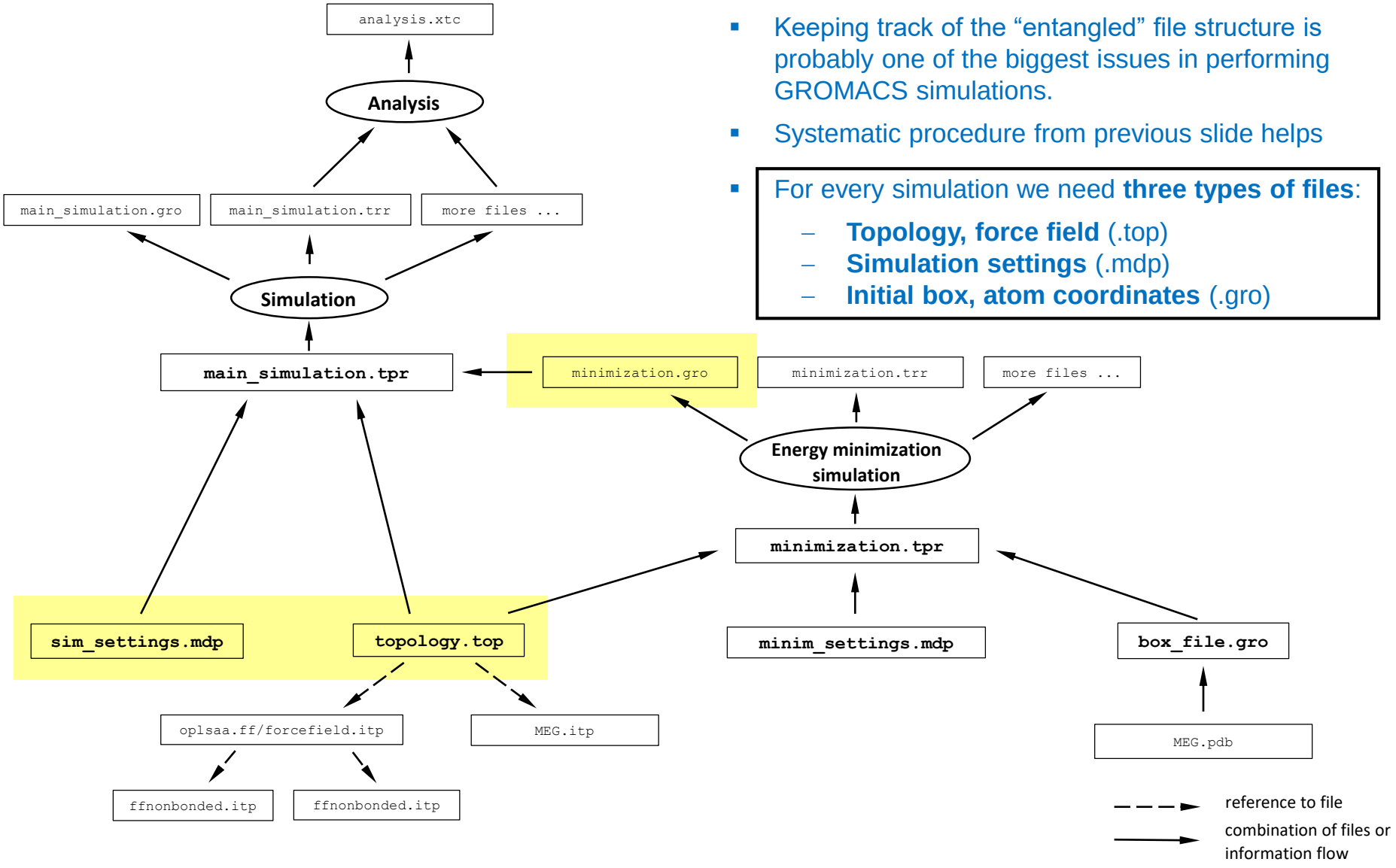
GROMACS FAST. FLEXIBLE. FREE.

File structure and information flow in GROMACS



- Keeping track of the “entangled” file structure is probably one of the biggest issues in performing GROMACS simulations.
- Systematic procedure from previous slide helps
- For every simulation we need **three types of files**:
  - **Topology, force field** (.top)
  - **Simulation settings** (.mdp)
  - **Initial box, atom coordinates** (.gro)

File structure and information flow in GROMACS



- Keeping track of the “entangled” file structure is probably one of the biggest issues in performing GROMACS simulations.
- Systematic procedure from previous slide helps
- For every simulation we need **three types of files**:
  - **Topology, force field** (.top)
  - **Simulation settings** (.mdp)
  - **Initial box, atom coordinates** (.gro)

Step 1 – Define molecular structure of components

```
COMPND    UNNAMED
AUTHOR    GENERATED BY OPEN BABEL 2.3.90
HETATM    1  C   MEG     1       1.263  -0.256   0.128  1.00  0.00           C
HETATM    2  C   MEG     1       0.367  -0.637   1.303  1.00  0.00           C
HETATM    3  O   MEG     1      -0.640   0.357   1.491  1.00  0.00           O
HETATM    4  O   MEG     1       2.270   0.659   0.558  1.00  0.00           O
HETATM    5  H   MEG     1       0.684   0.221  -0.670  1.00  0.00           H
HETATM    6  H   MEG     1       1.771  -1.141  -0.264  1.00  0.00           H
HETATM    7  H   MEG     1       0.946  -0.718   2.228  1.00  0.00           H
HETATM    8  H   MEG     1      -0.141  -1.583   1.100  1.00  0.00           H
HETATM    9  H   MEG     1      -1.088   0.148   2.327  1.00  0.00           H
HETATM   10  H   MEG     1       2.718   0.981  -0.242  1.00  0.00           H
CONECT    1    2    4    5    6
CONECT    2    1    3    7    8
CONECT    3    2    9
CONECT    4    1   10
CONECT    5    1
CONECT    6    1
CONECT    7    2
CONECT    8    2
CONECT    9    3
CONECT   10    4
MASTER        0    0    0    0    0    0    0    0   10    0   10
END
```

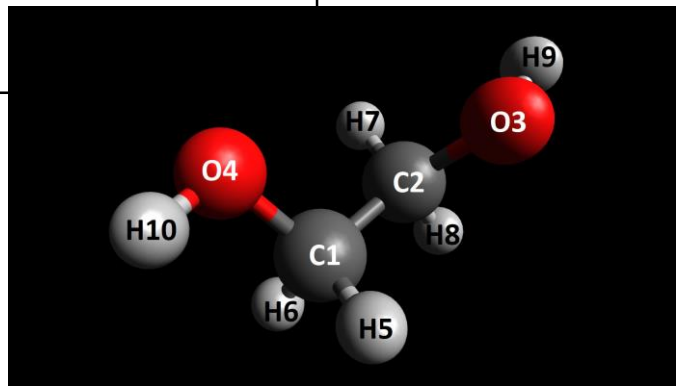
atom positions in the molecule

connectivity

MEG.pdb

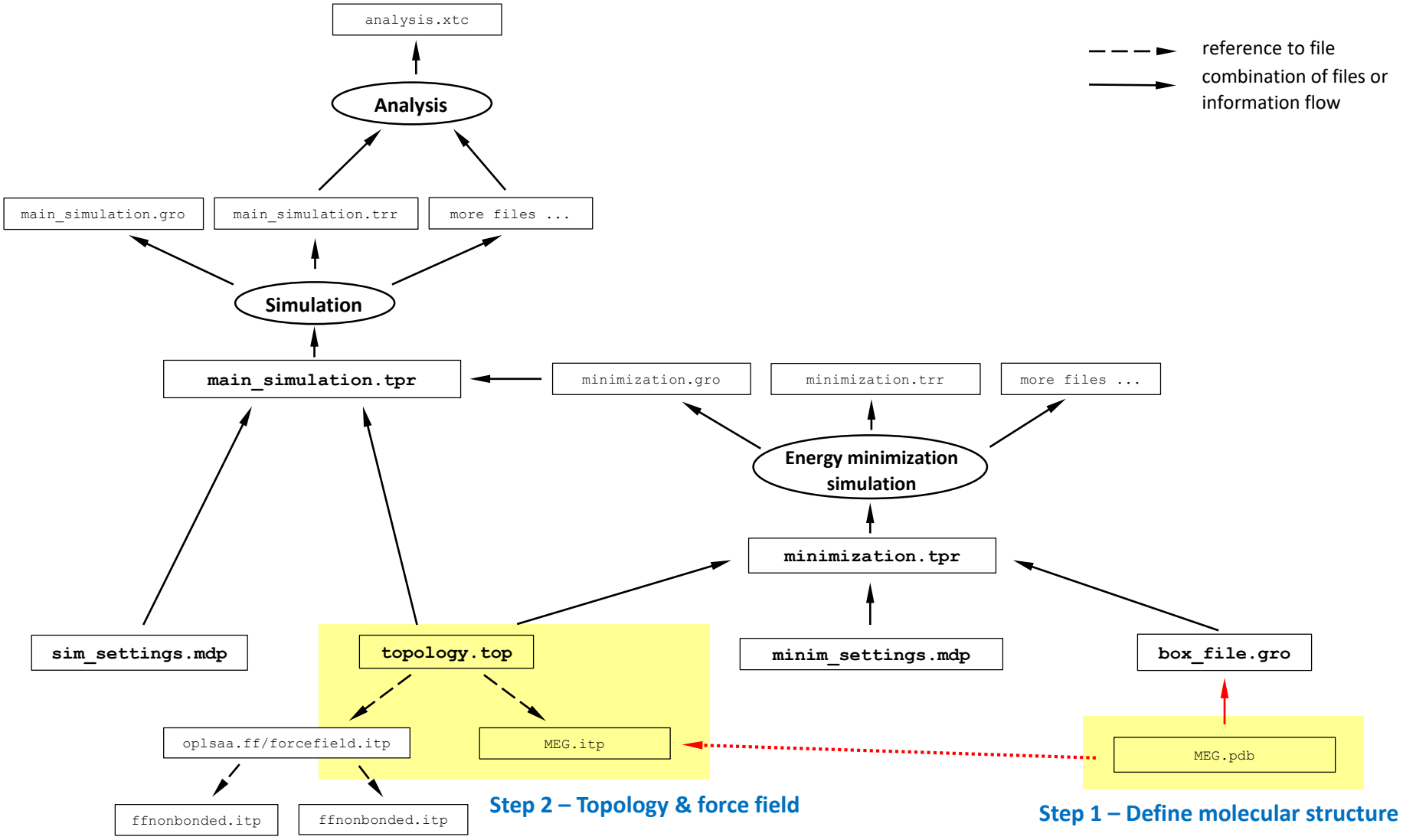
Geometry file (.pdb)

Avogadro



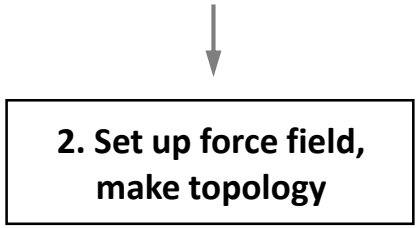
- Simple file structure that contains a lot of important information

File structure and information flow in GROMACS

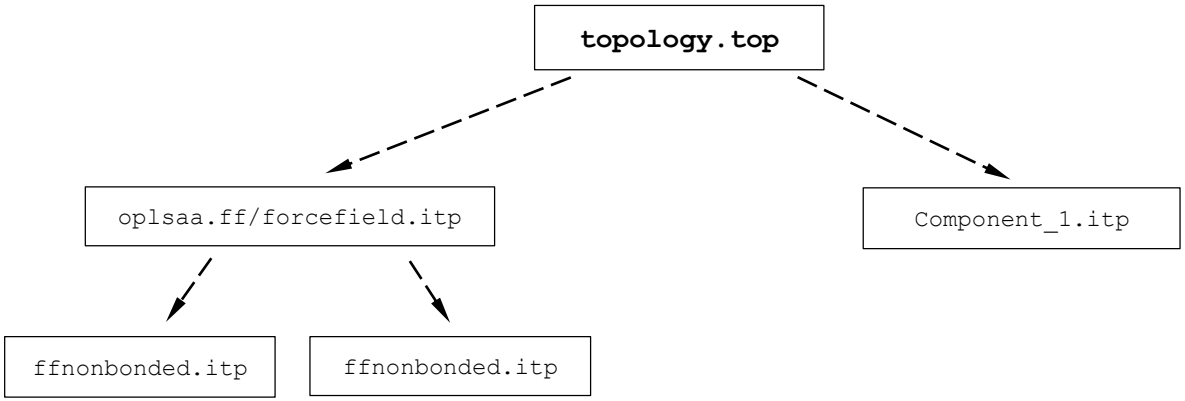


Step 2 – Make topology

1. Define molecular structure of components



2. Set up force field, make topology



What is the topology?

- A set of all **constant attributes** of our system
  - Information about force field
  - Number of molecules

How does it look like?

- Information about force field is structured into multiple sub-files
- User has to set up the force field file(s) for the component(s), e.g. `Component_1.itp`

```
; Force field
#include "oplsaa.ff/forcefield.itp"
#include „Component_1.itp“

[ system ]
; Name
Liquid Simulation

[ molecules ]
; Compound        #mols
MEG                 1000
```

topology.top

Step 2 – Set up force field

1. Define molecular structure of components

↓

2. Set up force field, make topology

The OPLS force field

Set of functional forms

- Implementation of **OPLS functional terms** (and many more) is already done in GROMACS library

Parameters

- Describe how atoms interact with other atoms
- User has to provide parameters that match with particular system
  - Many parameter sets already implemented in GROMACS
  - User (only) has to refer to **correct parameter set** for every atom
  - This is realized as **specification of an *atom type*** for every atom

force field file (.itp)
that we have to create (one for every species)

Functional forms of OPLS force field:

$$V(\mathbf{r}^N) = \sum_{bonds} k_b(r - r_0)^2 + \sum_{angles} k_\theta(\theta - \theta_0)^2$$
$$+ \sum_{torsion} \left\{ \frac{V_1}{2}[1 + \cos(\phi)] + \frac{V_2}{2}[1 - \cos(2\phi)] + \frac{V_3}{2}[1 + \cos(3\phi)] + \frac{V_4}{2}[1 - \cos(4\phi)] \right\}$$
$$+ \sum_i \sum_{j>i} \left\{ V_{ij}^{Coulomb}(r_{ij}) + V_{ij}^{LJ}(r_{ij}) \right\}$$

Step 2 – Set up force field

1. Define molecular structure of components

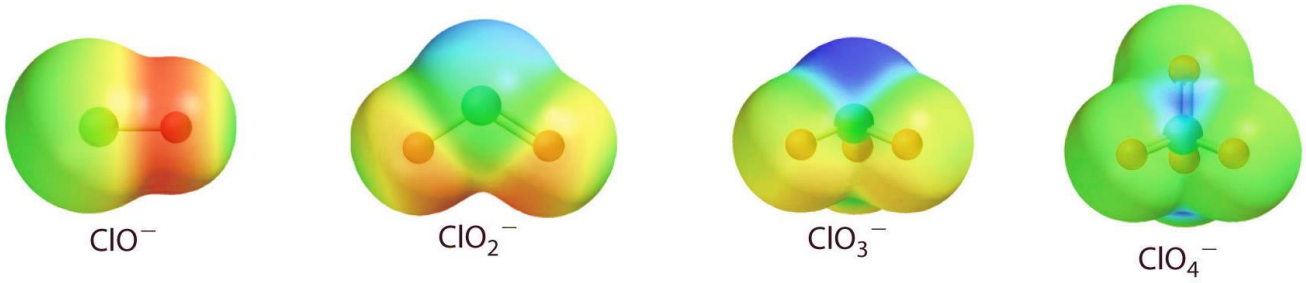
↓

2. Set up force field, make topology

Tasks in step 2

- State **correct *atom type* for every atom** in every molecule
- Generate `.itp` **force field file** for every molecule
- Finally, generate `.top` **topology** for the system

Atom types? Why? – The same element in a different molecule interacts in a different way

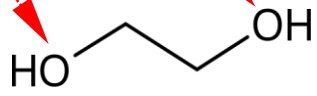


Same element (Cl), but different partial charges

Source: https://saylordotorg.github.io/text_general-chemistry-principles-patterns-and-applications-v1.0/s20-03-molecular-structure-and-acid-b.html

How are *atom types* defined?

```
164  opls_150   12.01100  ; diene =CH-CH=; use #178 for =CR-CR=
165  opls_151   35.45300  ; Cl in alkyl chlorides
166  opls_152   12.01100  ; RCH2Cl in alkyl chlorides
167  opls_153    1.00800  ; H in RCH2Cl in alkyl chlorides
168  opls_154   15.99940  ; all-atom O: mono alcohols
169  opls_155    1.00800  ; all-atom H(O): mono alcohols, OP(=O)2
170  opls_156    1.00800  ; all-atom H(C): methanol
171  opls_157   12.01100  ; all-atom C: CH3 & CH2, alcohols
172  opls_158   12.01100  ; all-atom C: CH, alcohols
173  opls_159   12.01100  ; all-atom C: C, alcohols
174  opls_160   12.01100  ; CH2 Trifluoroethanol
175  opls_161   12.01100  ; CF3 Trifluoroethanol
176  opls_162   15.99940  ; OH  Trifluoroethanol
177  opls_163    1.00800  ; HO  Trifluoroethanol
178  opls_164   18.99840  ; F   Trifluoroethanol
179  opls_165    1.00800  ; H   Trifluoroethanol
180  opls_166   12.01100  ; C(OH) phenol  Use with all
181  opls_167   15.99940  ; O     phenol  atom C, H 145 & 146
182  opls_168    1.00800  ; H     phenol
183  opls_169   15.99940  ; O:    diols
184  opls_170    1.00800  ; H(O): diols
185  opls_171   15.99940  ; O:    triols
186  opls_172    1.00800  ; H(O): triols
187  opls_173   12.01100  ; C(H2OH): triols
188  opls_174   12.01100  ; C(HROH): triols
189  opls_175   12.01100  ; C(R2OH): triols
190  opls_176    1.00800  ; H(CXOH): triols
191  opls_178   12.01100  ; diene =CR-CR=; use #150 for =CH-CH=
```



Ethylene Glycol

***atom type* code**

atomtypes.atp

Step 2 – Generate .itp file

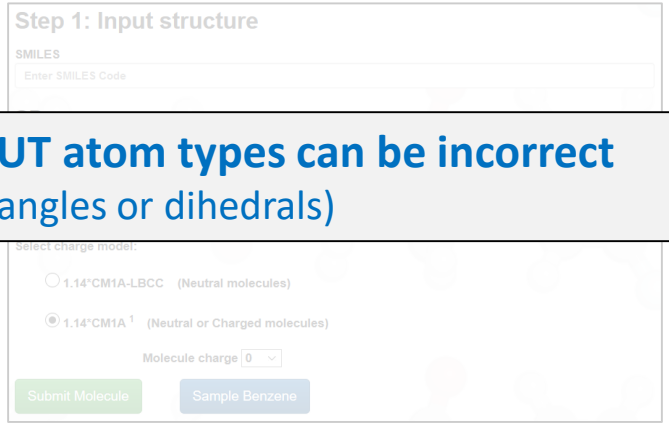
```
[ moleculetype ]
; Name               nrexcl
 MEG                 3
[ atoms ]
;   nr       type  resnr residue  atom   cgnr
     1   opls_800      1     MEG   C00      1
     2   opls_801      1     MEG   C01      1
     3   opls_802      1     MEG   O02      1
     4   opls_803      1     MEG   O03      1
     5   opls_804      1     MEG   H04      1
     6   opls_805      1     MEG   H05      1
     7   opls_806      1     MEG   H06      1
     8   opls_807      1     MEG   H07      1
     9   opls_808      1     MEG   H08      1
    10   opls_809      1     MEG   H09      1
```

```
[ bonds ]
;  i     j     funct
    2     1     1
    3     2     1
    4     1     1
    5     1     1
    6     1     1
    7     2     1
    8     2     1
    9     3     1
   10     4     1
```

```
[ angles ]
;  i     j     k     funct
    1     2     3     1
    2     1     4     1
    2     1     5     1
    2     1     6     1
    1     2     7     1
    1     2     8     1
    2     3     9     1
    1     4    10     1
    4     1     5     1
    5     1     6     1
```

Use a script to generate .itp file

structure from geometry .pdb file



http://zarbi.chem.yale.edu/ligpargen/

Very helpful – BUT atom types can be incorrect
(as well as bonds, angles or dihedrals)

Index sets for bonds, angles, torsion

$$V(\mathbf{r}^N) = \sum_{bonds} k_b(r - r_0)^2 + \sum_{angles} k_\theta(\theta - \theta_0)^2$$
$$+ \sum_{torsion} \left\{ \frac{V_1}{2}[1 + \cos(\phi)] + \frac{V_2}{2}[1 - \cos(2\phi)] + \frac{V_3}{2}[1 + \cos(3\phi)] + \frac{V_4}{2}[1 - \cos(4\phi)] \right\}$$
$$+ \sum_i \sum_{j>i} \left\{ V_{ij}^{Coulomb}(r_{ij}) + V_{ij}^{LJ}(r_{ij}) \right\}$$

MEG.itp

Atom types

1. Define molecular structure of components

↓

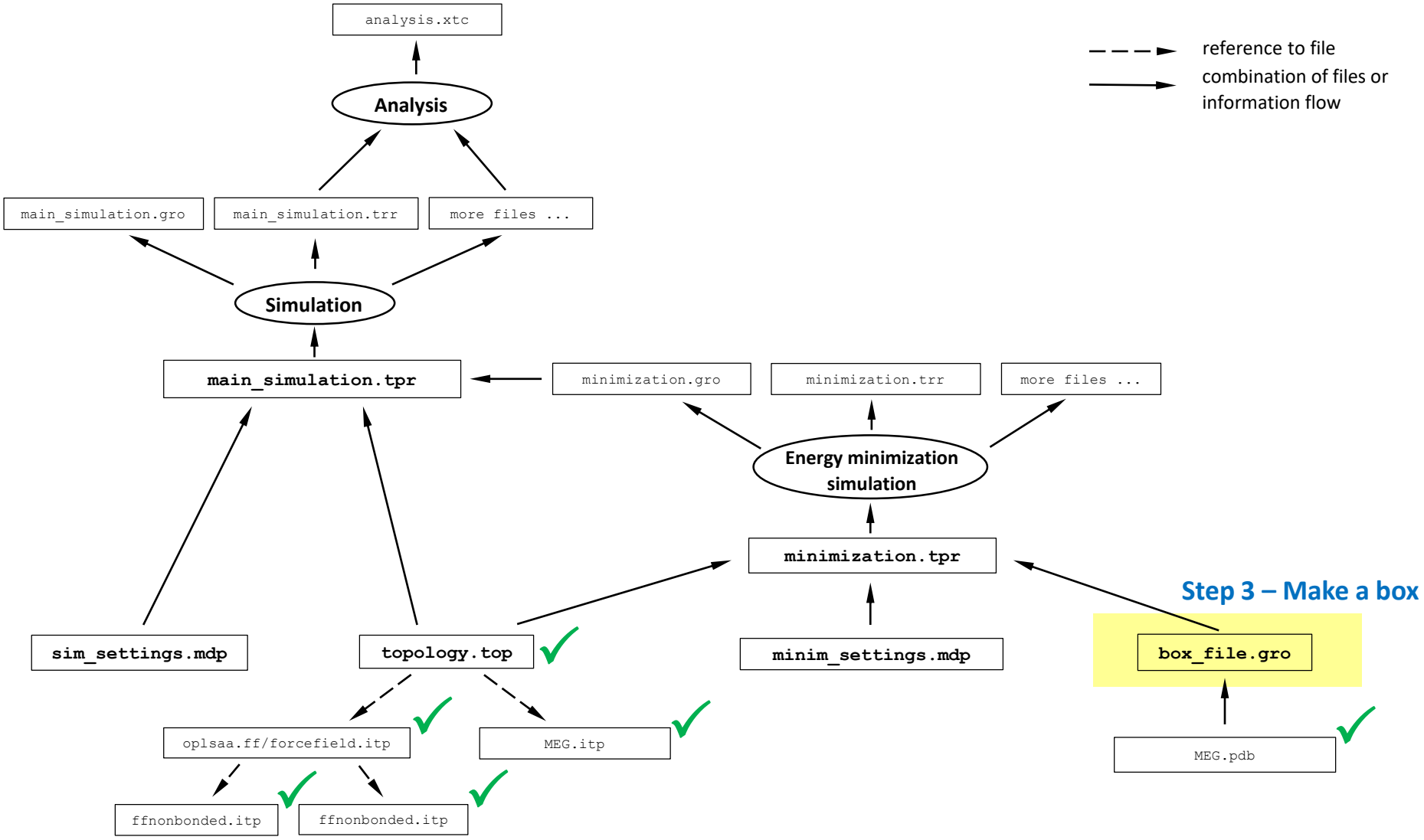
2. Set up force field, make topology

```
164  opls_150   12.01100  ; diene =CH-CH=; use #178 for =CR-CR=
165  opls_151   35.45300  ; Cl in alkyl chlorides
166  opls_152   12.01100  ; RCH2Cl in alkyl chlorides
167  opls_153    1.00800  ; H in RCH2Cl in alkyl chlorides
168  opls_154   15.99940  ; all-atom O: mono alcohols
169  opls_155    1.00800  ; all-atom H(O): mono alcohols, OP(=O)2
170  opls_156    1.00800  ; all-atom H(C): methanol
171  opls_157   12.01100  ; all-atom C: CH3 & CH2, alcohols
172  opls_158   12.01100  ; all-atom C: CH, alcohols
173  opls_159   12.01100  ; all-atom C: C, alcohols
174  opls_160   12.01100  ; CH2 Trifluoroethanol
175  opls_161   12.01100  ; CF3 Trifluoroethanol
176  opls_162   15.99940  ; OH  Trifluoroethanol
177  opls_163    1.00800  ; HO  Trifluoroethanol
178  opls_164   18.99840  ; F   Trifluoroethanol
179  opls_165    1.00800  ; H   Trifluoroethanol
180  opls_166   12.01100  ; C(OH) phenol  Use with all
181  opls_167   15.99940  ; O     phenol  atom C, H 145 & 146
182  opls_168    1.00800  ; H     phenol
183  opls_169   15.99940  ; O:    diols
184  opls_170    1.00800  ; H(O): diols
185  opls_171   15.99940  ; O:    triols
186  opls_172    1.00800  ; H(O): triols
187  opls_173   12.01100  ; C(H2OH): triols
188  opls_174   12.01100  ; C(HROH): triols
189  opls_175   12.01100  ; C(R2OH): triols
190  opls_176    1.00800  ; H(CXOH): triols
191  opls_178   12.01100  ; diene =CR-CR=; use #150 for =CH-CH=
```

How to check and select correct atom types?

- The OPLS force field provides a **list of all parametrized atom types** + description
- **Read through this list** and find atom types that apply on your molecular structure
- Conduct **literature research** and see what other authors have used

File structure and information flow in GROMACS



MD skills – Generation of a liquid phase

- **High density of liquid** poses a problem for insertion algorithms
- **Atom shells overlap** frequently, causing the molecule insertion algorithm to fail
- How to choose the box dimensions then?

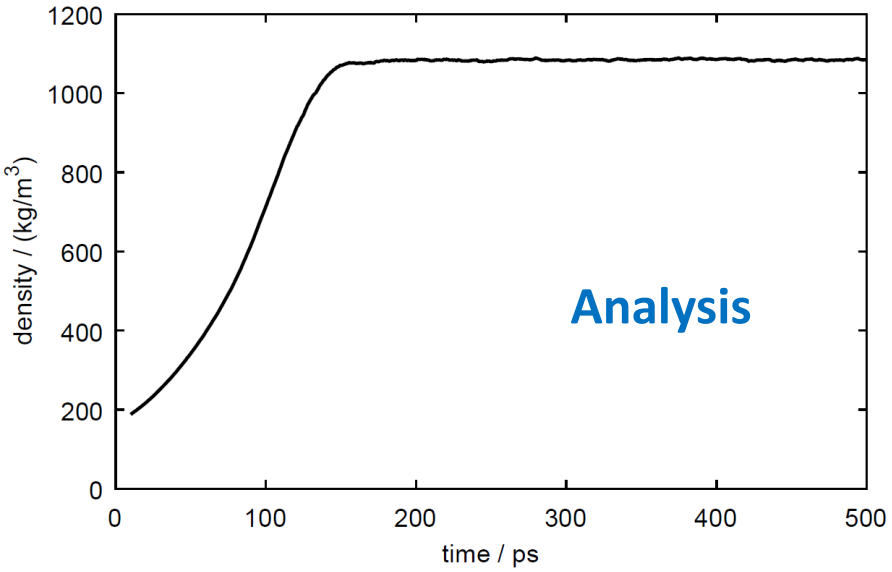
```
Try 1 success (now 10 atoms)!
Try 3 success (now 20 atoms)!
Try 5 success (now 30 atoms)!
Try 7 success (now 40 atoms)!
Try 9 success (now 50 atoms)!
Try 19 success (now 60 atoms)!
Try 29 success (now 70 atoms)!
Try 100
Added 7 molecules (out of 10 requested)
Writing generated configuration to minibox.gro

Output configuration contains 70 atoms in 7 residues
```

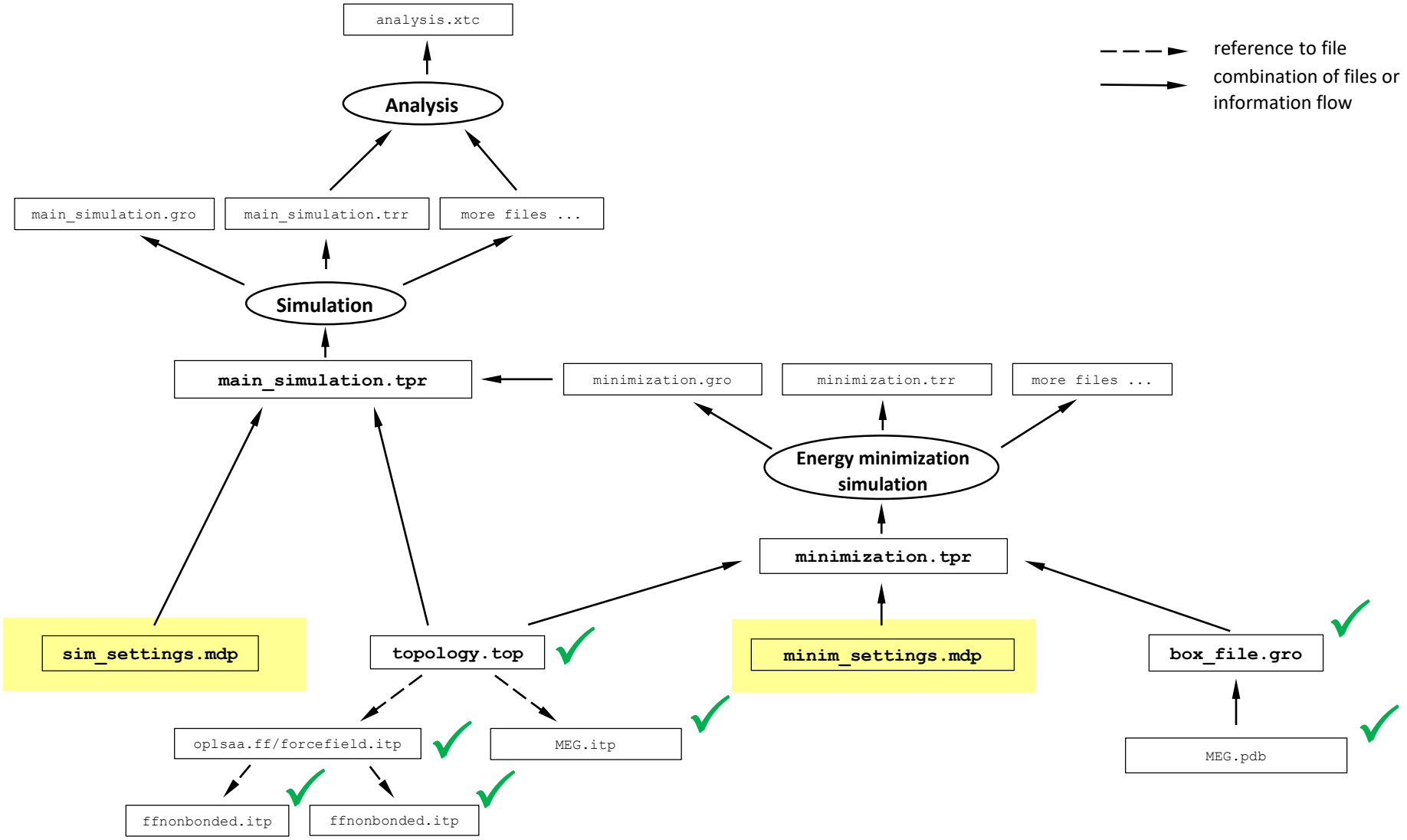
Box dimensions chosen according to realistic density

<u>Strategy for liquid box:</u>

- Insert molecules into an **oversized box** (factor ~ 10)
- Shrink the box dimensions in an **NpT simulation**
- Simulation until **equilibration** at realistic density reached (equilibration)



File structure and information flow in GROMACS



http://manual.gromacs.org/documentation/5.1/user-guide/mdp-options.html

Simulation settings – Example: NpT simulation

```
o VARIOUS PREPROCESSING OPTIONS =
title                    = NPT
cpp                      = /lib/cpp
include                  = -I../top
define                   =
```

MD integrator

```
; RUN CONTROL PARAMETERS =
integrator               = md
; start time and timestep in ps =
tinit                    = 0
dt                       = 0.0001
nsteps                   = 100000
; number of steps for center of mass motion removal =
nstcomm                  = 100
```

data logging (output files)

```
; OUTPUT CONTROL OPTIONS =
; Output frequency for x, v and f=
nstxout                  = 1000
nstvout                  = 0
nstfout                  = 0
; Output frequency for energies =
nstlog                   = 1000
nstenergy                = 1000
; Output frequency and precision for xtc file =
nstxtcout                = 50000
xtc-precision            = 1000
```

cutoff radius

```
; NEIGHBORSEARCHING PARAMETERS =
cutoff-scheme            = Verlet
nstlist                  = 10
; ns algorithm (simple or grid) =
ns_type                  = grid
; nblist cut-off         =
rlist                    = 1.0
```

periodic boundaries

```
; Periodic boundary conditions: xyz or none =
pbc                      = xyz
...
```

```
...
```

cutoff radius

```
; OPTIONS FOR ELECTROSTATICS AND VDW =
; Method for doing electrostatics =
coulombtype              = PME
rcoulomb                 = 1.0
; Method for doing Van der Waals =
vdw-type                 = PME
rvdw                     = 1.0
; Apply long range dispersion corrections for Energy and Pressure =
DispCorr                 = EnerPres
; Spacing for the PME/PPPM FFT grid =
fourierspacing           = 0.12
; EWALD/PME/PPPM parameters =
pme_order                = 4
ewald_rtol               = 1e-05
epsilon_surface          = 0
optimize_fft             = no
```

thermostat

```
; OPTIONS FOR WEAK COUPLING ALGORITHMS =
; Temperature coupling   =
tcoupl                   = V-rescale
; Groups to couple separately =
tc-grps                  = system
;Time constant (ps) and reference temperature (K) =
tau_t                    = 0.1
ref_t                    = 300
```

barostat

```
; Pressure coupling      =
Pcoupl                   = Berendsen
Pcoupltype               = semiisotropic
; Time constant (ps), compressibility (1/bar) and reference P (bar)
=
tau_p                    = 1.0
compressibility          = 0 4.5e-5
ref_p                    = 1.0 1000.0
```

initial velocities

```
; GENERATE VELOCITIES FOR STARTUP RUN =
gen_vel                  = yes
gen_temp                 = 300
gen_seed                 = -1
```

http://manual.gromacs.org/documentation/5.1/user-guide/mdp-options.html

Simulation settings – Example: NpT simulation

```
o VARIOUS PREPROCESSING OPTIONS =
title                    = NPT
cpp                      = /lib/cpp
include                  = -I../top
define                   =

; RUN CONTROL PARAMETERS =
integrator               = md
; start time and timestep in ps =
tinit                    = 0
dt                       = 0.0001
nsteps
; numbe
nstcomm

; OUTPU
; Outpu
nstxout
nstvout
nstfout
; Outpu
nstlog
nstener
; Outpu
nstxtco
xtc-pre

; NEIGHBORSEARCHING PARAMETERS =
cutoff-scheme            = Verlet
nstlist                  = 10
; ns algorithm (simple or grid) =
ns_type                  = grid
; nblist cut-off         =
rlist                    = 1.0

; Periodic boundary conditions: xyz or none =
pbc                      = xyz
```

MD integrator

cutoff radius

periodic boundaries

```
...

; OPTIONS FOR ELECTROSTATICS AND VDW =
; Method for doing electrostatics =
coulombtype              = PME
rcoulomb                 = 1.0
; Method for doing Van der Waals =
vdw-type                 = PME
rvdw                     = 1.0
; Apply long range dispersion corrections for Energy and Pressure =
DispCorr                 = EnerPres

; Pressure coupling      =
Pcoupl                   = Berendsen
Pcoupltype               = semiisotropic
; Time constant (ps), compressibility (1/bar) and reference P (bar)
=
tau_p                    = 1.0
compressibility          = 0 4.5e-5
ref_p                    = 1.0 1000.0

; GENERATE VELOCITIES FOR STARTUP RUN =
gen_vel                  = yes
gen_temp                 = 300
gen_seed                 = -1
```

cutoff radius

rmostat

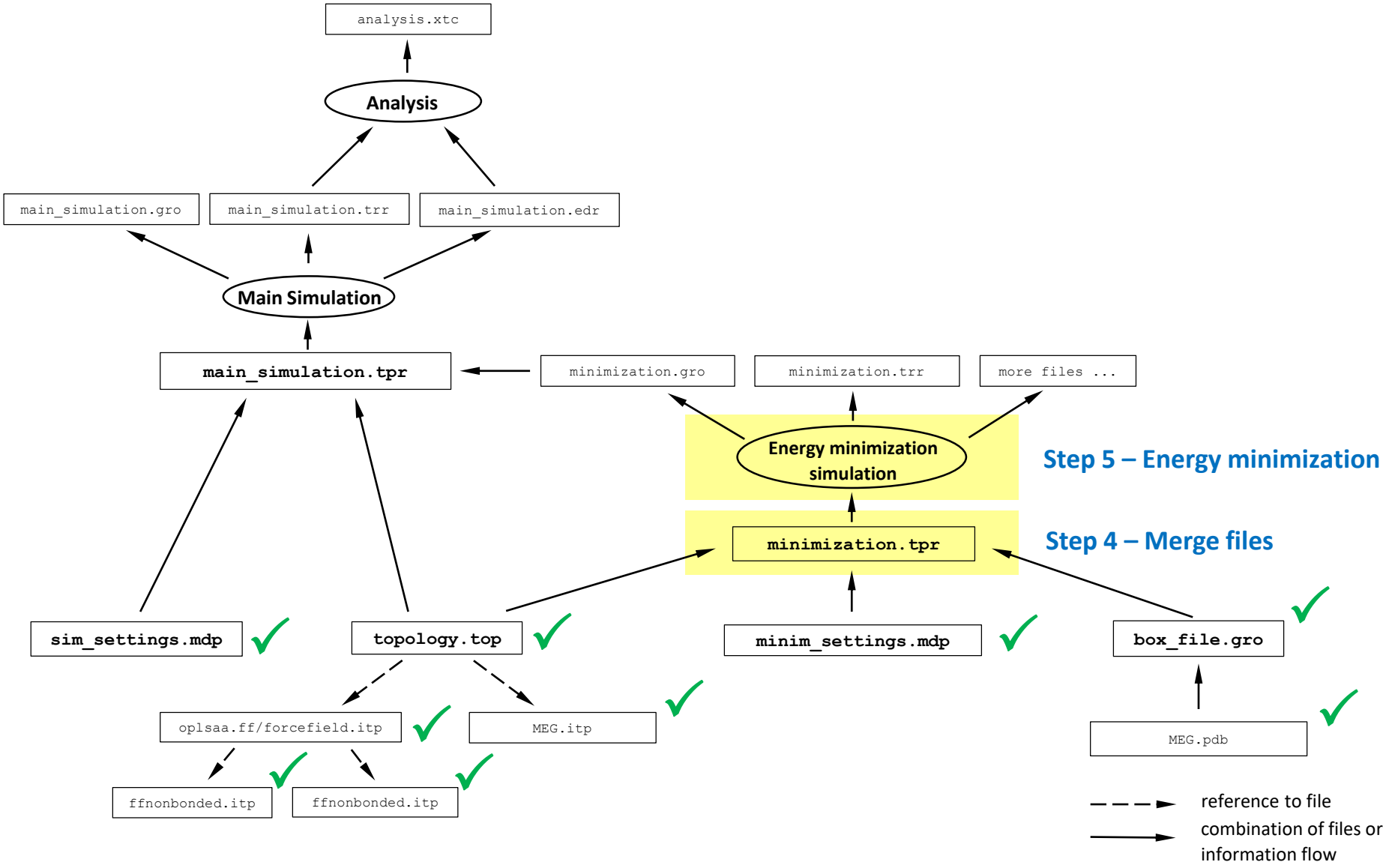
barostat

initial velocities

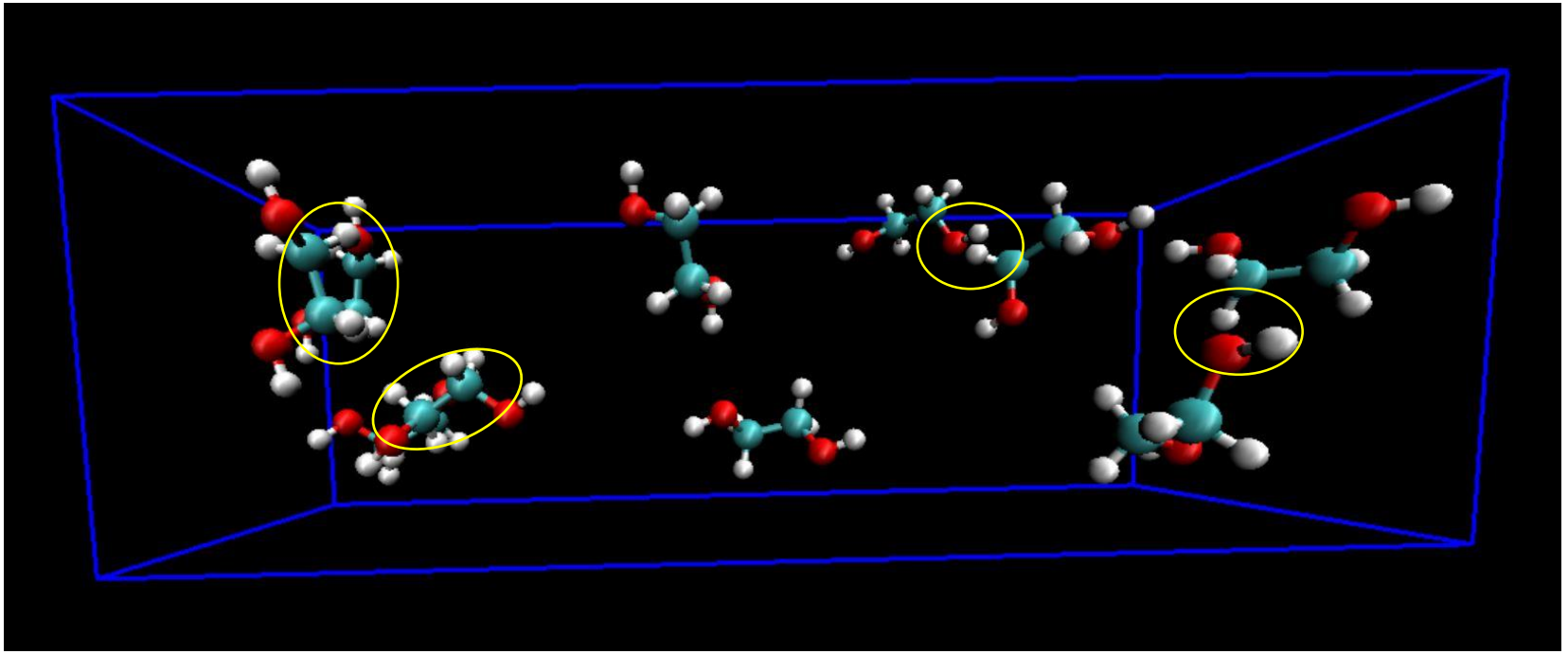
Bottom line

- Great number of simulation parameters
- Need to go deep into theory to understand them all
- Setting up a completely new simulation is a task for a “professional”
- For simulations of typical chemical engineering systems, you usually do not have to adjust many of them. Allows for using a template like this one.

File structure and information flow in GROMACS

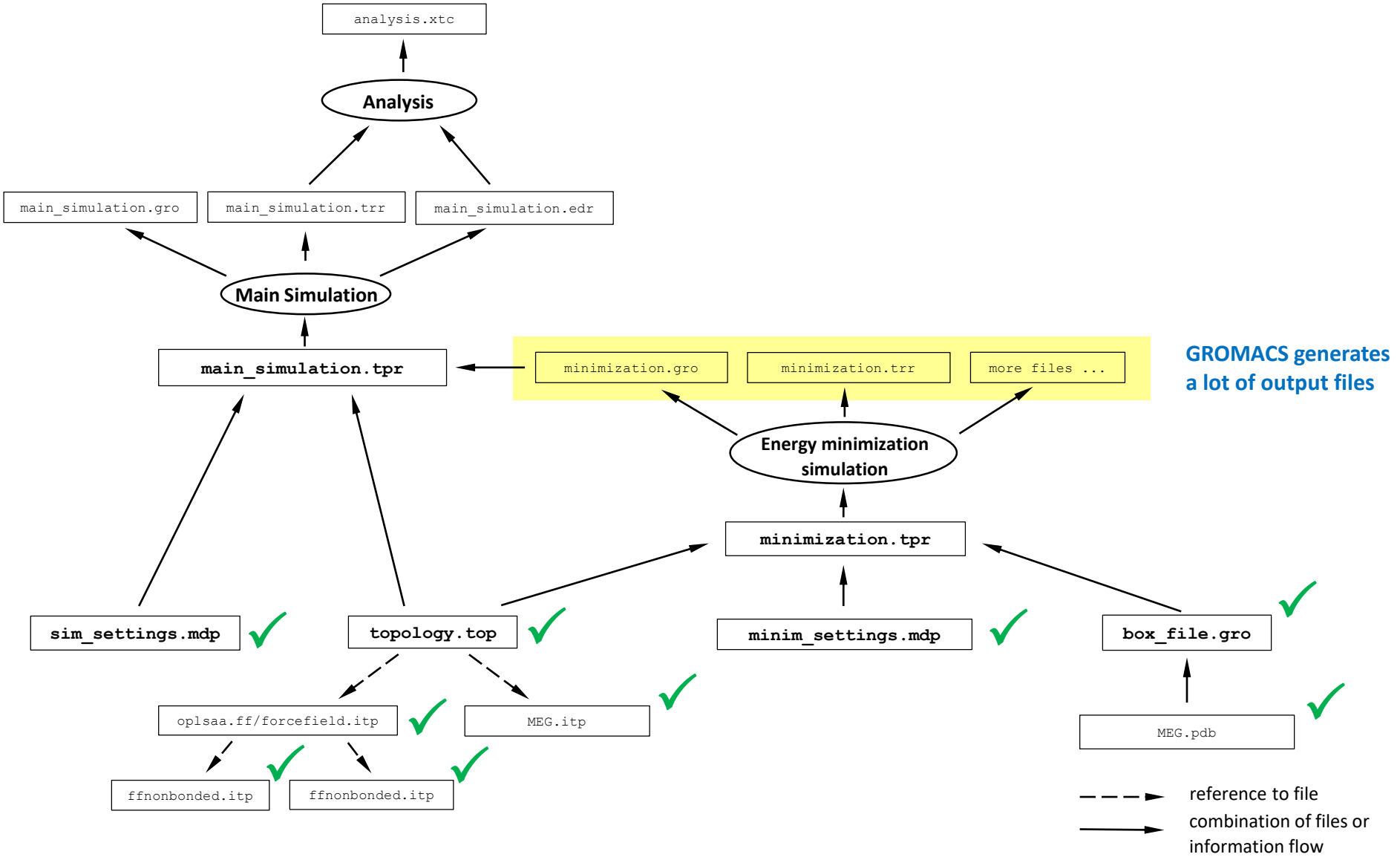


Step 5 – Energy minimization



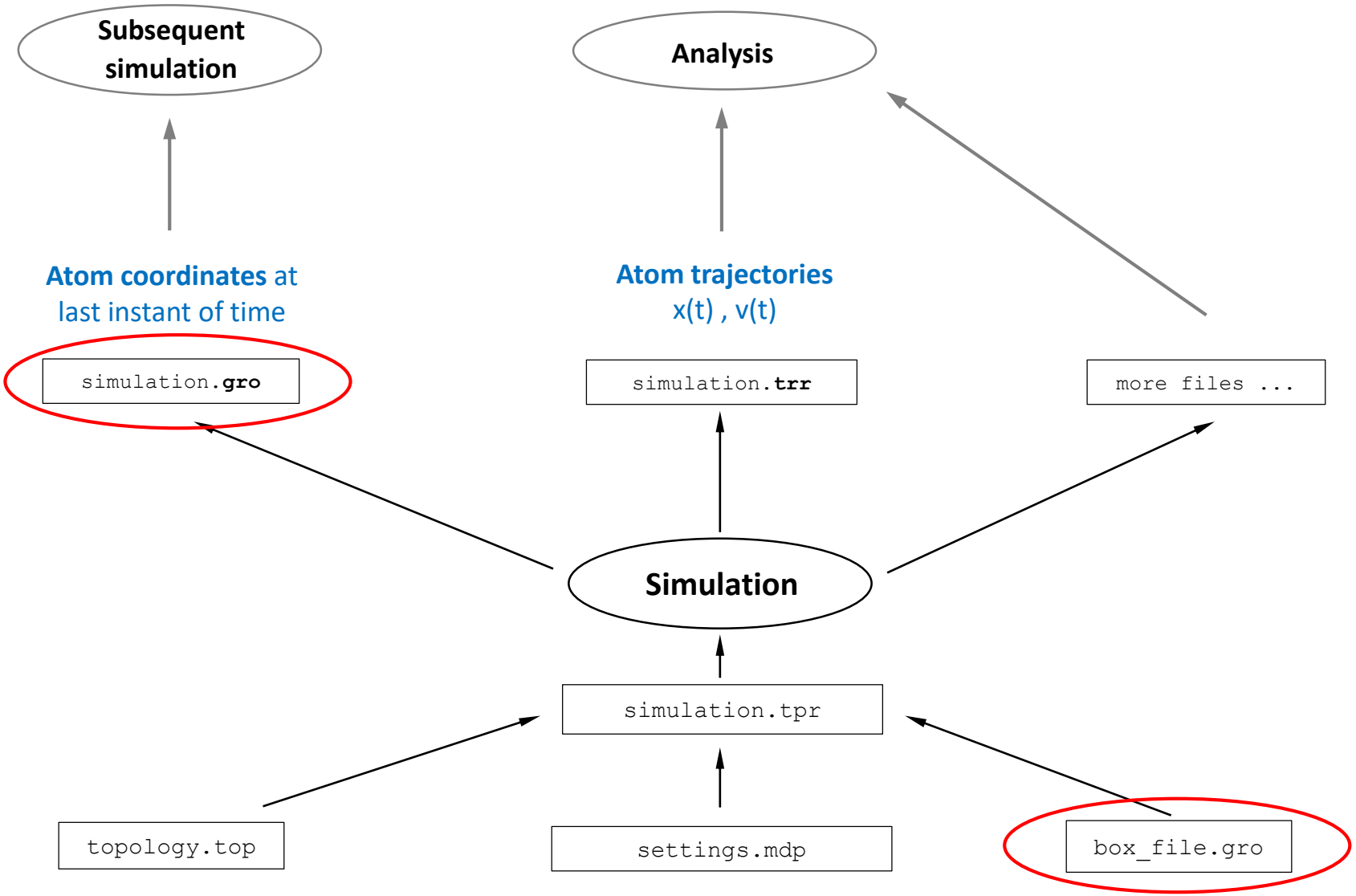
- Molecules **randomly inserted** by GROMACS, strong overlapping atom shells avoided
- Arrangement of molecules **can still be problematic**
- Motivates energy minimization – “Allow molecules to make themselves comfortable.”
- **Avoids “kicks”** and thereby errors at beginning of simulation, i.e. strong repulsive forces

File structure and information flow in GROMACS

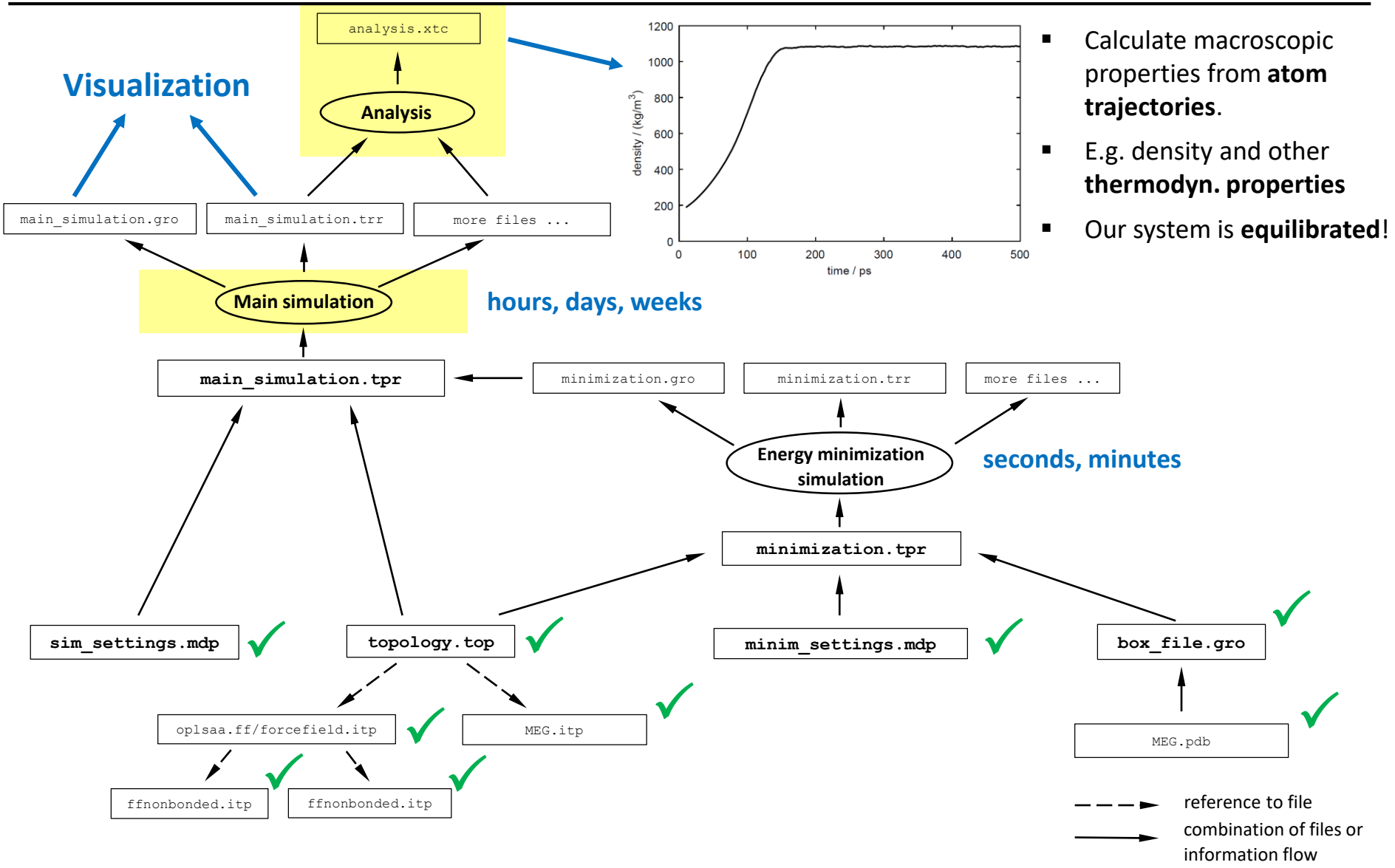


GROMACS generates a lot of output files

Important output files

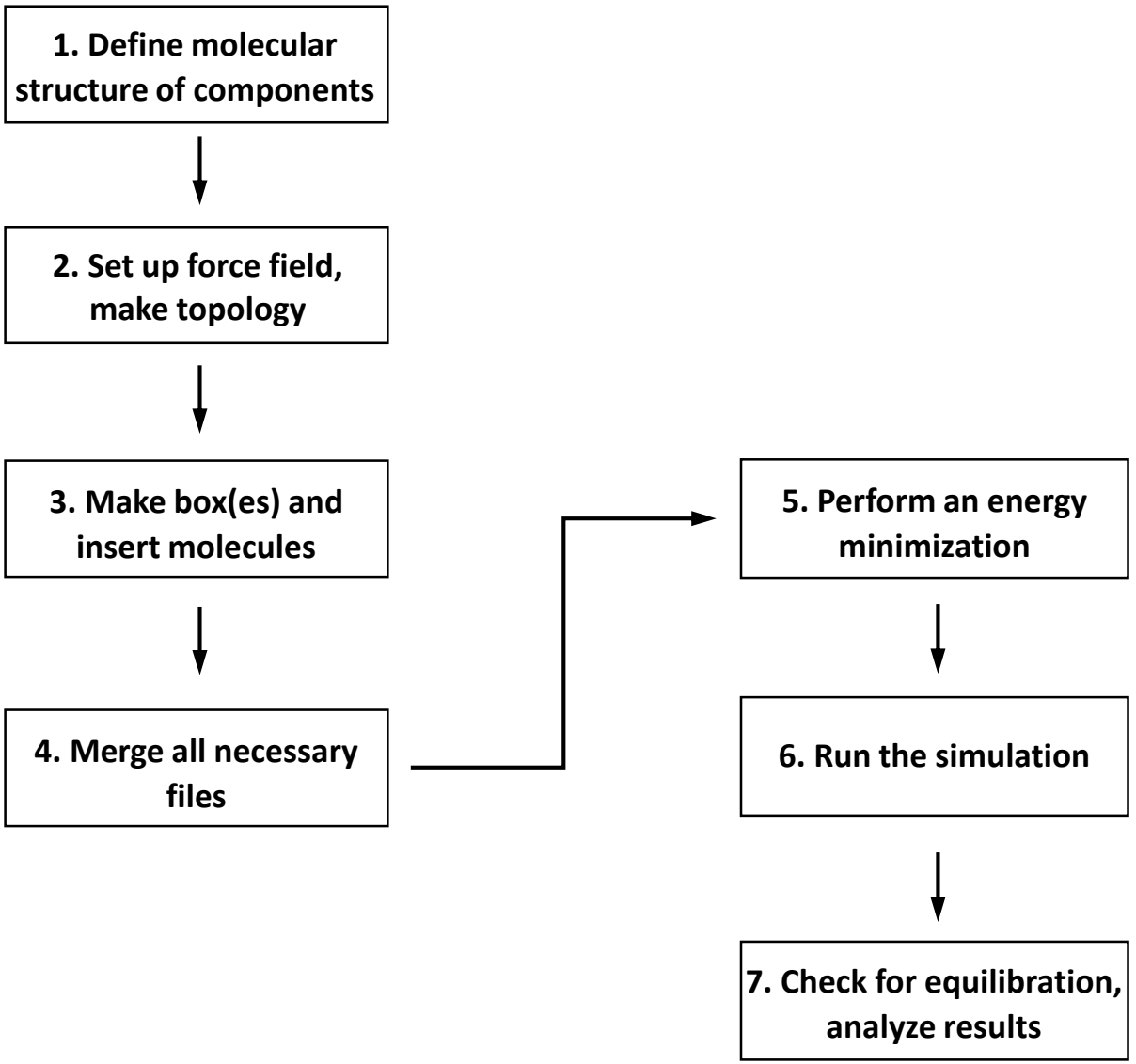


File structure and information flow in GROMACS



- Calculate macroscopic properties from **atom trajectories**.
- E.g. density and other **thermodyn. properties**
- Our system is **equilibrated**!

Main steps in a GROMACS simulation

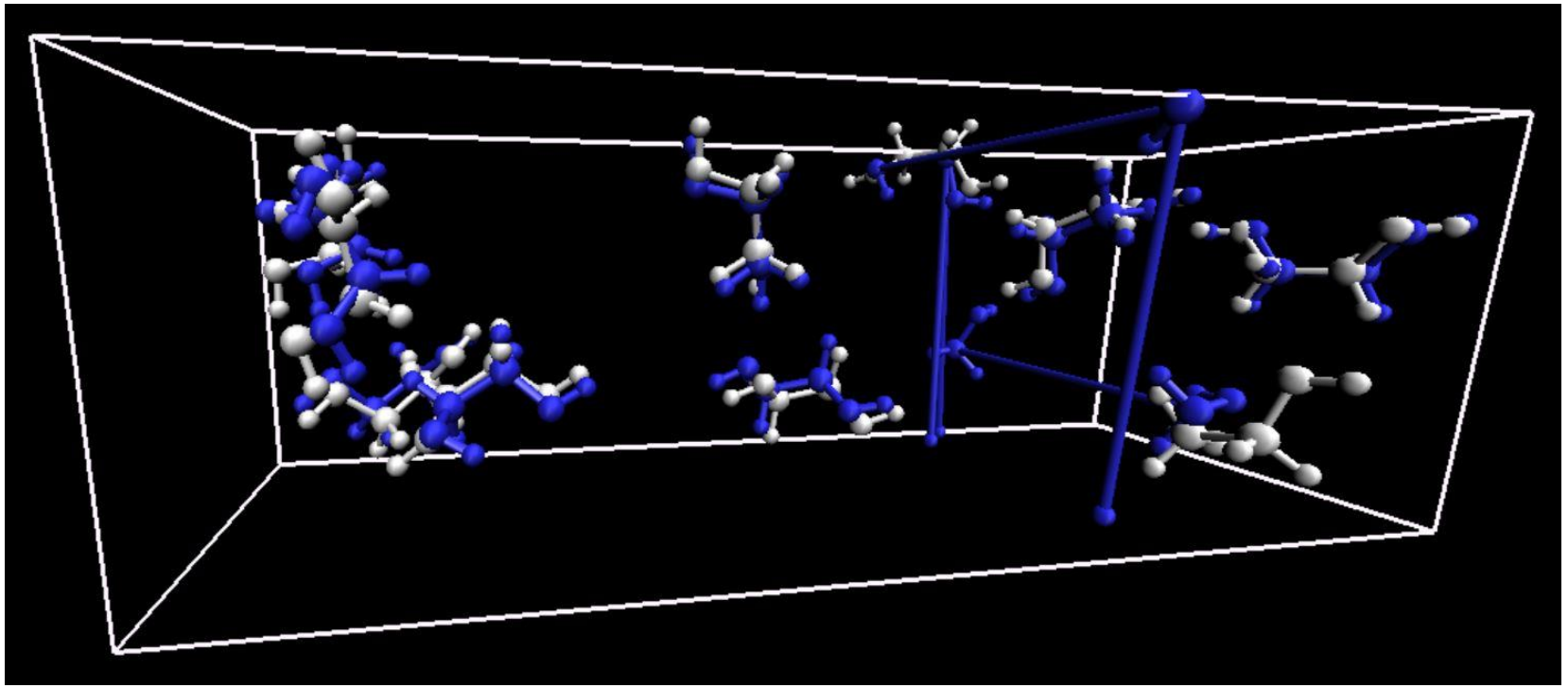


End

Thanks for your attention!

Questions?

Step 5 – Energy minimization



White: Before energy minimization

Blue: After energy minimization (you can see periodic boundary conditions for two molecules)

- **Potential energy can be reduced** up to factor 10^4 depending on system
- **Avoids "kicks"** and thereby errors at beginning of simulation, i.e. strong repulsive forces

Thermostat: Berendsen coupling

- Equations of motion for velocities and positions of the atoms are not only influenced by interactions but also coupled with control algorithms for pressure and temperature.

- Implement a temperature control that realizes a **first order behavior**:

$$\frac{dT}{dt} = \frac{T_0 - T}{\tau}$$

with set-point temperature T_0 and time constant τ (tuning parameter).

- "Strength of coupling", i.e. aggressiveness of temperature controller can be varied through τ.
  - high for fast equilibration
  - small during equilibrium simulation subsequent to equilibration process

- Time-dependent temperature coupling factor (velocity rescaling factor):

$$\lambda(t) = \underbrace{\left[1 + \frac{n_{TC} \cdot \Delta t}{\tau_T} \cdot \left(\frac{T_0}{T(t) \cdot (t - 0.5 \cdot \Delta t)} - 1\right)\right]^{1/2}}_{\begin{array}{l}> 1, \text{ if } T < T_0 \\ < 1, \text{ if } T > T_0\end{array}} \quad \Rightarrow \quad \boldsymbol{v}\left(t + \frac{\Delta t}{2}\right) = \lambda \cdot \left[\boldsymbol{v}\left(t - \frac{\Delta t}{2}\right) + \frac{\mathbf{F}(\mathrm{t})}{\mathrm{m}} \cdot \Delta t\right]$$