

Simulations in Statistical Physics

Molecular dynamics

Janos Török

Department of Theoretical Physics

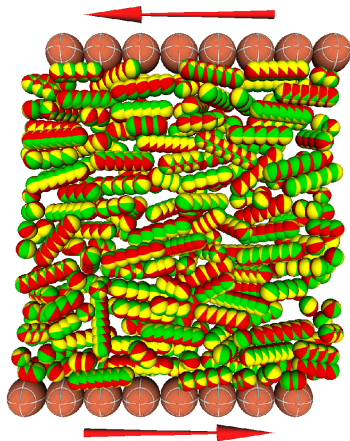
February 24, 2022

Boundary conditions

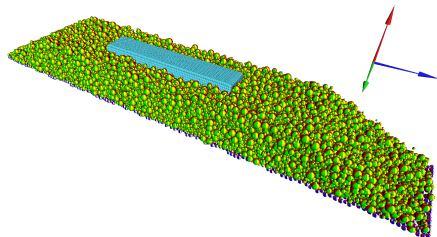
- ▶ Real boundary conditions
 - ▶ Closed (nothing)
 - ▶ Walls (with temperature)
 - ▶ Substrate (often too expensive)
- ▶ Computer based boundary conditions
 - ▶ **Periodic boundary conditions**
 - ▶ Absorbing (whatever leaves is gone)
 - ▶ Reflecting (everything is reflected back)
 - ▶ Walls (some potential)
 - ▶ Substrate (fixed basis)
 - ▶ ~~Wall with temperature~~

Boundary conditions: Examples

- ▶ Periodic boundary conditions
- ▶ Walls (some potential)

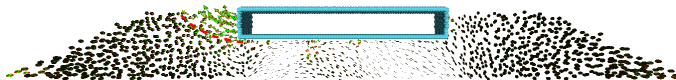
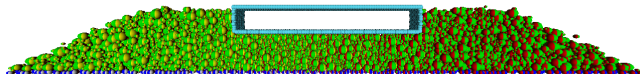


Boundary conditions: Examples

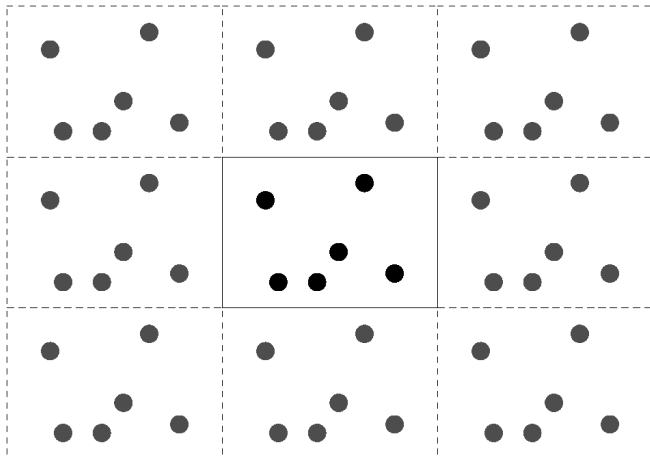


Periodic boundary conditions

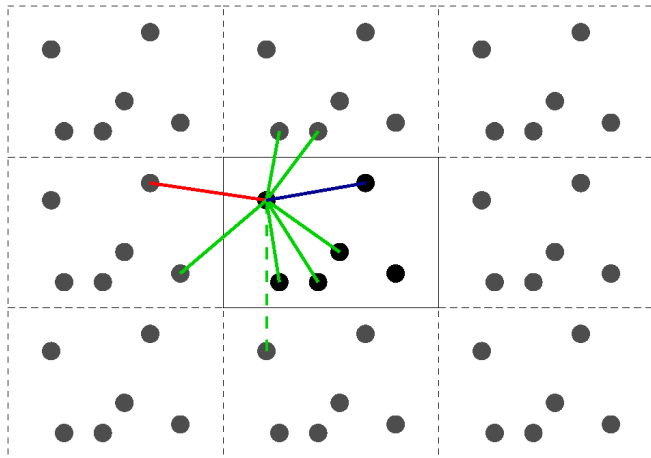
Substrate (fixed basis)



Periodic boundary conditions

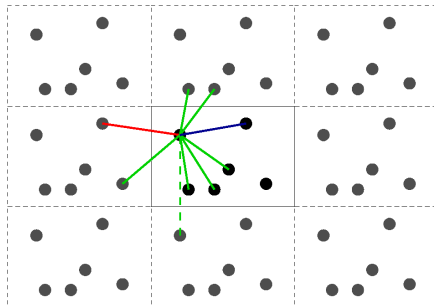


Periodic boundary conditions \rightarrow contacts



Periodic boundary conditions

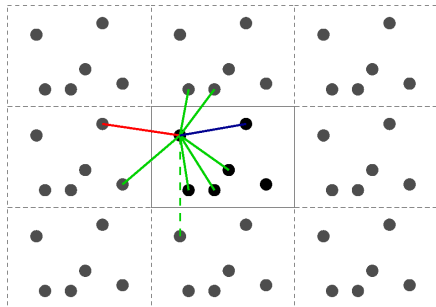
- ▶ Infinitely many neighboring cells if long range interactions
- ▶ Possibility of self interaction (must be charge neutral)
 - ▶ General solution: long range interactions are handled in k -space
- ▶ Linear momentum is conserved
- ▶ Angular momentum is **not** conserved



Periodic boundary conditions

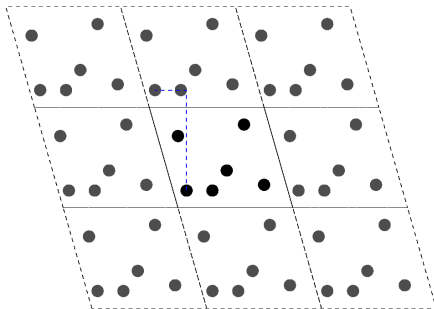
Distance

```
dx = x[i] - x[j]  
if (dx < -Lx/2) dx+=Lx;  
if (dx >  Lx/2) dx-=Lx;
```



Periodic boundary conditions deformed box

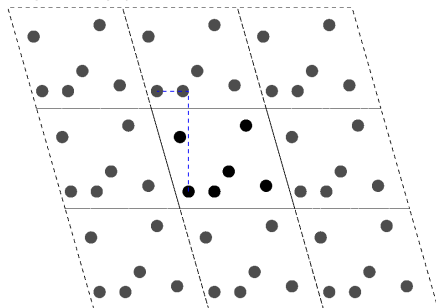
- ▶ Box is tilted, positions of particles artificially moved
- ▶ Homogeneous shear



Periodic boundary conditions deformed box

Distance

```
dx = x[i] - x[j]
dy = y[i] - y[j]
dz = z[i] - z[j]
if (dz < -Lz/2) { dz+=Lz; dx+=Dxz; dy+=Dyz; }
if (dz > Lz/2) { dz-=Lz; dx-=Dxz; dy-=Dyz; }
if (dy < -Ly/2) { dy+=Ly; dx+=Dxy; }
if (dy > Ly/2) { dy-=Ly; dx-=Dxy; }
if (dx < -Lx/2) dx+=Lx;
if (dx > Lx/2) dx-=Lx;
```



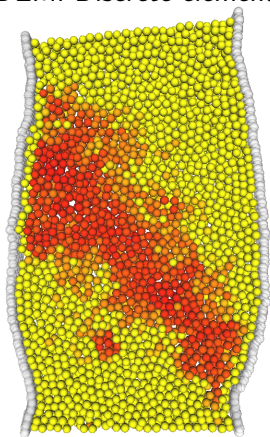
Particle based simulation

- ▶ Molecular dynamics
- ▶ Event Driven Dynamics
- ▶ Contact Dynamics
- ▶ Kinetic Monte Carlo

Molecular dynamics

MD: Molecular dynamics

DEM: Discrete element method



Application of molecular dynamics

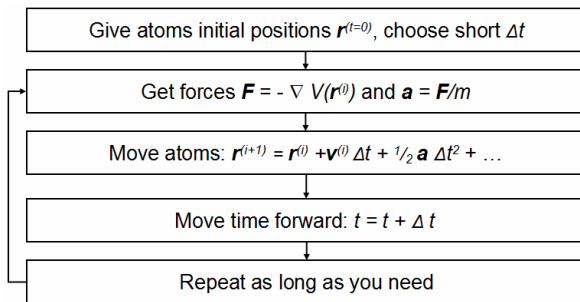
- ▶ Molecular systems (classic potentials, temperature)
 - ▶ Biophysics
 - ▶ Structural biology
 - ▶ Glasses
 - ▶ Amorphous materials
 - ▶ Liquids
- ▶ Granular materials (hard core, dissipative)
 - ▶ Stones, seeds, pills
 - ▶ Railbed
- ▶ Pedestrians
- ▶ Astrological systems (conservative, large scale)

Molecular dynamics

Simulate nature

- Solve Newton's equation of motion

$$m_i \ddot{\mathbf{r}}_i = \mathbf{f}_i = \mathbf{f}_i^{\text{ext}} + \sum_j \mathbf{f}_{ij}^{\text{int}}, \quad i, j = 1, 2 \dots N$$



Program

- ▶ Have an algorithm to calculate forces
- ▶ Get list of interacting particles
- ▶ Determine accelerations and velocities; step particles
- ▶ Set temperature

Forces

Internal forces

- ▶ Pair potential:

$$\mathbf{f}_{ij}^{\text{int}} = -\mathbf{f}_{ji}^{\text{int}} = -\nabla V(r_{ij})$$

- ▶ Many body potentials (molecular bonds)

$$\mathbf{f}_{ijk}^{\text{int}} = \mathbf{F}(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k)$$

- ▶ e.g. 3-body Stillinger-Weber potential:

$$E = \sum_i \sum_{j>i} \phi_2(r_{ij}) + \sum_i \sum_{j \neq i} \sum_{k>j} \phi_3(r_{ij}, r_{ik}, \theta_{ijk})$$

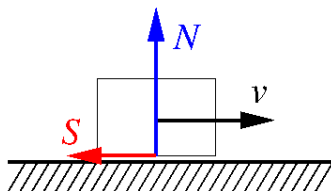
$$\phi_2(r_{ij}) = A_{ij} \epsilon_{ij} \left[B_{ij} \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{p_{ij}} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{q_{ij}} \right] \exp \left(\frac{\sigma_{ij}}{r_{ij} - a_{ij} \sigma_{ij}} \right)$$

$$\phi_3(r_{ij}, r_{ik}, \theta_{ijk}) = \lambda_{ijk} \epsilon_{ijk} [\cos \theta_{ijk} - \cos \theta_{0ijk}]^2 \exp \left(\frac{\gamma_{ij} \sigma_{ij}}{r_{ij} - a_{ij} \sigma_{ij}} \right) \exp \left(\frac{\gamma_{ik} \sigma_{ik}}{r_{ik} - a_{ik} \sigma_{ik}} \right)$$

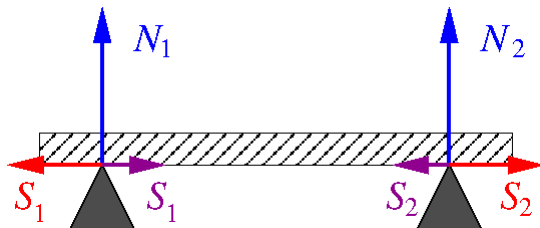
- ▶ Friction forces (next slide...)

Friction forces

- Moving:



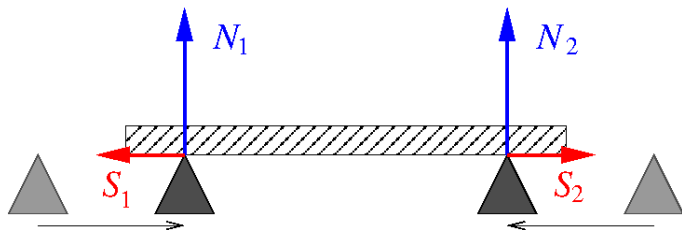
- Stationary:



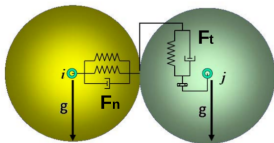
Friction forces

- ▶ Position is not enough to set friction forces
- ▶ No movement \rightarrow no friction forces
- ▶ Solution:

We need history:



Contact history: spring dashpot model



- ▶ Position is not enough to set friction forces
- ▶ Normal force:

$$F_n = k_n \delta n_{ij} - m_{\text{eff}} \gamma_n \Delta v_n$$

- ▶ Tangential force:

$$F_t = k_t \Delta s_t + m_{\text{eff}} \gamma_t \Delta v_t$$

$$\Delta s_t = n_t \int_{t_c}^t \{ \Delta v_t(t') + [\omega_i(t') r_i - \omega_j(t') r_j] \} dt'$$

- ▶ Limit Δs_t to satisfy $|F_t| \leq \mu F_n$
- ▶ k stiffness, γ damping (critical)

Program

- ▶ Have an algorithm to calculate forces
- ▶ **Get list of interacting particles**
- ▶ Determine accelerations and velocities; step particles
- ▶ Set temperature

Find pairs

Now we know how to calculate forces. How to get pairs?

- ▶ All pairs: $\sim N^2$ calculations. *Only* if there is no other way!
- ▶ Short range interactions: box method
- ▶ Long range interactions: k-space

k-space solution

- ▶ Long range interactions (e.g. Coulomb) cannot be cut off
- ▶ Often more periodic images are needed
- ▶ k-space (Fourier-transformation in 3d!)
 - ▶ Solution of linear problems by Green's-function
 - ▶ Coulomb problem: in Fourier space \rightarrow multiplication with $1/k^2$!
 - ▶ Generally it is done by Ewalds summation

Ewald summation

- ▶ The total electrostatic potential energy

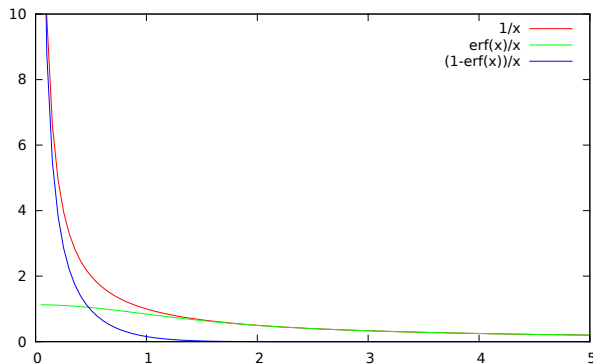
$$W = \frac{1}{8\pi\epsilon_0} \sum_{i \neq j} \frac{q_i q_j}{|r_i - r_j|}$$

- ▶ Factor 2 is for double counting all contacts
- ▶ For infinite system the expression does not converge
- ▶ Ewald's idea:

$$\frac{1}{r} = \frac{\text{erf}(\sqrt{\eta}r/2)}{r} + \frac{1 - \text{erf}(\sqrt{\eta}r/2)}{r}$$

- ▶ The first term goes to a constant for small r but has long range interactions
- ▶ The second term has a singular behavior at $r \rightarrow 0$ but vanishes exponentially
- ▶ $\text{erfc}(x) = 1 - \text{erf}(x)$

Ewald summation



► Thus the calculation of the electrostatic energy is

$$W = \frac{1}{8\pi\epsilon_0} \left(\sum_{i \neq j} \frac{q_i q_j \text{erf}(\sqrt{\eta}|r_i - r_j|/2)}{|r_i - r_j|} + \right. \\ \left. + \sum_{i \neq j} \frac{q_i q_j \text{erfc}(\sqrt{\eta}|r_i - r_j|/2)}{|r_i - r_j|} \right)$$

Ewald summation

- ▶ Thus the calculation of the electrostatic energy is

$$W = \frac{1}{8\pi\epsilon_0} \left(\sum_{i \neq j} \frac{q_i q_j \operatorname{erf}(\sqrt{\eta}|r_i - r_j|/2)}{|r_i - r_j|} + \sum_{i \neq j} \frac{q_i q_j \operatorname{erfc}(\sqrt{\eta}|r_i - r_j|/2)}{|r_i - r_j|} \right)$$

- ▶ For an appropriate choice of η , the second term converges fast
- ▶ The first term is evaluated in the Fourier space

$$W_1 = \frac{4\pi}{L^3} \sum_{i \neq j} q_i q_j \left(\sum_{\mathbf{k}} e^{-i\mathbf{k} \cdot \mathbf{r}_{ij}} \frac{1}{k^2} e^{-k^2/4\eta^2} \right)$$

Program

- ▶ Have an algorithm to calculate forces
- ▶ Get list of interacting particles
- ▶ Determine accelerations and velocities; step particles
- ▶ Set temperature

Euler method

- ▶ Second order differential equation:

$$\ddot{y} = f(\dot{y}(t), y(t), t)$$

- ▶ First velocity ($v = \dot{y}$)

$$v_{n+1} = v_n + \Delta t f_n + \mathcal{O}(\Delta t^2)$$

- ▶ Then position

$$y_{n+1} = y_n + \Delta t v_n + \mathcal{O}(\Delta t^3)$$

- ▶ Do not use it!

Implicit Euler method (backward)

- ▶ Second order differential equation:

$$\ddot{y} = f(\dot{y}(t), y(t), t)$$

- ▶ First velocity ($v = \dot{y}$)

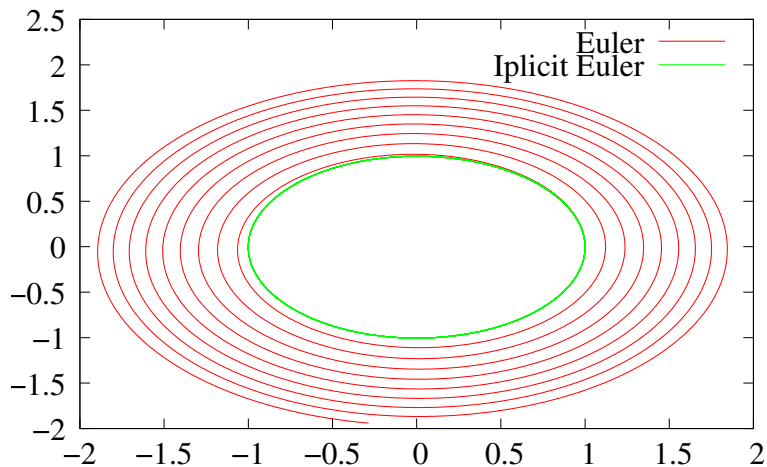
$$v_{n+1} = v_n + \Delta t f_n + \mathcal{O}(\Delta t^2)$$

- ▶ Then position

$$y_{n+1} = y_n + \Delta t v_{n+1} + \mathcal{O}(\Delta t^3)$$

- ▶ Surprisingly good!

Euler



Verlet method

- ▶ Second order differential equation:

$$\ddot{y} = f(y(t), t)$$

- ▶ From central difference

$$y_{n+1} = 2y_n - y_{n-1} + \Delta t^2 f_n + \mathcal{O}(\Delta t^4)$$

- ▶ Leapfrog

$$y_{n+1} = y_n + \Delta t v_{n+\frac{1}{2}}$$

$$v_{n+\frac{1}{2}} = v_{n+\frac{1}{2}} + \Delta t f_n$$

- ▶ None of them is used
- ▶ Velocity dependent forces are difficult to add

Velocity Verlet method

- ▶ The one actually used in all codes:

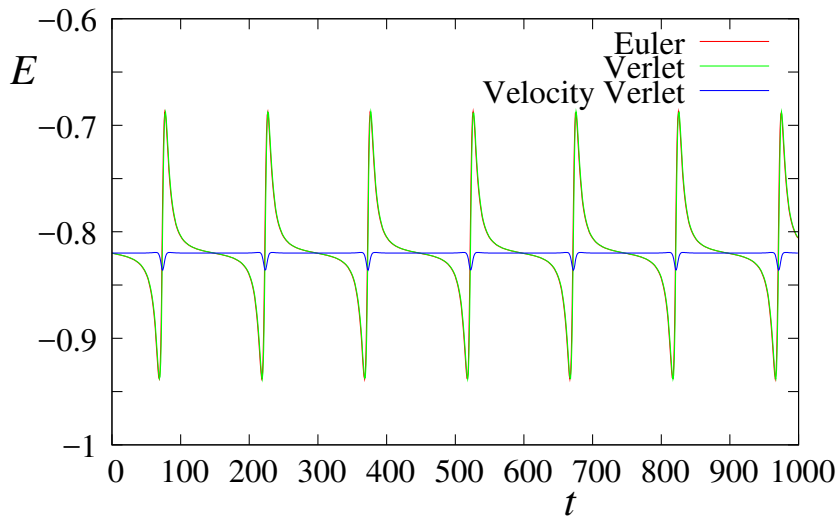
$$y_{n+1} = y_n + \Delta t v_n + \frac{1}{2} \Delta t^2 f_n$$

$$v_{n+1} = v_n + \frac{1}{2} \Delta t (f_n + f_{n+1})$$

- ▶ Implementation

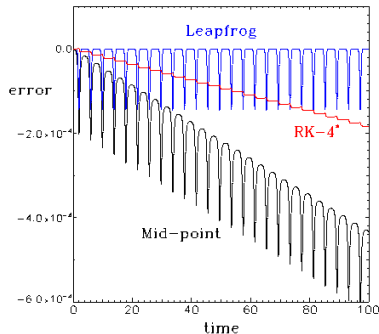
1. $v_{n+1/2} = v_n + \frac{1}{2} f_n \Delta t$
2. $y_{n+1} = y_n + \Delta t v_{n+1/2}$
3. Calculate forces
4. $v_{n+1} = v_{n+1/2} + \frac{1}{2} f_{n+1} \Delta t$

Energy comparison



Error

Method	Error	Cumulative error
Euler:	Δt^3	Δt
Runge-Kutta:	Δt^5	Δt^4
Verlet:	Δt^4	Δt^2
Leapfrog:	Δt^4	Δt^2



Symplectic integrator

- ▶ Energy (slightly modified) is conserved
- ▶ Time reversibility
 - ▶ Verlet
 - ▶ Leapfrog
- ▶ Most molecular dynamics methods use Verlet!
 - ▶ Forces are calculated once per turn
 - ▶ Microcanonical (NVE) modelling can be only done with these

Multiple time scale integration

- ▶ Different force range
 - ▶ Short range change fast
 - ▶ Long range change slowly
- ▶ Recalculate long range forces only in every n th times-step
 - ▶ Forces are calculated once per turn
- ▶ Typical examples:
 - ▶ Intramolecular forces: strong, high frequency
 - ▶ Intermolecular forces (e.g. Lennard-Jones, Coulomb) slow
- ▶ Similar technic: coupling to fields

Molecular dynamics

Program:

- ▶ Have an algorithm to calculate forces
- ▶ Get list of interacting particles
- ▶ Determine accelerations and velocities; step particles
- ▶ Set temperature

Temperature

Definition:

- ▶ Encyclopedia Britannica, Wikipedia:
"A temperature is a numerical measure of hot or cold."
- ▶ Manifestation of thermal energy
- ▶ Thermodynamics:
Second law of thermodynamics & Carnot engine

$$\delta Q = TdS$$

$$\eta_{\max} = \eta_{\text{Carnot}} = 1 - T_C/T_H$$

- ▶ Statistical physics:

$$\beta \equiv \frac{1}{k_B} \left(\frac{\partial S}{\partial E} \right)_{V,N} = \frac{1}{k_B T}$$

Definition of temperature

Temperature is a measure of the random submicroscopic motions and vibrations of the particle constituents of matter.

The average **kinetic** energy per particle degrees of freedom is

$$\bar{E} = \frac{1}{2} k_B T$$

Molecular dynamics conserves only the *total* energy!

Task: Control kinetic energy!

Setting temperature

- ▶ Experiment
 - ▶ Environment
 - ▶ Mixing → uniform temperature
- ▶ Simulation
 - ▶ Control the kinetic energy (velocities)
 - ▶ Mixing → Maxwell-Boltzmann distribution

Nosé-Hoover thermostat

- ▶ Original Hamiltonian

$$H_0 = \sum_i \frac{p_i^2}{2m_i} + U(q)$$

- ▶ Heatbath in the Hamiltonian:

$$H_n = \sum_i \frac{p_i'^2}{2m_i} + U(q') + \frac{p_s^2}{2Q} + gk_B T \log(s)$$

- ▶ Extra degree of freedom s .
- ▶ Q "mass" related to $s \rightarrow$ controls the speed of convergence
- ▶ $g = 3N$ the number degrees of freedom
- ▶ p' and q' are virtual coordinates

Nosé-Hoover thermostat

- ▶ Virtual coordinates, vs. original ones:

$$p = p'/s$$

$$q = q'$$

$$t = \int \frac{1}{s} dt'$$

- ▶ Solution of the new Hamiltonian:

$$\xi = \dot{s}/s = p_s/Q$$

$$\dot{q}' = \frac{p'}{m}$$

$$\dot{p}'_i = -\frac{\partial U}{\partial q'_i} - \xi p'_i$$

$$\dot{\xi} = \frac{1}{Q} \left(\sum_i \frac{\dot{p}'_i{}^2}{m_i} - g k_B T \right)$$

Molecular dynamics

- ▶ Create sample
 - ▶ Crystal
 - ▶ Random deposition
 - ▶ Distorted crystal
 - ▶ Simulation
- ▶ Temperate sample
- ▶ Make test
- ▶ Collect data
 - ▶ Data size: e.g. $N = 10^4$, $t = 10^6$ small simulation:
 - ▶ 1 hour on 1 core PC
 - ▶ 3 doubles/atom \rightarrow 24 bytes/atom/timesteps
 - ▶ Result $2.4 \cdot 10^{11}$ bytes = 240 Gigabytes