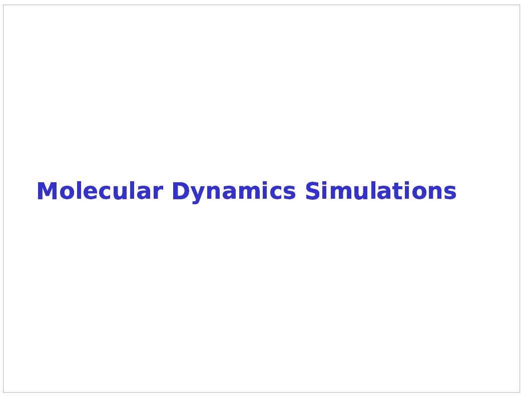
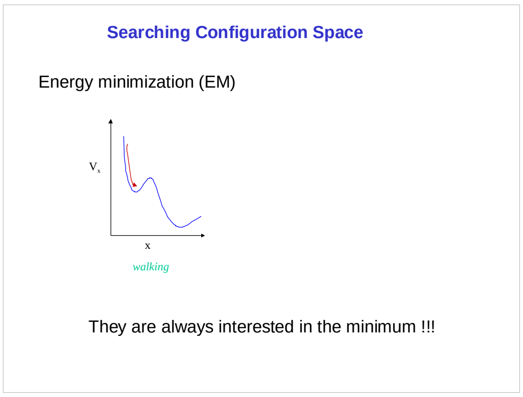


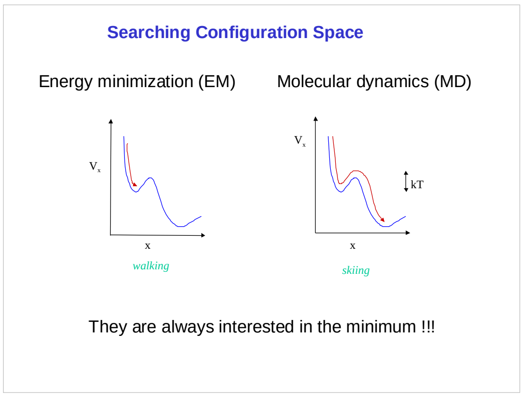
# Molecular Dynamics Simulations



## Searching Configuration Space

Energy minimization (EM)

They are always interested in the minimum !!!

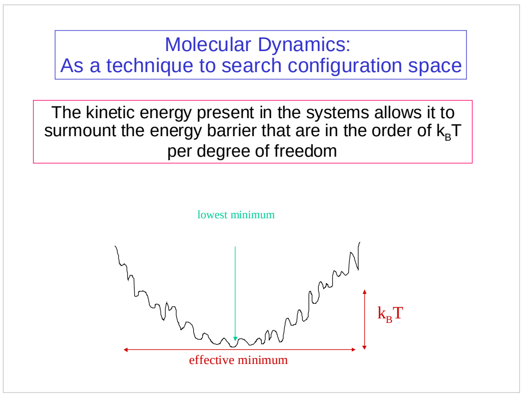


## Searching Configuration Space

Energy minimization (EM)

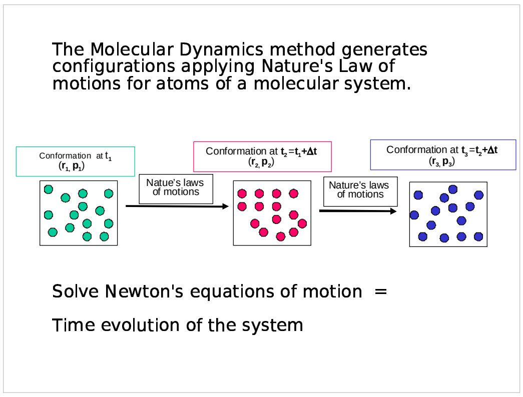
Molecular dynamics (MD)

They are always interested in the minimum !!!



## Molecular Dynamics: As a technique to search configuration space

The kinetic energy present in the systems allows it to surmount the energy barrier that are in the order of $k_BT$ per degree of freedom



The Molecular Dynamics method generates configurations applying Nature's Law of motions for atoms of a molecular system.

Conformation at $t_1$ $(\mathbf{r}_1, \mathbf{p}_1)$

Natue's laws of motions

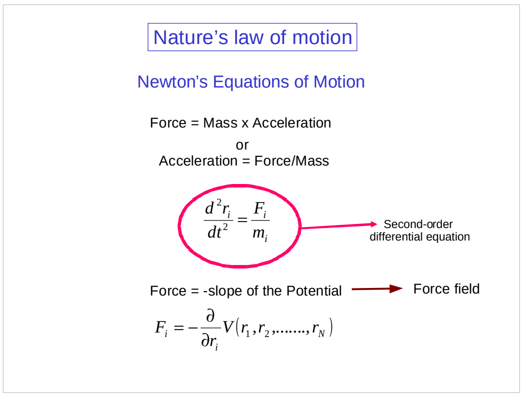
Conformation at $\mathbf{t}_2=\mathbf{t}_1+\Delta\mathbf{t}$ $(\mathbf{r}_2, \mathbf{p}_2)$

Nature's laws of motions

Conformation at $\mathbf{t}_3=\mathbf{t}_2+\Delta\mathbf{t}$ $(\mathbf{r}_3, \mathbf{p}_3)$

Solve Newton's equations of motion =

Time evolution of the system



## Nature’s law of motion

Newton’s Equations of Motion

Force = Mass x Acceleration

or

Acceleration = Force/Mass

$$\frac{d^2 r_i}{dt^2} = \frac{F_i}{m_i}$$

Second-order differential equation

Force = -slope of the Potential → Force field

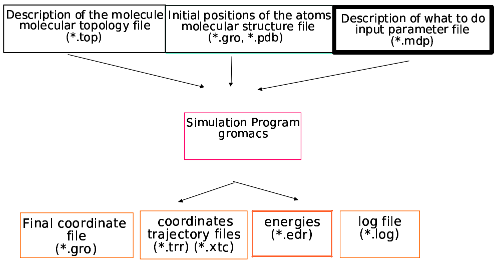
$$F_i = -\frac{\partial}{\partial r_i} V\left(r_1, r_2, \ldots\ldots, r_N\right)$$

## What governs the outcome of a MD simulation?

- starting configuration
- choice of degrees of freedom
- force field parameters
- **boundary conditions**
- **treatment of non-bonded interactions**

- **integration time step**
- **treatment of temperature and pressure**

## Running a MD simulation


Description of the molecule
molecular topology file
(*.top)

Initial positions of the atoms
molecular structure file
(*.gro, *.pdb)

Description of what to do
input parameter file
(*.mdp)

Simulation Program
gromacs

Final coordinate file
(*.gro)

coordinates
trajectory files
(*.trr) (*.xtc)

energies
(*.edr)

log file
(*.log)

**Keywords for non-bonded interactions**

**and periodic boundary conditions in *.mdp file**

```
;Interactions
coulombtype = cut-off  ;treatment of electrostatic interactions
rcoulomb  = 1.0          ;long range electrostatic cutoff
vdwtype       = cut-off  ;treatment of van der Waals interactions
rvdw          = 1.0      ;long range van der Waals cutoff
;the neighbor list
nstlist        = 10  ;Frequency to update the list.
ns_type    = grid ;Method to determine list (simple, grid).
rlist          = 1.0   ;cutoff for making list (short range forces).

;Periodic Boundary Conditions
pbc      = xyz                  ;Periodic Boundary Conditions
```

## Algorithms for Molecular Dynamics

Equation
*(What do we have to solve)*

$$\frac{d^2 r_i}{dt^2} = \frac{F_i}{m_i} \rightarrow \boxed{\text{Second-order differential equation}}$$

Newton's Equations of Motion

Solution: simple algorithms to integrate it in small time steps

For example: the leap-frog integrator

*no derivative of force*

$$r_i(\mathbf{t}+\mathbf{\Delta t}) = r_i(\mathrm{t}) + v_i(\mathrm{t}+1/2\Delta\mathrm{t})\,\Delta t + O(\Delta t^3)$$

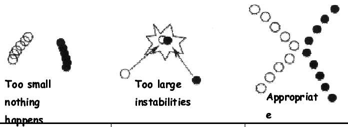
$$v_i(\mathbf{t}+1/2\mathbf{\Delta t}) = v_i(\mathrm{t}-1/2\Delta\mathrm{t}) + m^{-1} F(r_i(\mathrm{t}))\Delta t + O(\Delta t^3)$$

time

time step

## Choosing integration time step

- What is an appropriate time step for the integration?

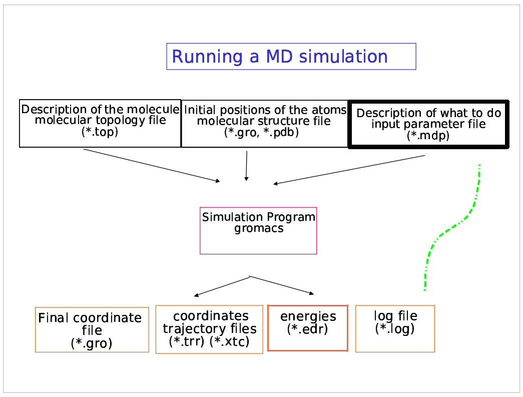


| System | Types of motion | Suggested time step |
|---|---|---|
| Rigid molecules | translation, rotation | 5 fs |
| Flexible molecules, **rigid bonds** | translation, rotation, torsions | 2 fs |
| Flexible molecules, flexible bonds | Translation, rotation, torsions, vibrations | 1 or 0.5 fs |

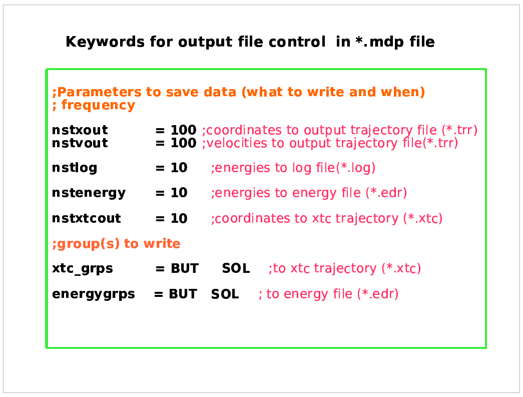
**Keywords for MD algorithms in *.mdp file**

```
title     = MD simulation of .....
;Perform an MD simulation for 20 ps (example)
integrator = md      ;Algorithm (md = molecular dynamics)
tinit              = 0.0  ; starting time in picoseconds
dt          =    0.002    ;time step in picoseconds
nsteps  = 10000        ;number of steps (total simulation time )

;constraints
constraints     = all-bonds   ;Bond types to replace by constraints
```

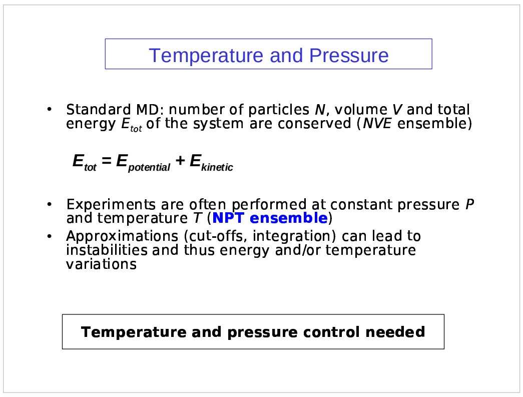


## Running a MD simulation



**Keywords for output file control in *.mdp file**

```
;Parameters to save data (what to write and when)
; frequency
nstxout        = 100 ;coordinates to output trajectory file (*.trr)
nstvout        = 100 ;velocities to output trajectory file(*.trr)
nstlog         = 10    ;energies to log file(*.log)
nstenergy      = 10    ;energies to energy file (*.edr)
nstxtcout      = 10    ;coordinates to xtc trajectory (*.xtc)
;group(s) to write
xtc_grps       = BUT    SOL    ;to xtc trajectory (*.xtc)
energygrps     = BUT   SOL    ; to energy file (*.edr)
```



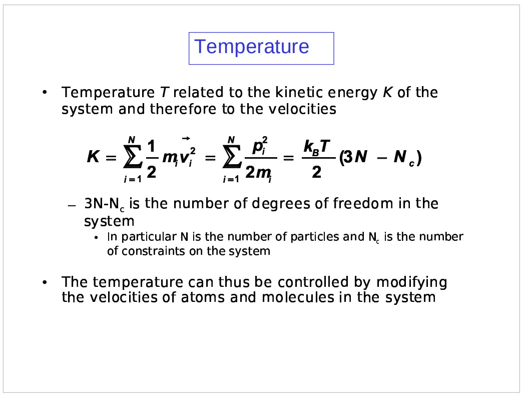
## Temperature and Pressure

- Standard MD: number of particles *N*, volume *V* and total energy $E_{tot}$ of the system are conserved (*NVE* ensemble)

$$E_{tot} = E_{potential} + E_{kinetic}$$

- Experiments are often performed at constant pressure *P* and temperature *T* (**NPT ensemble**)
- Approximations (cut-offs, integration) can lead to instabilities and thus energy and/or temperature variations

**Temperature and pressure control needed**

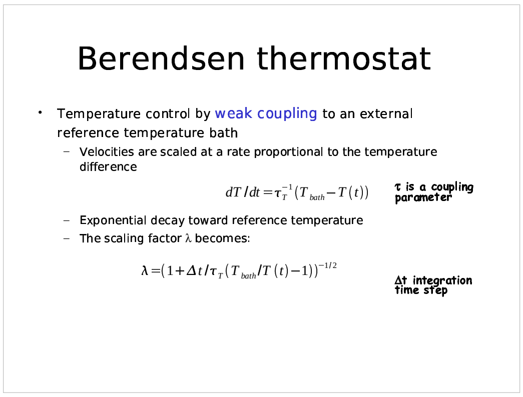


## Temperature

- Temperature *T* related to the kinetic energy *K* of the system and therefore to the velocities

$$K = \sum_{i=1}^{N} \frac{1}{2} m_i \vec{v}_i^{\,2} = \sum_{i=1}^{N} \frac{p_i^2}{2m_i} = \frac{k_B T}{2}(3N - N_c)$$

  - 3N-$N_c$ is the number of degrees of freedom in the system
    - In particular N is the number of particles and $N_c$ is the number of constraints on the system
- The temperature can thus be controlled by modifying the velocities of atoms and molecules in the system



## Berendsen thermostat

- Temperature control by weak coupling to an external reference temperature bath
  - Velocities are scaled at a rate proportional to the temperature difference

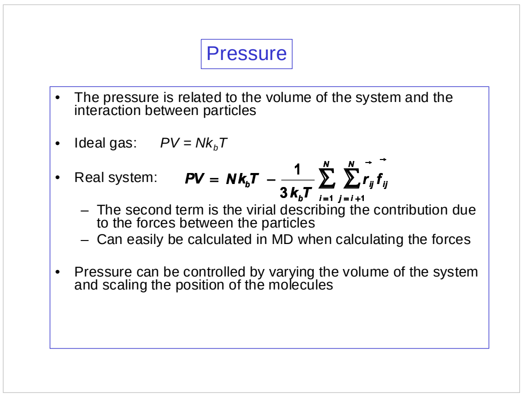
$$dT/dt = \tau_T^{-1}(T_{bath} - T(t))$$

τ is a coupling parameter

  - Exponential decay toward reference temperature
  - The scaling factor λ becomes:

$$\lambda = (1 + \Delta t / \tau_T (T_{bath}/T(t) - 1))^{-1/2}$$

Δt integration time step



## Pressure

- The pressure is related to the volume of the system and the interaction between particles
- Ideal gas: $PV = Nk_bT$
- Real system: $PV = Nk_bT - \frac{1}{3k_bT}\sum_{i=1}^{N}\sum_{j=i+1}^{N}\vec{r}_{ij}\vec{f}_{ij}$
  - The second term is the virial describing the contribution due to the forces between the particles
  - Can easily be calculated in MD when calculating the forces
- Pressure can be controlled by varying the volume of the system and scaling the position of the molecules

# Berendsen barostat

- Pressure can be controlled in a similar way as T by weak coupling to an external pressure bath with the rate of change given by:

$$dP/dt = \tau_P^{-1}(P_{ref} - P(t))$$

- Pressure can adjusted by scaling the volume by a factor $\lambda$ or the coordinate by $\lambda^{1/3}$

$$\lambda = 1 - \kappa\,\Delta t/\tau_P(P_{ref} - P(t))$$

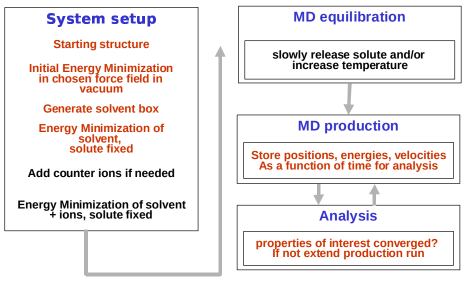
where $\kappa$ is the iso-thermal compressibility

## Keywords for temperature and pressure control in *.mdp file

```
;Temperature coupling at T=333 K (example )
tcoupl      = berendsen       ;Temperature coupling scheme
tc-grps     = BUT  SOL        ;couple which group
tau_t   = 0.1  0.1            ;relaxation time
ref_t   = 333  333        ;reference temperature (K)

;Pressure coupling
Pcoupl      = berendsen       ;pressure coupling scheme
tau_p       = 0.5              ;pressure relaxation time
compressibility = 4.5e-5       ;compressibility of system
ref_p   = 1.0                 ;reference pressure (atm)
```

## Example of a simulation setup



System setup

Starting structure

Initial Energy Minimization in chosen force field in vacuum

Generate solvent box

Energy Minimization of solvent, solute fixed

Add counter ions if needed

Energy Minimization of solvent + ions, solute fixed

MD equilibration

slowly release solute and/or increase temperature

MD production

Store positions, energies, velocities As a function of time for analysis

Analysis

properties of interest converged? If not extend production run

## MD simulations of butane in water solution at constant temperature and pressure

•Create the parameter file for around 500 ps simulation (using 2 fs time step) at the following simulation conditions:
a) Constant temperature (T=300 K);
Constant pressure (p= 1 atm);
b) Constant temperature (T=350 K);
Constant pressure (p= 1 atm).

•Run one MD simulation for each temperature.

Tips:
Use different directories for each simulation.
Use top to check that your simulations are running
Use df to check that disk space

## Files we need for the MD simulation in water of butane at constant temperature and pressure.

•**Geometry file** (*.gro or *.pdb):
the output file of the minimizations performed in water solution: *confout.gro*.

•Corrected **topology file** for all-atom force field (butane.top)
PS: include "water.itp" and check the total number of water molecule at the end of the file
•**Water topology file**, water.itp.
•**Force field files** (ff.itp and ffnb.itp)

•**Parameter file** for molecular dynamics simulation in water
Fill the file md.mdp with the correct keywords

## Check that the molecular dynamics simulation is equilibrated.

•Use the tools g_energy to extract the **temperature** and the **pressure** of the system from the energy file (ener.edr) using
*"g_energy -f ener.edr -o t.xvg" and*
*"g_energy -f ener.edr -o p.xvg"*
Plot t.xvg and p.xvg file (output of g_energy) using
*"xmgrace t.xvg" and "xmgrace p.xvg"*.

•After how many ps do the **temperature** and the **pressure** reach *a plateau*? How large is the temperature at t=0 ps ?

•Use the tools g_energy to extract **total energy** from the energy file *"g_energy -f ener.edr -o energy.xvg"*.
•Plot energy.xvg file (output of g_energy) using *"xmgrace energy.xvg"*. After how many ps does **the potential energy** reach *a plateau*?
•Is your simulation equilibrated before 500 ps? If not, run longer, if yes, after how many picoseconds?

•Visualise the trajctory using the software vmd (traj.xtc)