



Numerical Methods II - Statistical Mechanics and Molecular Dynamics

Professori:

G. BUSSI
C. MICHELETTI

Autore:

Iacopo POLI

Indice

Capitolo 1

Statistical Mechanics

For reference you can look to ch. 6-7 of *Statistical Mechanics*, Kerson, Huang, MIT.

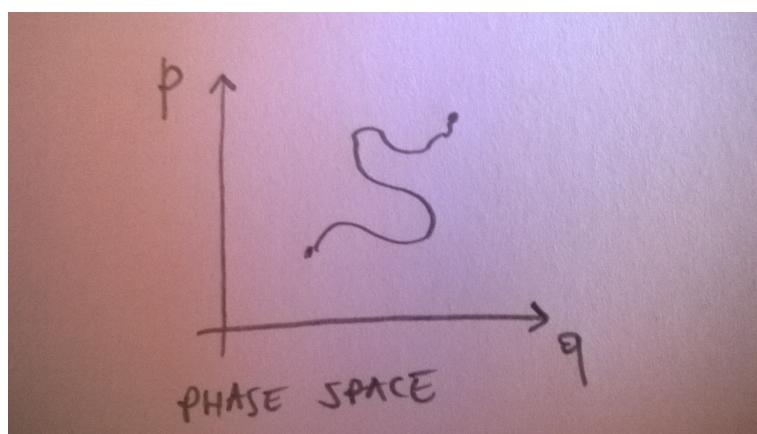
1.1 Postulate of Classical Statistical Mechanics

Statistical mechanics concerns equilibrium properties of systems, it doesn't study how the system approaches the equilibrium nor if it can reach it, only what the situation at equilibrium is for a given system.

Statistical mechanics studies systems at thermodynamic equilibrium. In particular we assume to be in *thermodynamic limit*: N, V are large enough that it makes sense to introduce $n = \frac{N}{V}$ and $N \rightarrow \infty, V \rightarrow \infty$ but n is fixed and constant (volume and number of particles grow in proportion).

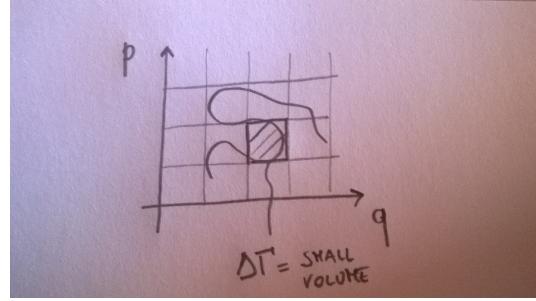
In the study of gas of N particles, we introduce $3N q_i$ coordinates and $3N p_i$ coordinates. Weak interactions are assumed, in order to let the system reach equilibrium, but apart from this, the motion is governed by Hamilton equations of motion.

$$\begin{aligned}\dot{q}_i &= \frac{\partial H}{\partial p_i} \\ \dot{p}_i &= -\frac{\partial H}{\partial q_i}\end{aligned}$$



In the phase space, trajectories that intersect themselves are not allowed, since at the intersection I wouldn't know where to proceed and this would imply a non-deterministic behavior. Given this rule, I can have two kinds of trajectories:

- closed curves: they conserve the energy, but don't equilibrate, for example, harmonic oscillators.
- curves that fill out the space compatible with constraints on the system without self-intersecting or closing.



If we divide the phase space in small volumes $\Delta\Gamma$, we can build an histogram adding +1 every time the system visits a particular spot of the grid. At the end we will have an histogram of how many times that state has been visited by the system. The dimension of the volume gives the precision of momenta and coordinates: it is the level of fine graining.

We want to know how much time the system spends in every bin:

$$\lim_{T \rightarrow \infty} \frac{\Delta t_i(\Gamma_i)}{T} = \rho(\vec{q}_i, \vec{p}_i) \cdot \Delta\Gamma$$

if the system is in equilibrium, where $\rho(\vec{q}_i, \vec{p}_i)$ is a probability density.
Let's take $f(\vec{q}, \vec{p})$ as a generic observable:

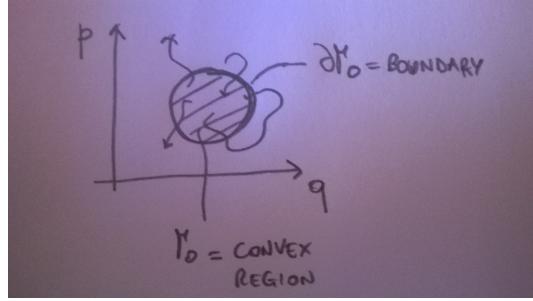
$$\langle f \rangle = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T dt f(\vec{q}(t), \vec{p}(t)) = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_i \Delta t(q_i, p_i) \cdot f(q_i, p_i)$$

since $\lim_{T \rightarrow \infty} \frac{1}{T} \Delta t(q_i, p_i) = \rho_i$:

$$\langle f \rangle = \sum_i \lim_{T \rightarrow \infty} \frac{\Delta t_i}{T} \cdot f(\vec{q}_i, \vec{p}_i) = \sum_i \rho(\vec{q}_i, \vec{p}_i) \cdot \Delta\Gamma \cdot f(\vec{q}_i, \vec{p}_i) = \int d\Gamma f(\vec{q}, \vec{p}) \cdot \rho(\vec{q}, \vec{p})$$

The expected value of an observable can be computed as time average of a very long trajectory, or value of observable in any point of the phase space per a weight of time spent in that point.

Ensemble: a collection of mental copies of the system picked with an appropriate probability density function $\rho(\vec{q}, \vec{p})$



Liouville's Theorem Let's choose a convex region Γ_0 of the phase space, whose boundary is $\partial\Gamma_0$. We can write:

$$-\frac{\partial}{\partial t} \int_{\Gamma_0} d\Gamma \rho = \int_{\partial\Gamma_0} dS \vec{n} \cdot (\rho \cdot \vec{v})$$

and using Gauss' Theorem:

$$\int_{\partial\Gamma_0} dS \vec{n} \cdot (\rho \cdot \vec{v}) = \int_{\Gamma_0} d\Gamma \vec{\nabla} \cdot (\rho \cdot \vec{v})$$

since the choice of Γ_0 is arbitrary, this implies:

$$-\frac{\partial \rho}{\partial t} = \vec{\nabla} \cdot (\rho \cdot \vec{v})$$

exploit divergence:

$$-\frac{\partial \rho}{\partial t} = \sum_{i=1}^{3N} \left[\frac{\partial \rho}{\partial q_i} \dot{q}_i + \frac{\partial \rho}{\partial p_i} \dot{p}_i \right] + \rho \sum_{i=1}^{3N} \left[\frac{\partial}{\partial q_i} \dot{q}_i + \frac{\partial}{\partial p_i} \dot{p}_i \right]$$

$$\text{since } \left[\frac{\partial}{\partial q_i} \dot{q}_i + \frac{\partial}{\partial p_i} \dot{p}_i \right] = \left[\frac{\partial^2 H}{\partial q_i \partial p_i} - \frac{\partial^2 H}{\partial p_i \partial q_i} \right] = 0$$

$$\sum_{i=1}^{3N} \left[\frac{\partial \rho}{\partial q_i} \frac{\partial q_i}{\partial t} + \frac{\partial \rho}{\partial p_i} \frac{\partial p_i}{\partial t} \right] + \frac{\partial \rho}{\partial t} = 0$$

and the left hand side is the definition of total derivative of ρ in time:

$$\frac{d\rho}{dt} = 0$$

which is the Liouville's theorem: *total derivative of ρ with respect to time is zero, which means probability density ρ is constant with time.*

1.2 Microcanonical ensemble

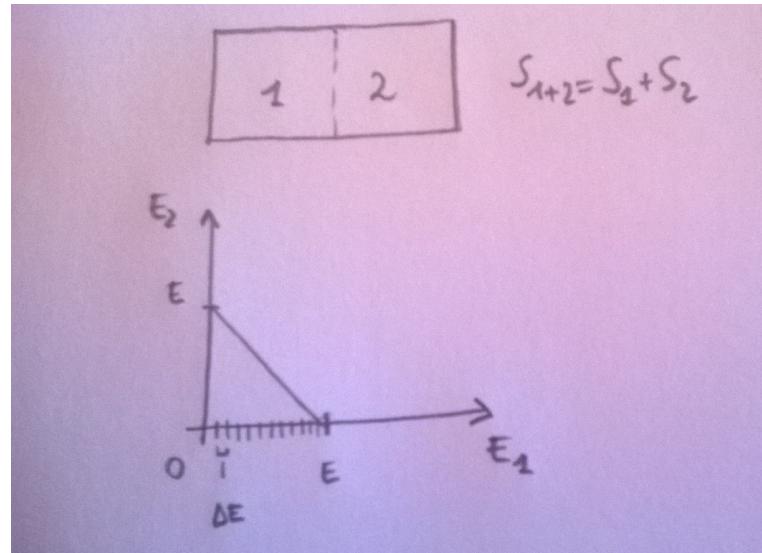
Let's consider a system with weak interactions, in which $\varepsilon < H < \varepsilon + \Delta\varepsilon$, so that a priori probability postulate is:

$$\rho(\vec{q}, \vec{p}) = \begin{cases} \text{const, if } H(\vec{q}, \vec{p}) \in [\varepsilon, \varepsilon + \Delta\varepsilon] \\ 0, \text{ otherwise} \end{cases}$$

We define entropy as:

$$S(E) = K_B \ln(\Gamma_\Delta(E))$$

where $\Gamma_\Delta(E) = \iint_{\varepsilon < H < \varepsilon + \Delta\varepsilon} d^{3N}q d^{3N}p \cdot 1.$
 Since entropy is an extensive property, $S_{1+2} = S_1 + S_2$ must hold.



if I know E_1 , then $E_2 = E - E_1$. Then:

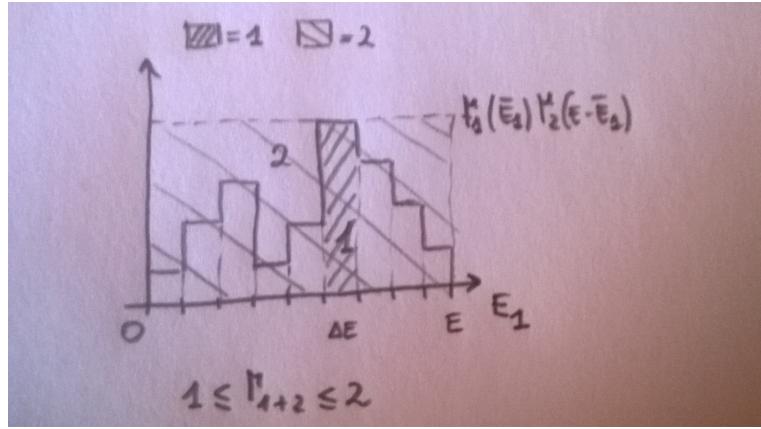
$$S_1 = K_B \ln(\Gamma_1(E_1))$$

$$S_2 = K_B \ln(\Gamma_2(E_2))$$

$$\Gamma_{1+2} = \sum_{E_1=0}^E \Gamma_1(E_1) \cdot \Gamma_2(E - E_1)$$

by construction:

$$\Gamma_1(\bar{E}_1) \cdot \Gamma_2(E - \bar{E}_1) \leq \Gamma_{1+2} \leq \Gamma_1(\bar{E}_1) \cdot \Gamma_2(E - \bar{E}_1) \cdot \frac{E}{\Delta E}$$



Since logarithm is a monotonically increasing function, I can apply it:

$$\ln(\Gamma_1(\bar{E}_1) \cdot \Gamma_2(E - \bar{E}_1)) \leq \ln(\Gamma_{1+2}) \leq \ln(\Gamma_1(\bar{E}_1)) + \ln(\Gamma_2(E - \bar{E}_1)) + \ln(E) - \ln(\Delta E)$$

using logarithm properties:

$$\ln(\Gamma_1(\bar{E}_1)) + \ln(\Gamma_2(E - \bar{E}_1)) \leq \ln(\Gamma_{1+2}) \leq \ln(\Gamma_1(\bar{E}_1)) + \ln(\Gamma_2(E - \bar{E}_1)) + K_B \ln(E) - K_B \ln(\Delta E)$$

multiplying by K_B :

$$S_1(\bar{E}_1) + S_2(E - \bar{E}_1) \leq S_{1+2}(E) \leq S_1(\bar{E}_1) + S_2(E - \bar{E}_1) + K_B \ln(E) - K_B \ln(\Delta E)$$

Using *squeeze theorem*, we take the limit for $N \rightarrow \infty$. Volume in phase space increases exponentially with N, this means that entropy grows linearly with N. The last two terms are negligible as $N \rightarrow \infty$ with respect to entropies, because $K_B \ln(E) \rightarrow \ln(N)$ and $K_B \ln(\Delta E)$ is a constant independent of N. So:

$$S_1(\bar{E}_1) + S_2(E - \bar{E}_1) \leq S_{1+2}(E) \leq S_1(\bar{E}_1) + S_2(E - \bar{E}_1)$$

that implies:

$$S_{1+2}(E) = S_1(\bar{E}_1) + S_2(E - \bar{E}_1)$$

which proves the extensive property of the entropy. Actually we have proven also that system one can take any value of energy, but choose to spend the most of the time in \bar{E}_1 , that is the value of energy that maximizes the product $\Gamma_1(E_1)\Gamma_2(E_2)$.

This means that \bar{E}_1 is the value that makes the product stationary:

$$\left[\frac{\partial}{\partial E_1} \Gamma_1(E_1)\Gamma_2(E - E_1) \right]_{\bar{E}_1} = 0$$

making the derivative and using $\partial E_2 = -\partial E_1$:

$$\left[\Gamma_2(E_2) \frac{\partial}{\partial E_1} \Gamma_1(E_1) \right]_{\bar{E}_1} - \left[\Gamma_1(E_1) \frac{\partial}{\partial E_2} \Gamma_2(E_2) \right]_{\bar{E}_1} = 0$$

so we find an equation that describes the condition for equilibrium, in which left hand side depends only on properties of system 1 and right hand side from properties of system 2:

$$\left[\frac{1}{\Gamma_1(E_1)} \frac{\partial}{\partial E_1} \Gamma_1(E_1) \right]_{\bar{E}_1} = \left[\frac{1}{\Gamma_2(E_2)} \frac{\partial}{\partial E_2} \Gamma_2(E_2) \right]_{\bar{E}_1}$$

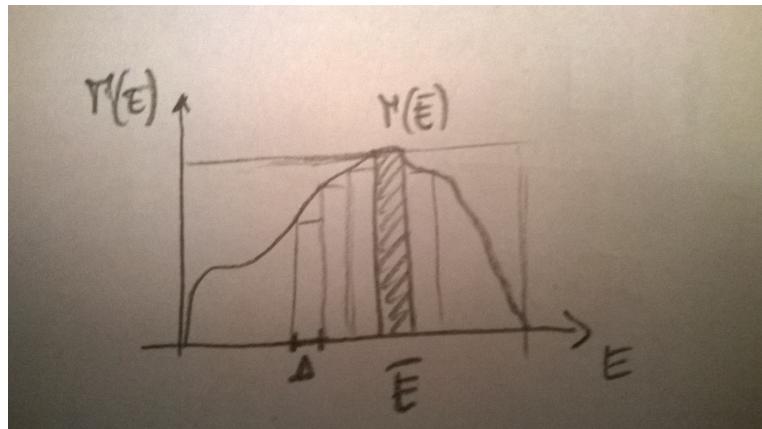
A second definition of entropy We now want to demonstrate that summing states on $0 \leq H \leq E$ or $E \leq H \leq E + \Delta$ is the same.

Claim:

$$S_\Delta(E) = k_B \ln \left(\int_{E \leq H \leq E + \Delta} d^{6N} \Gamma \right) = k_B \ln \left(\int_{0 \leq H \leq E} d^{6N} \Gamma \right)$$

Proof: with analogous procedure as before

$$\Gamma_\Delta(\bar{E}) \leq \sum_{i=0}^{E/\Delta} \Gamma_\Delta(E_i) \leq \Gamma_\Delta(\bar{E}) \cdot \frac{E}{\Delta}$$



taking the logarithm of each member and multiplying by k_B (monotonically increasing function and positive number, so we are allowed):

$$S_{\text{1st def}}(\bar{E}) \leq S_{\text{2nd def}}(E) \leq S_{\text{1st def}}(\bar{E}) + k_B \ln(E) - k_B \ln(\Delta)$$

since $\ln(E) \propto \ln(N)$ and $\ln(\Delta) \propto \text{constant}$ as $N \rightarrow \infty$, while $S \propto N$, using squeeze theorem for the limit $N \rightarrow \infty$:

$$S_{\text{2nd def}}(E) = S_{\text{1st def}}(\bar{E})$$

For all reasonable systems the maximum of S is obtained at the largest value of energy, in this case the two definitions coincide. \bar{E} is the value that makes $\Gamma(E)$ the largest and it's the maximum value of E. In other words $\Gamma(E)$ is an increasing function of E.

N.B.: a posteriori we can say that the volume of integration in the first definition of entropy is $\bar{E} \leq H \leq \bar{E} + \Delta$, but a priori we don't know that, so we use a more general notation.

We now derive the explicit expression of $S(E)$ from the second definition:

$$S(E) = k_B \ln \left(\int_{0 \leq H \leq E} d^6 N \Gamma \right) = k_B \ln \left[\int d^{3N} q \int_{0 \leq \sum_i |p_i|^2 \leq 2mE} d^{3N} p \right] = k_B \ln \left[V^N (2mE)^{3N/2} \cdot c_{3N} \right]$$

where $c_{3N} = \frac{3N}{2} \ln \pi - \frac{3N}{2} \ln \frac{3N}{2} + \frac{3N}{2} + \text{higher order terms}$.
Taking the partial derivative in energy:

$$\frac{\partial S(E)}{\partial E} = \frac{3}{2} N k_B E^{-1}$$

Experimentally

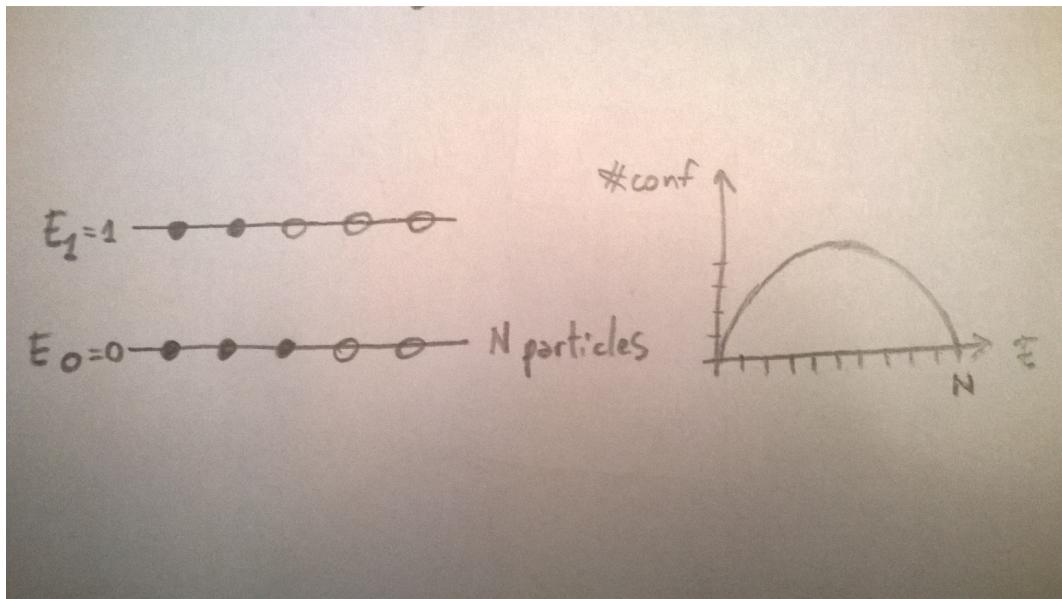
$$E = \frac{3}{2} N k_B T$$

so:

$$\frac{\partial S(E)}{\partial E} = \frac{1}{T}$$

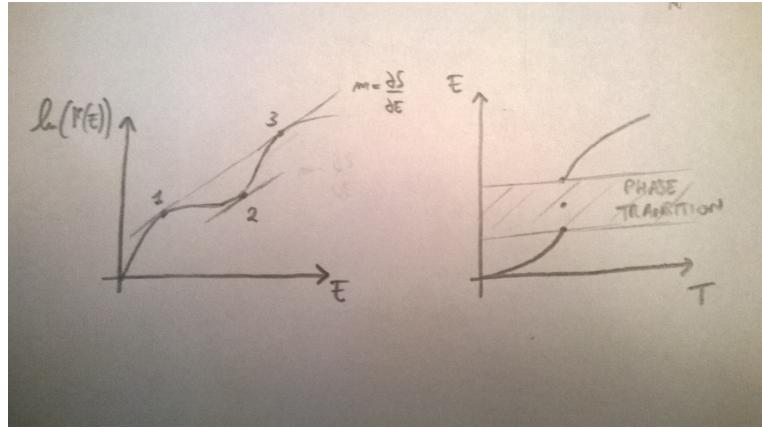
If $\Gamma(E)$ is an increasing function of E , its logarithm will be and $S(E)$ will be an increasing function too.

Its partial derivative in E will be positive, so $T > 0$.



Laser - Different behavior of S and T $\Gamma(E)$ is a non monotonic function of E . In this case, until \tilde{E} is reached, T is rising, at \tilde{E} $T \rightarrow \infty$, then $T < 0$ (figure above).

Phase transitions In some situations if I plot E vs T , I get that one slope touches three points, it's a **phase transition**.



Computing observables Any observable can be computed as time average or:

$$\langle A \rangle = \frac{\int d^{6N} \Gamma \rho(\vec{q}, \vec{p}) \cdot A(\vec{q}, \vec{p})}{\int d^{6N} \Gamma \rho(\vec{q}, \vec{p})}$$

$S = k_B \ln(\Gamma_\Delta(E))$ is not a definition like that, we can define it in a more compliant way (*Shannon Entropy*):

$$S \propto - \int d^{6N} \Gamma \rho(\vec{q}, \vec{p}) \cdot \ln(\rho(\vec{q}, \vec{p}))$$

1.2.1 Computing ρ

Given this definition of entropy, we have to maximize the functional $F[\rho]$, in order to find ρ^* that makes it stationary. To simplify notation: $d^{6N} \Gamma = d\Gamma$.

$$F[\rho] = -k_B \int d\Gamma [\rho \ln(\rho)]$$

with conditions:

$$\begin{cases} \int d\Gamma \rho(q, p) = 1 \\ \rho(q, p) = 0 \text{ if } H(q, p) \in [E, E + \Delta E] \end{cases}$$

We use **Lagrange multipliers**:

$$F[\rho] = -k_B \int d\Gamma [\rho \ln(\rho) - \lambda \rho]$$

If we find solution ρ^* and introduce a pointwise perturbation $\Delta\rho$, then we won't have linear dependence of ΔF from $\Delta\rho$, but $\Delta F \propto (\Delta\rho)^2$.

$$\delta F = F[\rho + \Delta\rho] - F[\rho]$$

if I'm dealing with ρ^* that maximizes F , in the limit $\frac{\rho}{\Delta\rho}$ small:

$$F[\rho^* + \Delta\rho] - F[\rho^*] = \Delta\rho^2 \text{ or higher order terms}$$

$$\delta F = -k_B \int d\Gamma \{(\rho + \delta\rho) \ln(\rho + \delta\rho) - \lambda(\rho + \delta\rho) - [\rho \ln \rho - \lambda \rho]\}$$

given $\ln(\rho + \delta\rho) \sim \ln \rho + \frac{\delta\rho}{\rho} + \text{higher order terms}$, that is a "generalization of Taylor expansion to functionals":

$$\delta F = -k_B \int d\Gamma \{\rho \ln \rho + \delta\rho + \delta\rho \ln \rho - \lambda \delta\rho - \rho \ln \rho\} = -k_B \int d\Gamma \{\delta\rho[(1 - \lambda) + \ln \rho]\}$$

if we are dealing with ρ^* , then the first order term in $\delta\rho$ needs to disappear, so, since we can choose $\delta\rho$ in an arbitrary way, we have to annihilate the integrand:

$$\ln(\rho^*(q, p)) + 1 - \lambda = 0 \text{ for all } q, p \text{ compatible with } E \leq H \leq E + \Delta E$$

then:

$$\rho^*(q, p) = e^{\lambda-1} = \text{constant}$$

1.3 Canonical Ensemble

1.3.1 Computing ρ

Now our constraint is that the **average energy of the system is constant**. Our conditions are:

$$\begin{cases} \int d\Gamma \rho(q, p) = 1 \\ \int d\Gamma \rho(q, p) H(q, p) = \langle H \rangle = \bar{E} \end{cases}$$

Again we take the functional and maximize it with Lagrange multipliers (this time λ and β). The calculation is very similar as before and we use always $\rho \ln \rho \sim \ln \rho + \frac{\delta\rho}{\rho}$:

$$\delta F = -k_B \int d\Gamma \{\delta\rho[(1 - \lambda) + \ln \rho + \beta H]\}$$

for the same reason as before:

$$\ln(\rho^*) + 1 - \lambda + \beta H = 0$$

then

$$\rho^* = e^{\lambda-1-\beta H} \propto e^{-\beta H}$$

that is the result for the **canonical ensemble**. Actually we find:

$$\rho^* = \frac{e^{-\beta H}}{\int d\Gamma e^{-\beta H}}$$

Physical meaning of β

$$\langle H \rangle = \frac{\int d\Gamma H e^{-\beta H}}{\int d\Gamma e^{-\beta H}} = -\frac{\partial}{\partial \beta} \left[\ln \int d\Gamma e^{-\beta H} \right] = -\frac{\partial}{\partial \beta} \ln \mathcal{Z}$$

In an ideal gas $H = \sum_i \frac{p_i^2}{2m}$:

$$\mathcal{Z} = \int d^{3N} q \int d^{3N} p e^{-\beta \sum_i \frac{p_i^2}{2m}} = V^N \left[\int dp e^{-\beta \frac{p^2}{2m}} \right]^{3N}$$

that is a gaussian integral, given:

$$\int dx e^{-\frac{x^2}{2\sigma^2}} = \sqrt{2\pi\sigma^2}$$

we have:

$$\mathcal{Z} = V^N \left(2\pi \frac{m}{\beta} \right)^{\frac{3N}{2}}$$

taking the logarithm of \mathcal{Z} , doing its derivative in β and then multiplying by a minus sign:

$$\langle H \rangle = \frac{3}{2} \frac{1}{\beta}$$

experimentally:

$$\langle H \rangle = \frac{3}{2} \frac{1}{\beta} = \frac{3}{2} N k_B T$$

so $\beta = \frac{1}{k_B T}$

1.3.2 Fluctuation and dissipation relationship

If we take the second derivative of $\ln \mathcal{Z}$ in β , we have:

$$\begin{aligned} \frac{\partial^2}{\partial \beta^2} \ln \mathcal{Z} &= -\frac{\partial}{\partial \beta} \left(-\frac{\partial}{\partial \beta} \ln \mathcal{Z} \right) = -\frac{\partial}{\partial \beta} \frac{\int d\Gamma H e^{-\beta H} \mathcal{Z} - \int d\Gamma H e^{-\beta H} \int d\Gamma H e^{-\beta H}}{\int d\Gamma e^{-\beta H}} = \\ &= \frac{\int d\Gamma H^2 e^{-\beta H}}{\mathcal{Z}} - \left(\frac{\int d\Gamma H e^{-\beta H}}{\mathcal{Z}} \right)^2 = \langle H^2 \rangle - \langle H \rangle^2 \end{aligned}$$

but, since $d\beta = -\frac{1}{k_B T^2} dT$

$$\frac{\partial^2}{\partial \beta^2} \ln \mathcal{Z} = -\frac{\partial}{\partial \beta} \langle H \rangle = k_B T^2 \frac{\partial \langle H \rangle}{\partial T}$$

and $c_v = \frac{\partial \langle H \rangle}{\partial T}$ is the specific heat. So we've found:

$$\langle H^2 \rangle - \langle H \rangle^2 = k_B T^2 \frac{\partial \langle H \rangle}{\partial T}$$

a relationship between an equilibrium property on the left and a something that depends on how energy of the system varies changing T. This is called **fluctuation and dissipation relationship**

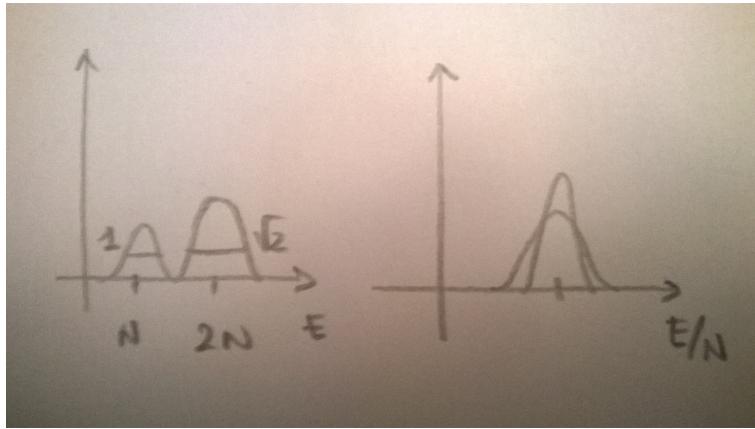
We have

$$\frac{\Delta E}{E} = \sqrt{\frac{< H^2 > - < H >^2}{< H >^2}} \propto \sqrt{N/N^2} = \frac{1}{\sqrt{N}}$$

as $N \rightarrow \infty$, $\frac{\Delta E}{E} \rightarrow 0$
while

$$\Delta E \propto \sqrt{N}$$

so the absolute displacement increases, but the relative displacement tends to 0.



1.3.3 Helmholtz Free Energy

If we compute the entropy with the ρ we found:

$$S = -k_B \int d\Gamma \rho(q, p) \ln \rho(q, p) = -k_B \int d\Gamma \frac{e^{-\beta H}}{\mathcal{Z}} (-\beta H - \ln \mathcal{Z}) = k_B \beta < H > + k_B \ln \mathcal{Z}$$

multiplying both sides by T:

$$-k_B T \ln \mathcal{Z} = < H > - TS$$

We finally find:

$$F = U - TS = \text{free energy}$$

so $\mathcal{Z} = e^{-\beta F}$: the canonical partition function is related to the free energy of the system.

$$\begin{aligned} \mathcal{Z} &= N k_B T \ln \left[V \left(\frac{2\pi m}{\beta} \right)^{3/2} \right] + \frac{3}{2} N k_B T \\ F &= -\frac{1}{\beta} \left[N \ln V + \frac{3}{2} N \ln (2\pi m) - \frac{3}{2} N \ln \beta \right] \end{aligned}$$

If we did things in the right way, we should find $-\frac{\partial F}{\partial V} = \text{pressure} = p$, check:

$$-\frac{\partial F}{\partial V} = \frac{1}{\beta} \frac{N}{V} = \frac{N k_B T}{V} = p$$

that is correct.

Disturbing fact We have two systems, with the same density and the same temperature.

$$\frac{N_1}{V_1} = \frac{N_2}{V_2}$$

If I put them in contact and I don't allow flow of particles, nothing forbids me to look at the system N_1+N_2, V_1+V_2 , but I find $S_{TOT} - (S_1 + S_2) = (N_1 + N_2)\ln(V_1 + V_2) - N_1\ln V_1 - N_2\ln V_2 \neq 0$ in general. This is known as **Gibbs paradox**. There's a problem related to the dependence of S from N. This will be solved by **correct Boltzmann counting**: whenever integrating over phase space, we should add a factor $\frac{1}{N!}$ (further information on paper by Robert H. Swendsen, *Gibbs' Paradox and the Definition of Entropy*, Carnegie Mellon University).

$$\int d\Gamma \rightarrow \frac{1}{N!} \int d\Gamma$$

This factor reduces the counting of the states. It is an **ad hoc correction**. When computing the entropy:

$$S^{correct} = S - \ln(N!) \simeq S - N\ln(N)$$

where we have used Stirling approximation. With this definition of entropy:

$$S_{1+2}^{correct} = S_{1+2} - (N_1 + N_2)\ln(N_1 + N_2)$$

$$S_1^{correct} + S_2^{correct} = S_1 + S_2 - N_1\ln(N_1) - N_2\ln(N_2)$$

so

$$S_{1+2} - (S_1 + S_2) = N_1\ln\left(\frac{V_1 + V_2}{N_1 + N_2}\right) - N_1\ln\left(\frac{V_1}{N_1}\right) - N_2\ln\left(\frac{V_2}{N_2}\right)$$

given that arguments of logarithms are the densities and they are all equal, we have $S_{1+2} - (S_1 + S_2) = 0$, which is correct.

1.4 Grand canonical ensemble

In the grand canonical ensemble we have $\langle N \rangle$ fixed, but N can vary, so we have $6N+1$ degrees of freedom.

$$S = -k_B \sum_N \int d\Gamma \rho_N \ln(\rho_N)$$

CONSTRAINTS:

- $\sum_N \int d\Gamma \rho_N = 1 \rightarrow \lambda$
- $\sum_N \int d\Gamma \rho_N H = \bar{E} \rightarrow \beta$
- $\sum_N \int d\Gamma \rho_N N = \bar{N} \rightarrow \beta\mu$

In order to find ρ of the grand canonical ensemble, we maximize S .

$$\tilde{F} = -k_B \sum_N \int d\Gamma \rho_N [\ln(\rho_N) + \lambda + \beta H_n - \beta \mu n]$$

We are adding a linear term in $\delta\rho$, so when we do the functional derivative and put it equal zero, we get:

$$\delta\tilde{F} = 0 = -k_B \sum_N \int d\Gamma \delta\rho [\ln(\rho) + 1 + \lambda + \beta H - \beta \mu N]$$

This should be true, no matter the choice of $\delta\rho$, so the integrand must be zero and we get:

$$\begin{aligned}\rho^* &= \frac{e^{-\beta[H_N - \mu N]}}{\mathcal{Z}} \\ \mathcal{Z} &= \sum_N \frac{1}{N!} \int d\Gamma e^{-\beta(H_N - \mu N)}\end{aligned}$$

We can define the free energy of the system by saying that:

$$\mathcal{Z} = e^{-\beta F}$$

and redefine

$$\rho = \frac{e^{-\beta[H_N - \mu N]}}{e^{-\beta F}}$$

Computing the entropy:

$$S = -k_B \sum_N \int d\Gamma \rho \ln(\rho) = k_B \sum_N \int d\Gamma \rho [\beta(H_N - \mu N) - \beta F] = k_B [\beta \langle H \rangle - \mu \beta \langle N \rangle - \beta F]$$

this means

$$S = \frac{\bar{E}}{T} - \frac{\mu \bar{N}}{T} - \frac{F}{T}$$

multiplying by T both sides and reorganizing terms:

$$\bar{F} = \bar{E} - \mu \bar{N} - TS$$

from this definition:

$$\mu = -\frac{\partial F}{\partial N}$$

μ is the chemical potential: energy associated with the introduction of one or more particles in the system. We have also:

$$\bar{N} = -\frac{\partial F}{\partial \mu}$$

We can check this:

$$\ln \mathcal{Z} = -\beta F \rightarrow F = -\frac{1}{\beta} \ln(\mathcal{Z})$$

and

$$\frac{\partial F}{\partial \mu} = \frac{1}{\beta \mathcal{Z}} \frac{\partial \mathcal{Z}}{\partial \mu}$$

given that $\mathcal{Z} \propto \sum_N \int d\Gamma e^{-\beta[H_N - \mu N]}$:

$$\frac{\partial \mathcal{Z}}{\partial \mu} \propto \sum_N \int d\Gamma e^{-\beta[H_N - \mu N]} \beta N \rightarrow -\frac{\partial F}{\partial \mu} = \langle N \rangle$$

Differentiating F with respect to one of the intensive thermodynamic quantities returns the average value of its conjugate quantity. F is a thermodynamic potential.

1.4.1 Ideal gas

$$\mathcal{Z}_{grandcanonical} = \sum_N \frac{1}{N!} \int d\Gamma e^{-\beta[H_N - \mu N]} = \sum_N \frac{1}{N!} e^{-\beta \mu N} \int d\Gamma e^{-\beta H_N}$$

This is equal to:

$$\mathcal{Z} = \sum_N \frac{1}{N!} e^{\beta \mu N} \left[V^N \left(\frac{2\pi m}{\beta} \right)^{\frac{3N}{2}} \right] = \sum_N \frac{1}{N!} x^N = e^x$$

where $x = e^{\beta \mu} V \left(\frac{2\pi m}{\beta} \right)^{\frac{3}{2}}$ and $e^{\beta \mu}$ is called *fugacity*.

We usually deal with $\log \mathcal{Z}$, that is:

$$\begin{aligned} \ln \mathcal{Z} &= zV \left(\frac{2\pi m}{\beta} \right)^{\frac{3}{2}} \\ \langle N \rangle &= -\frac{\partial F}{\partial \mu} = \frac{1}{\beta} \frac{\partial \ln \mathcal{Z}}{\partial \mu} = e^{\beta \mu} V \left(\frac{2\pi m}{\beta} \right)^{\frac{3}{2}} \end{aligned}$$

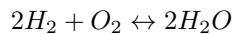
so we find:

$$\frac{\langle N \rangle}{V} = e^{\beta \mu} \left(\frac{2\pi m}{\beta} \right)^{\frac{3}{2}}$$

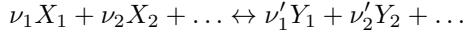
that is a relation between intensive quantities: density, fugacity, mass and temperature. Dependence on mass is particularly interesting: it comes from kinetic energy and is important to control equilibrium of chemical reactions.

1.4.2 Derivation of law of mass action

Let's take a chemical reaction as example:



We want to know what are the fractions of each species of molecules in the mixture at equilibrium. More generally we consider a reaction:



where ν_i are called *stoichiometric coefficients*, X_i are *volume concentrations*.

Free energy has to be as little as possible at equilibrium and there are little fluctuations of N, since it is a dynamic equilibrium, however these fluctuations can not be arbitrary.

$$\partial F = \sum_i \frac{\partial F}{\partial N_i} \delta N_i = - \sum_i \mu_i \delta N_i = 0 \text{ at equilibrium}$$

so:

$$\sum_i \mu_i \delta N_i = 0$$

μ_i are known, but δN_i are tricky: they can't be arbitrary, but must satisfy $\frac{\delta N_i}{\pm \nu_i} = \text{constant} = \delta n$. So we have:

$$\sum_i \mu_i v_i = \sum_j \mu_j v_j$$

where the right hand side refers to products and left hand side to reagents and this is the condition for equilibrium.

If we take our first example:

$$2\mu_{H_2} + 1\mu_{O_2} = 2\mu_{H_2O}$$

multiply both sides by β and remember $\beta\mu = \ln \left(\left(\frac{2\pi m}{\beta} \right)^{\frac{3}{2}} \frac{1}{V} \right)$:

$$2\ln \left(\left(\frac{2\pi m_{H_2}}{\beta} \right)^{\frac{3}{2}} \frac{1}{V_{H_2}} \right) + \ln \left(\left(\frac{2\pi m_{O_2}}{\beta} \right)^{\frac{3}{2}} \frac{1}{V_{O_2}} \right) = 2\ln \left(\left(\frac{2\pi m_{H_2O}}{\beta} \right)^{\frac{3}{2}} \frac{1}{V_{H_2O}} \right)$$

using properties of logarithms:

$$\frac{V_{H_2O}^2}{V_{H_2}^2 V_{O_2}} \propto \frac{m_{H_2}^3 m_{O_2}^{\frac{3}{2}}}{m_{H_2O}^3}$$

that is the **law of mass action**. It is very important because shows that we can shift chemical equilibrium towards products or reagents by changing the mass (using isotopes). This comes from the kinetic contribution to the partition function \mathcal{Z} .

Capitolo 2

Molecular Dynamics

2.1 Hamilton Formalism

q = position

\dot{q} = velocity

p = momentum

\dot{p} = force

These are all vectors of $3N$ components in the phase space (x, y, z components for each of N particles). **Hamilton equations:**

$$\begin{cases} \dot{q} = \frac{\partial H}{\partial p} \\ \dot{p} = -\frac{\partial H}{\partial q} \end{cases}$$

We make some **non general assumptions** on H , in order to be able to do computations:

$$H(q, p) = K(p) + U(q)$$

$$K(p) = \sum_i \frac{p_i^2}{2m_i}$$

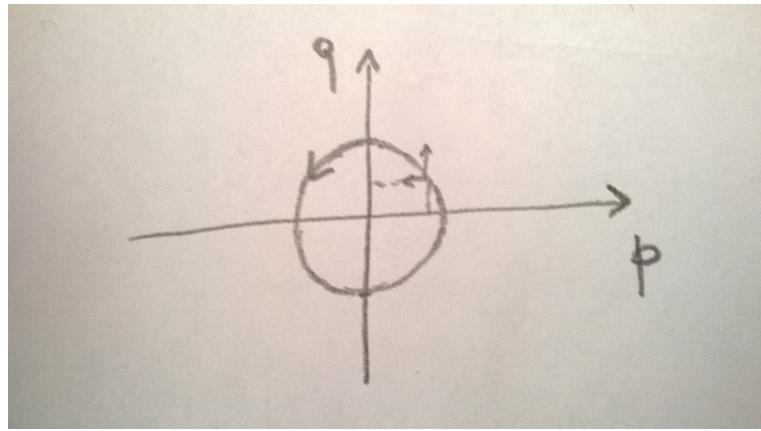
that are: Hamiltonian is the sum of a term K depending only on p and a term U depending only on q ; kinetic energy K is quadratic in p . With these assumptions we have:

$$\begin{cases} \dot{q} = \frac{\partial K}{\partial p} = \frac{p}{m} \\ \dot{p} = -\frac{\partial U}{\partial q} = f \end{cases}$$

that are two first order differential equations. Or we can state Newton Law:

$$\ddot{q} = \frac{f}{m}$$

that is a single second order differential equation.



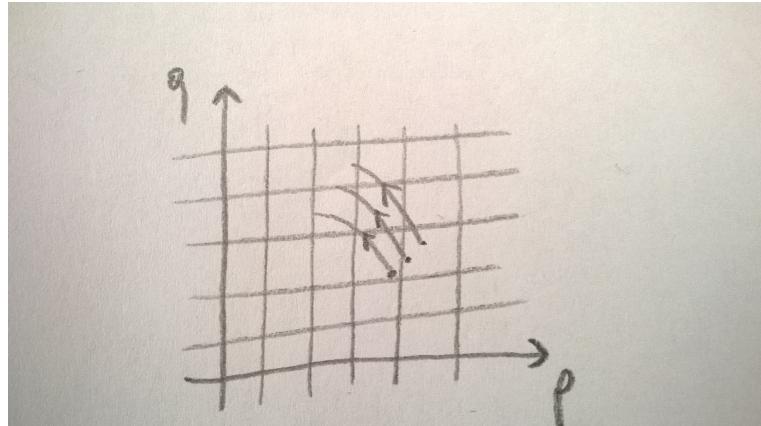
Harmonic Oscillator

$$H = \frac{p^2}{2m} + \frac{1}{2}Kq^2$$

if we assume $m = 1$, $k = 1$, then $H = \frac{p^2}{2} + \frac{q^2}{2}$ and

$$\begin{aligned}\dot{p} &= -q \\ \dot{q} &= p\end{aligned}$$

2.2 Liouville Formalism



We define points $x = (q, p)$ and we can define a velocity \dot{x} , that is not the physical velocity, but the velocity of points in phase space.

$$\dot{x} = v(x)$$

then we can write:

$$\dot{\rho}(x, t) = -\vec{\nabla} \cdot \vec{J}(x, t) = -\vec{\nabla} \cdot (\rho \vec{v}) = -\frac{\partial}{\partial p}(\rho \dot{p}) - \frac{\partial}{\partial q}(\rho \dot{q})$$

$$-\frac{\partial}{\partial p}(\rho \dot{p}) - \frac{\partial}{\partial q}(\rho \dot{q}) = \frac{\partial}{\partial p} \left(\rho \frac{\partial H}{\partial q} \right) - \frac{\partial}{\partial q} \left(\rho \frac{\partial H}{\partial p} \right) = \frac{\partial \rho}{\partial p} \frac{\partial H}{\partial q} + \rho \frac{\partial^2 H}{\partial q \partial p} - \frac{\partial \rho}{\partial q} \frac{\partial H}{\partial p} - \frac{\partial^2 H}{\partial q \partial p}$$

and finally:

$$\dot{\rho}(x, t) = \frac{\partial \rho}{\partial p} \frac{\partial H}{\partial q} - \frac{\partial \rho}{\partial q} \frac{\partial H}{\partial p} = -\hat{L} \cdot \rho$$

where \hat{L} is called the **Liouville operator**.

$$\hat{L} = \hat{L}_p + \hat{L}_q$$

with $\hat{L}_p = -\frac{\partial H}{\partial q} \frac{\partial}{\partial p}$ and $\hat{L}_q = \frac{\partial H}{\partial p} \frac{\partial}{\partial q}$. If we solve $\dot{\rho} = -\hat{L}\rho$ we find:

$$\rho(t + \Delta t) = e^{-\Delta t \hat{L}} \rho(t)$$

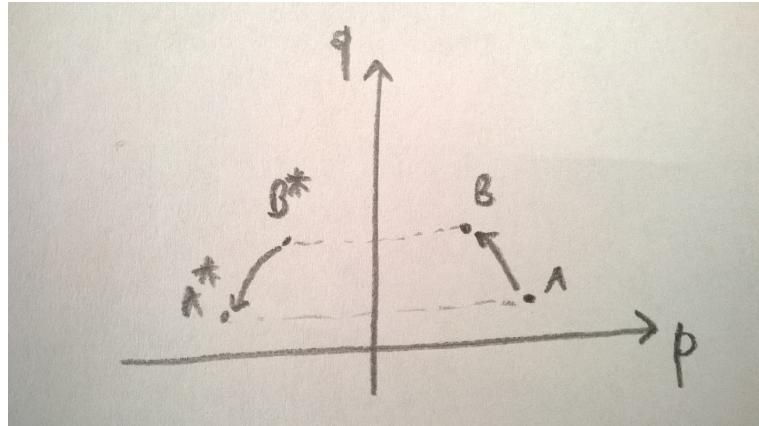
that tells how an ensemble of trajectories evolves with time. If we take the total derivative of $\rho(x, t)$, we find:

Properties of trajectories

- **Time reversal trajectory:** the time reversal trajectory satisfies the same Hamilton equation as the original one (requires non general assumptions we made before, it's not true in presence of a magnetic field for example).

$$(p(t), q(t)) \rightarrow (-p(-t), q(-t)) = (p*(t), q*(t))$$

$$\begin{cases} \frac{\partial q*(t)}{\partial t} = \frac{\partial q(-t)}{\partial t} = -\frac{\partial q(-t)}{\partial(-t)} = -\frac{\partial H}{\partial p} = \frac{\partial H}{\partial(-p)} = \frac{\partial H}{\partial p*} \\ \frac{\partial p*(t)}{\partial t} = -\frac{\partial p(-t)}{\partial t} = \frac{\partial p(-t)}{\partial(-t)} = -\frac{\partial H}{\partial q} = -\frac{\partial H}{\partial q*} \end{cases}$$



- **Energy conservation:** we assume H depends on t because of $p(t)$ and $q(t)$ and nothing else, then $\frac{\partial H}{\partial t} = 0$, so total derivative of H is:

$$\frac{dH}{dt} = \frac{\partial H}{\partial p} \frac{\partial p}{\partial t} + \frac{\partial H}{\partial q} \frac{\partial q}{\partial t} = \dot{q}\dot{p} - \dot{p}\dot{q} = 0$$

This is always true.

- **Incompressibility of flux:**

$$\frac{d\rho(x,t)}{dt} = \frac{\partial\rho(x,t)}{\partial t} + \frac{\partial\rho(x,t)}{\partial x} \cdot \dot{x} = \frac{\partial\rho}{\partial t} + \frac{\partial\rho}{\partial q} \cdot \dot{q} + \frac{\partial\rho}{\partial p} \cdot \dot{p} = \frac{\partial\rho}{\partial t} + \frac{\partial}{\partial x}(\rho\dot{x}) - \rho\frac{\partial\dot{x}}{\partial x}$$

$$\frac{\partial\rho}{\partial t} + \frac{\partial}{\partial x}(\rho\dot{x}) - \rho\frac{\partial\dot{x}}{\partial x} = -\nabla \cdot (\rho\dot{x}) + \nabla \cdot (\rho\dot{x}) - \rho\frac{\partial\dot{x}}{\partial x} = -\rho\frac{\partial\dot{p}}{\partial p} - \rho\frac{\partial\dot{q}}{\partial q} = -\rho\frac{\partial^2 H}{\partial p \partial q} - \rho\frac{\partial^2 H}{\partial q \partial p} = 0$$

so:

$$\frac{d\rho(x,t)}{dt} = 0$$

that is the **Liouville's Theorem**: the flux of the system is like the one of an incompressible fluid.

2.3 Verlet Algorithm

We have $\ddot{q} = \frac{f}{m}$ and we want to integrate it. We know initial conditions and we want to compute position and velocity after some time. In order to do this, we can do a Taylor expansion in time:

$$q(+\Delta t) = q(t) + \dot{q}(t)\Delta t + \ddot{q}(t)\frac{(\Delta t)^2}{2} + \text{higher order terms}$$

since we know only \ddot{q} we have to do some approximation, we write another Taylor expansion:

$$q(-\Delta t) = q(t) - \dot{q}(t)\Delta t + \ddot{q}(t)\frac{(\Delta t)^2}{2} + \text{higher order terms}$$

adding one another:

$$q(+\Delta t) + q(-\Delta t) = 2q(t) + \dot{q}(t)(\Delta t)^2 + o(\Delta t)^4$$

since the term of order 3 is canceled. Finally we get:

$$q(+\Delta t) = 2q(t) - q(t - \Delta t) + \frac{f(t)}{m}(\Delta t)^2$$

This is a numerical approximation to the original differential equation, with an error $o(\Delta t^4)$. It's called **Verlet equation**.

In this algorithm we don't need velocity, instead we store position at previous timestep and position at current timestep.

2.3.1 Pseudocode

```
//initialize
q = ...
qold = q - v*deltat - f*(deltat^2)/m
qnew = 0

for (i = 0; i < nsteps; i++) {
    qnew = 2*q - qold + (f/m)*(deltat^2)
    qold = q
    q = qnew
}
```

Since q is a vector, this computation can be parallelized for every component. This algorithm satisfies time reversibility: there could be errors due to representation of finite numbers of digits, but these are small and become important only for long time intervals.

$$q(t - \Delta t) = 2q(t) - q(t + \Delta t) + \frac{f(t)}{m} \Delta t^2$$

We can't say if energy is conserved because there is no velocity, so we can't compute it.

2.4 Velocity Verlet Algorithm

We can derive a better algorithm from $\dot{\rho} = -\hat{L}\rho$.

We have $\rho(t + \Delta t) = e^{-\Delta t \hat{L}} \rho(t)$, with $\hat{L} = \hat{L}_q + \hat{L}_p$. And:

$$\begin{cases} \hat{L}_q = f \frac{\partial}{\partial p} \\ \hat{L}_p = \frac{p}{m} \frac{\partial}{\partial q} \end{cases}$$

We have to compute the exponential of an operator and we can do that by Taylor expansion:

$$e^{-\Delta t \hat{L}_q} = 1 - \Delta t \hat{L}_q + \frac{\Delta t^2}{2} \hat{L}_q^2 + \dots = 1 - \Delta t \frac{p}{m} \frac{\partial}{\partial q} + \frac{\Delta t^2}{2} \frac{p^2}{m^2} \frac{\partial^2}{\partial q^2} + \dots$$

What happens if I apply it to a function?

$$e^{-\Delta t \hat{L}_q} \rho(q, p, t) = (1 - \Delta t \frac{p}{m} \frac{\partial}{\partial q} + \frac{\Delta t^2}{2} \frac{p^2}{m^2} \frac{\partial^2}{\partial q^2} + \dots) \rho(q, p, t) = \rho(q - \frac{p}{m} \Delta t, p, t)$$

the effect of the operator is to take the density and shift the first argument by an amount that is $\frac{p}{m} \Delta t = \dot{q} \Delta t$. It's the propagation of this equation ignoring $\dot{p} = f$.

We can do the same for $e^{-\Delta t \hat{L}_p} \rho(q, p, t)$, that propagates the other Hamilton equation.

Now we know how to propagate them singularly, but not how to propagate both of them at the same time.

If $[\hat{L}_p, \hat{L}_q] \neq 0$, $e^{-\Delta t (\hat{L}_q)} e^{-\Delta t \hat{L}_p} \neq e^{-\Delta t \hat{L}}$ and this is the case, in fact they do not commute, so applying first \hat{L}_p and then \hat{L}_q is not the same as doing the inverse. We propose different expressions for $\lim_{\lambda \rightarrow 0} e^{\lambda(A+B)} \neq e^{\lambda A} e^{\lambda B} \neq e^{\lambda B} e^{\lambda A}$.

For example doing Taylor expansion of our first proposal:

$$e^{\lambda A} e^{\lambda B} = (1 + \lambda A + (\frac{\lambda^2 A^2}{2} + o(\lambda^2))) (1 + \lambda B + (\frac{\lambda^2 B^2}{2} + o(\lambda^2))) = 1 + \lambda(A+B) + \lambda^2 (\frac{A^2}{2} \frac{B^2}{2} + AB) + o(\lambda^3)$$

while

$$e^{\lambda(A+B)} = 1 + \lambda(A+B) + \frac{\lambda^2}{2} (A^2 + B^2 + AB + BA) + o(\lambda^3)$$

so:

$$e^{\lambda(A+B)} - e^{\lambda A} e^{\lambda B} = \frac{\lambda^2}{2} [B, A] + o(\lambda^3)$$

For the second proposal it's the same, it only changes the sign of the error. There is a much better choice that makes second order error disappear.

Best choice: Trotter splitting

$$e^{\lambda(A+B)} \sim e^{\frac{\lambda A}{2}} e^{\lambda B} e^{\frac{\lambda A}{2}}$$

$$e^{\lambda(A+B)} \sim e^{\frac{\lambda A}{2}} e^{\lambda B} e^{\frac{\lambda A}{2}} + [A, B] \cdot o(\lambda^3)$$

equivalently one can use $e^{\frac{\lambda B}{2}} e^{\lambda A} e^{\frac{\lambda B}{2}}$. So we can write:

$$e^{-\Delta \hat{L}} \simeq e^{-\Delta \frac{L_p}{2}} e^{-\Delta \hat{L}_q} e^{-\Delta \frac{L_p}{2}}$$

2.4.1 Pseudocode

```
//initialize
q = ...
p = ...

for (i = 0; i < nsteps; i++) {
    p = p + f(q)*(deltat/2)
    q = q + (p/m)*deltat
    p = p + f(q)*(deltat/2)
    printf (p^2)/2m + U
}
```

This algorithm is called **Velocity Verlet**. Actually, it's like we are computing $\dot{x} = v(x)$ assuming $v(x) = v_1(x) + v_2(x)$ and then solving:

$$\dot{x} = v_1(x) \text{ ignoring } v_2(x) \text{ for } \frac{\Delta t}{2}$$

then

$$\dot{x} = v_2(x) \text{ ignoring } v_1(x) \text{ for } \Delta t$$

and again

$$\dot{x} = v_1(x) \text{ ignoring } v_2(x) \text{ for } \frac{\Delta t}{2}$$

similarly to trapezoid rule for integration.

When solving for v_1 , Hamilton equations are:

$$\begin{cases} \dot{p} = f \\ \dot{q} = 0 \end{cases}$$

and consider $H = U$, kinetic energy disappears since $\dot{q} = 0$. It's like the limit for mass going to infinity. When solving for v_2 :

$$\begin{cases} \dot{p} = 0 \\ \dot{q} = \frac{p}{m} \end{cases}$$

and consider $H = K$, potential energy disappears since \dot{p} disappear. It's the limit for force going to zero.

We can do this because we assumed that K depends only on p and U only on q , otherwise it would be a mess.

Velocity Verlet satisfies time-reversibility and incompressibility of flux, but not energy conservation. In Velocity Verlet algorithm the energy is almost conserved for $\Delta t \rightarrow 0$.

If we want a compact notation for this algorithm:

$$\begin{cases} q(t + \Delta t) = q(t) + \frac{p(t)}{m} \Delta t + \frac{f(q)}{m} \Delta t + \frac{f(q)}{m} \frac{\Delta t^2}{2} \\ p(t + \Delta t) = p(t) + \frac{f(q(t)) \Delta t + f(q(t + \Delta t)) \Delta t}{2} \end{cases}$$

We can implement a similar algorithm called **Position Verlet**, that comes from:

$$e^{-\Delta \hat{L}} \simeq e^{-\Delta \frac{\hat{L}_q}{2}} e^{-\Delta \hat{L}_p} e^{-\Delta \frac{\hat{L}_q}{2}}$$

This isn't used because it needs to compute the force twice per cycle, while the first one only once and one at the beginning. Computing force is the most computationally expensive calculation, so one tries to reduce the times it is computed.

An advantage of Velocity Verlet over simple Verlet algorithm is that it has less roundoff errors, since we are not taking any difference. This problem becomes evident if we are using floating points number (32 bits) and simulating a very long time interval.

Example: harmonic oscillator Simplified notation: $\Delta t \rightarrow h$, $q(t) \rightarrow q$, $q(t + \Delta t) \rightarrow q'$, $f = -q$ and $H = \frac{p^2}{2} + \frac{q^2}{2}$
Then:

$$\begin{cases} q' = q + ph - q \frac{h^2}{2} \\ p' = p - \frac{q+q'}{2} h \rightarrow p' = p(1 - \frac{h^2}{2}) + (-h + \frac{h^3}{4})q \end{cases}$$

so, transformation matrix is:

$$\begin{bmatrix} 1 - \frac{h^2}{2} & h \\ -h + \frac{h^3}{4} & 1 - \frac{h^2}{2} \end{bmatrix}$$

That as determinant equal to 1, but isn't a rotation matrix, since it's not of the form

$$\begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix}$$

but has an error proportional to $o(\frac{h^3}{4})$, so energy is not conserved. If we compute the n-th power of the matrix with $n \rightarrow \infty$, we can know if the energy will be finite or will explode to infinity.

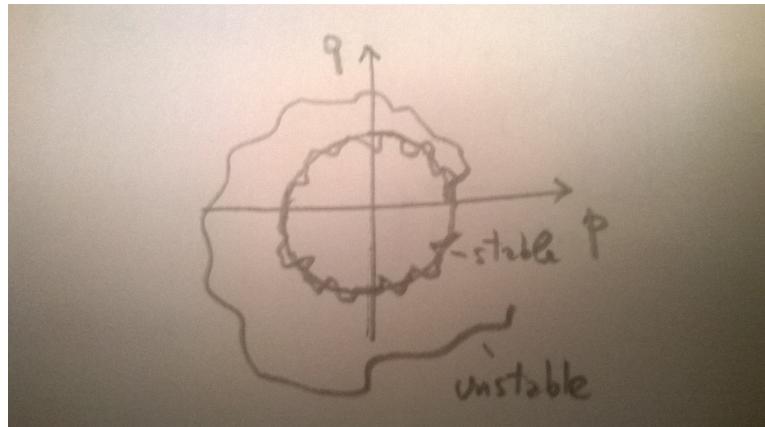
We diagonalize the matrix and compute the eigenvalues:

$$(1 - \frac{h^2}{2} - \lambda)^2 - h(\frac{h^3}{4} - h) = 0 \rightarrow \lambda_{1,2} = 1 - \frac{h^2}{2} \pm \frac{h}{2} \sqrt{-4 + h^2}$$

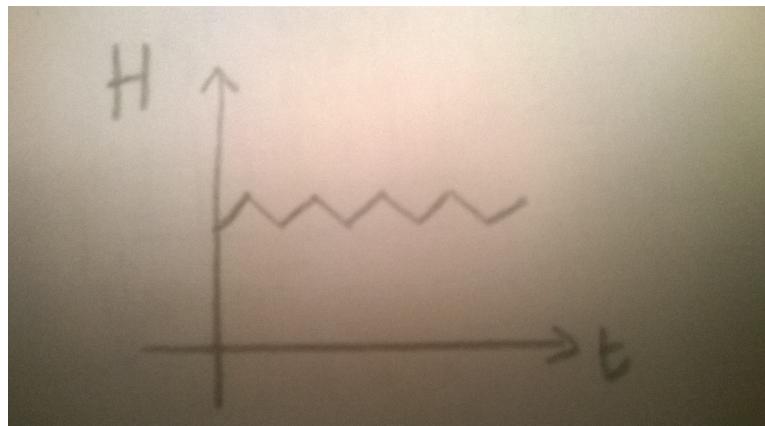
so we can have both real or both complex eigenvalues.

If $h > 2$, we have two real eigenvalues $\lambda_1 > 1, \lambda_2 < 1$, so the energy is going to explode.

If $h < 2$, we have two complex conjugate eigenvalues and the simulation will be stable. Actually given a period, we find that we have $\Delta t_{max} < \frac{T}{\pi}$ in order to have a stable simulation, in our case was $T = 6$.



If we draw $HvsT$:

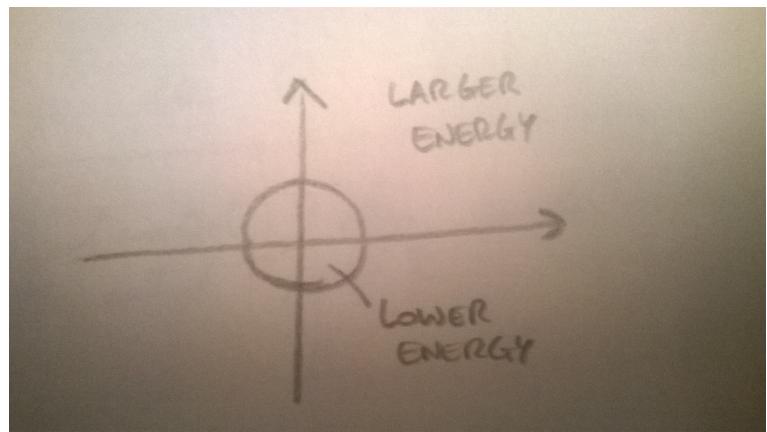


Every step we make an error, that comes from using an approximate hamiltonian, but errors at successive steps are anticorrelated. This is in a certain way correlated to time reversibility: if energy diverges, we expect energy to increase or decrease with the same likelihood.

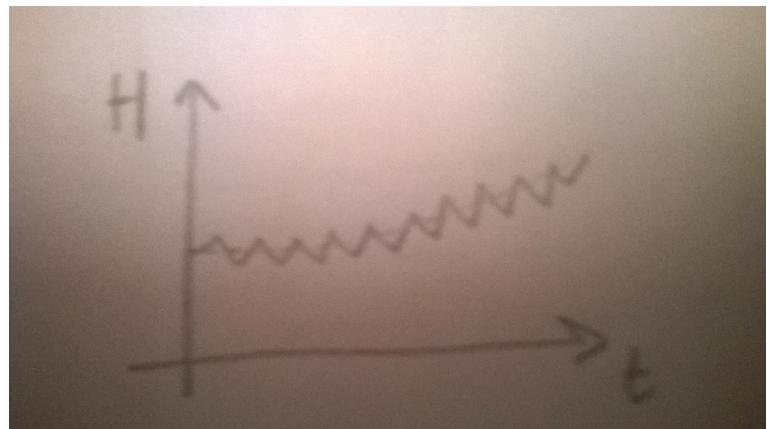
So far Δt is the only parameter to choose and we do it in a way for which the energy is conserved.

If we have more than one harmonic term, we choose $\Delta t < \frac{T_{min}}{\pi}$, that is: we choose the timestep in relation to the lowest period of all. We make a choice for the worst case.

If we have unharmonicity in the potential, it will introduce a drift up or down in the energy. If this is a random error, H will increase, since it's more likely that the random point of phase space will be at a higher energy than at a lower one.



If the system is quasi-harmonic the drift will be very small. Decreasing Δt makes fluctuations and slope of the drift decrease.

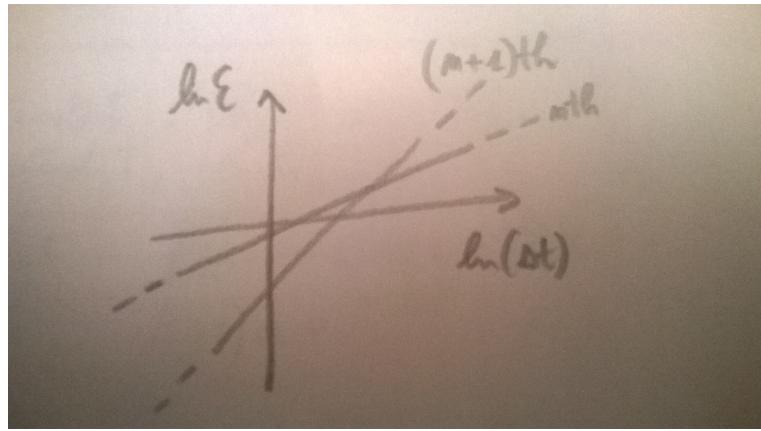


Usually it's said that fluctuations have to be $< k_B T$, but big systems have larger fluctuations and this is not what it's done.

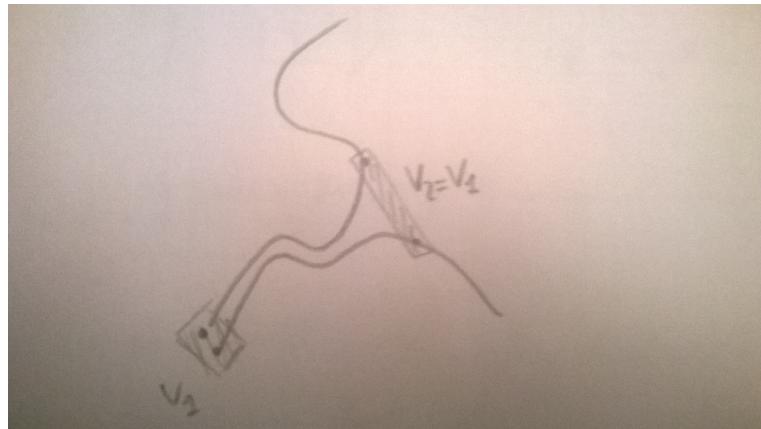
A useful thing is to look at drift and fluctuations to compare two simulations and say if they're equally accurate.

Fluctuations scale with \sqrt{N} , while drift scales linearly with N .

In molecular dynamics higher order integrators aren't used because $\epsilon \propto \Delta t^n$, so $\ln \epsilon \propto N \ln \Delta t$, if we plot it:



Higher order algorithms mean that error becomes smaller faster when decreasing Δt , but gets bigger faster when increasing timestep. Since in molecular dynamics quite large Δt are often used, integrators of higher order are useless to this purpose, higher order algorithms are used for very short Δt in computations like predicting trajectories of planets or artificial satellites. Another reason is that we already have approximate forces, approximate initial conditions, approximate equation... All of these introduce a much larger error than the one introduced by higher order integrators, so there's no point in using them. Often we have systems with many degrees of freedom that lead to non linear equations, in this case small differences in initial condition can lead to a very different behavior and a small mistake can become much larger.



2.5 Leap frog

2.5.1 Pseudocode

```
//initialize
q = ...
p = ...
for (i = 0; i < nsteps; i++) {
    q = q + (p/m)*deltat
    p = p - (q/m)*deltat
```

```

    p = p + f(q)*deltat
}

```

This is the algorithm obtained by using $e^A e^B$, but with a different interpretation: p at first step is $p(t + \Delta r)$ instead of $p(t)$. If we compare Velocity Verlet and Leap frog, we have:

$$q(t + \Delta t) = q(t) + \frac{p(t)}{m} \Delta t + \frac{f(t)}{2m} \Delta t^2$$

$$q(t + \Delta t) = q(t) + \frac{p(t + \frac{\Delta t}{2})}{m} \Delta t$$

where $p(t + \frac{\Delta t}{2})$ is the same velocity we get after the first update in Velocity Verlet. Actually Leap Frog is the result of combining the last operation of one step and the first of the successive step of Velocity Verlet algorithm.

There isn't a big difference between the two, because they compute forces (that is the most expensive computation and actually takes about 97% of the time of execution) the same number of times. In special cases where forces are relatively simple to compute and other operation weight more, Leap Frog is slightly faster. The same argument applies if we're trying to optimize at best the code: we sacrifice readability for a slightly better performance.

A difference between the two algorithm is that in Velocity Verlet positions and momenta are consistent $(p(t), q(t))$, while in Leap frog they're not $(p(t + \Delta t), q(t + \frac{3}{2}\Delta t))$.

2.6 Different types of ensembles and Properties of canonical ensemble

Often it is difficult to make an experiment on a single molecule for technical reasons, and it's simpler do an experiment on an ensemble, that gives an amplified signal to measure. If we have to compare numerical simulations and experiment what we do is actually comparing an average measure on an ensemble with a time average in the simulation.

We have different types of ensembles, that are named by what remains constant: NVE (micro-canonical), NVT (canonical), μ VT (grand canonical) and then there are others like NPT, NPH or μ VE. These are used according to what the experimentalist can control.

Properties of canonical ensemble

$$\rho(x) = \frac{e^{-\beta H(x)}}{\mathcal{Z}} \text{ with } \mathcal{Z} = \int dx e^{-\beta H(x)}$$

$$\langle A \rangle = \int dx P(x) A(x)$$

$$\frac{\partial \langle A \rangle}{\partial T} = -\frac{1}{k_B T^2} (-\langle HA \rangle - \langle H \rangle \langle A \rangle)$$

where $\langle A \rangle$ is to be considered $\langle A \rangle_T$, A doesn't depend on T but the average of A does.

$$\frac{\partial \langle H \rangle}{\partial T} = \frac{1}{k_B T^2} (\langle H^2 \rangle - \langle H \rangle^2)$$

$$H = K + U$$

$$K = \sum_i \frac{|p_i|^2}{2m_i}$$

$$\langle K \rangle_T = \frac{N_f k_B T}{2} \quad \text{Nf is the number of degrees of freedom}$$

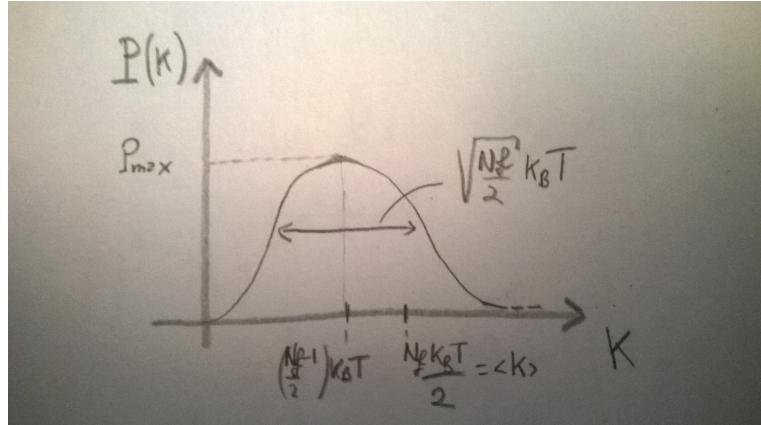
$$\frac{\partial \langle K \rangle_T}{\partial T} = \frac{N_f k_B}{2}$$

$$\frac{\partial \langle U \rangle}{\partial T} = \frac{1}{k_B T^2} (\langle HU \rangle - \langle H \rangle \langle U \rangle) = \frac{1}{k_B T^2} (\langle U^2 \rangle - \langle U \rangle^2) \quad \text{fluctuation of potential energy}$$

To compute c_V , you need to compute $\frac{\partial \langle K \rangle}{\partial T}$ and $\frac{\partial \langle U \rangle}{\partial T}$. To compute the first one, you just need to know how many atoms are there, the second one is more difficult, because it depends on position.

Notice that at constant T, $K \neq \frac{k_B T}{2}$, in fact it should be equal on average: $\langle K \rangle \neq \frac{k_B T}{2}$.

$$\langle K^2 \rangle - \langle K \rangle^2 = k_B T^2 \frac{\partial \langle K \rangle}{\partial T} = \frac{N_f k_B^2 T^2}{2} \quad \text{in units of } k_B T: \frac{N_f}{2}$$



$$\sqrt{\frac{\langle K^2 \rangle - \langle K \rangle^2}{\langle K \rangle}} = \sqrt{\frac{2}{N_f}}$$

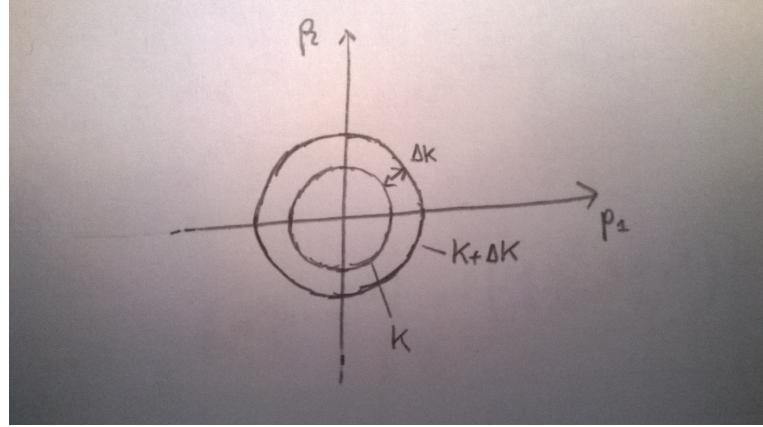
many degrees of freedom mean a more peaked distribution:

$$\langle K \rangle \propto N, \sigma(K) \propto \sqrt{N}$$

We can derive the function $\mathcal{P}(K)$:

$$\mathcal{P}(K) \propto \int dp \delta\left(\sum_i \frac{p_i^2}{2m_i}\right) e^{-\beta \sum_i \frac{p_i^2}{2m_i}} \propto e^{-\beta K} \Omega(K)$$

where $e^{-\beta K}$ says that large values of kinetic energy are not likely and $\Omega(K)$ how much the number of states changes with K .



We want to do this in $3N$ dimension ($= N_f$): width of the skin (in p) per surface of inner ipersphere.

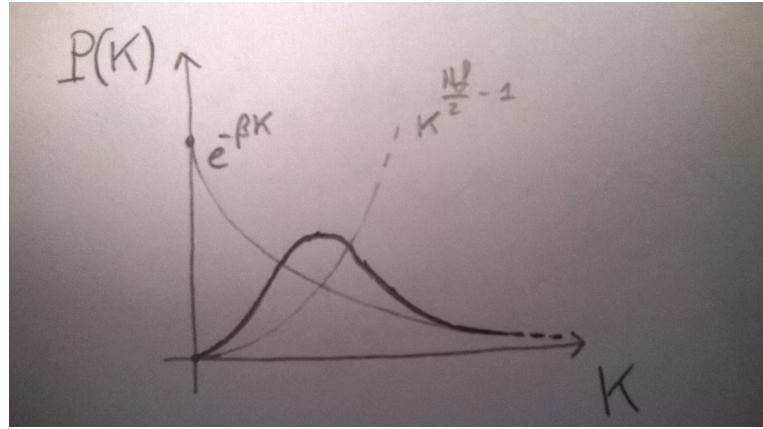
$$\Omega(K) \propto (\sqrt{K})^{N_f-1} \frac{1}{\sqrt{K}} = K^{\frac{N_f-1}{2}-\frac{1}{2}} = K^{\frac{N_f}{2}-1}$$

where \sqrt{K} is the radius and $\sqrt{1/K}$ is the width of the skin, since:

$$\frac{\partial K}{\partial p} \Delta|p| = \Delta K \rightarrow \Delta K \propto \Delta|p| \rightarrow \Delta p \propto \frac{\Delta K}{K} \propto \frac{\Delta K}{\sqrt{K}}$$

so

$$\mathcal{P}(K) \propto K^{\frac{N_f}{2}-1} e^{-\beta K}$$



Doing the first derivative of $\mathcal{P}(K)$ and imposing it's equal to zero, we can find K that maximizes \mathcal{P} , that is $K = \left(\frac{N_f}{2} - 1\right) k_B T$.
This analysis is valid only for K , but irrespectively of the system, given that it's canonical. This

holds also for U for the harmonic oscillator, because U has the same functional form of K in that case. So, **for the harmonic oscillator**:

$$\mathcal{P}(U) \propto U^{\frac{N_f}{2}-1} e^{-\beta U}$$

$$\mathcal{P}(H) \propto H^{N_f-1} e^{-\beta H} \text{ since degrees of freedom are } 6N$$

2.7 Monte Carlo Method

Monte Carlo integration is a technique for numerical integration using random numbers. It is a particular Monte Carlo method that numerically computes a definite integral. While other algorithms usually evaluate the integrand at a regular grid, Monte Carlo randomly choose points at which the integrand is evaluated. This method is particularly useful for higher-dimensional integrals. Monte Carlo is useful also when the number of points that are contributing to the integral is a tiny fraction.

Crude Monte Carlo sampling is not efficient if most of the points won't be relevant: most of the calculations will be useless.

2.7.1 Markov Chain Monte Carlo

This is a method that builds a list of points such that satisfies a prescribed distribution, even if it has a very difficult analytical form. A **Markov chain** is a random process that undergoes transitions from one state to another on a state space. It must possess a property that is usually characterized as *memorylessness*: **the probability distribution of the next state depends only on the current state and not on the sequence of events that preceded it.**

- Choose an arbitrary point x_0 to be the first sample
- propose a move
- check if it satisfies some rule: if yes, evaluate the function, otherwise, propose a new move.

Rules to be satisfied:

- **reversibility**: $P(A \rightarrow B) = P(B \rightarrow A)$;
- **ergodicity**: every point of the space can be reached in a given number of steps (tricky for continuous space, but you can define a neighborhood);
- **stationarity**: $P(X_i)$ does not depend on i, equivalently $\bar{P} = \pi \bar{P}$: \bar{P} is a stationary matrix associated to the transition matrix π , a right eigenvector of it, with eigenvalue 1.

Transition matrix π is a stochastic matrix:

$$\sum_i \pi_{ij} = 1 \text{ probability of going from j to i}$$

$\pi_{ij} \geq 0$ equality means transition is forbidden, it can happen

$$P_i^{NEW} = P_i^{OLD} + \sum_{j \neq i} \pi_{ij} P_j - \sum_{j \neq i} \pi_{ji} P_i = P_i^{OLD} + \sum_j \pi_{ij} P_j - \sum_j \pi_{ji} P_i^{OLD}$$

since $\sum_j \pi_{ji} = 1$:

$$P_i^{NEW} = \sum_j \pi_{ij} P_j$$

Another way of stating the same condition:

$$\sum_j \pi_{ij} \bar{P}_j = \sum_i \pi_{ji} \bar{P}_i$$

that means that the probability of going from i to j and from j to i is the same, so P is stationary. This condition is called **balance**.

One can have a more strict condition:

$$\sum_{j \neq i} (\pi_{ij} \bar{P}_j - \pi_{ji} \bar{P}_i) = 0$$

so:

$$\pi_{ij} \bar{P}_j = \pi_{ji} \bar{P}_i$$

that is called **detailed balance** or also microreversibility. Detailed balance implies balance: it's a sufficient but not necessary condition. In this case every single transition is compensated by an opposite transition.

Hamilton equations satisfy properties very close to detailed balance. There are notable exceptions, such as non equilibrium systems, in which this is not true: if a system is out of equilibrium, it satisfies only balance property. By the way detailed balance is not necessary in order to implement Monte Carlo algorithm, is just a way to do it.

An interesting thing we find from detailed balance is:

$$\frac{\pi_{ij}}{\pi_{ji}} = \frac{\bar{P}_i}{\bar{P}_j}$$

The ratio between one move and the inverse should be equal to the ratio of P of two end states. Actually you need only to know this rate to implement a correct algorithm.

Examples: two and three states systems

- **two state system:**

$$\bar{P}_1 = x, \bar{P}_2 = y$$

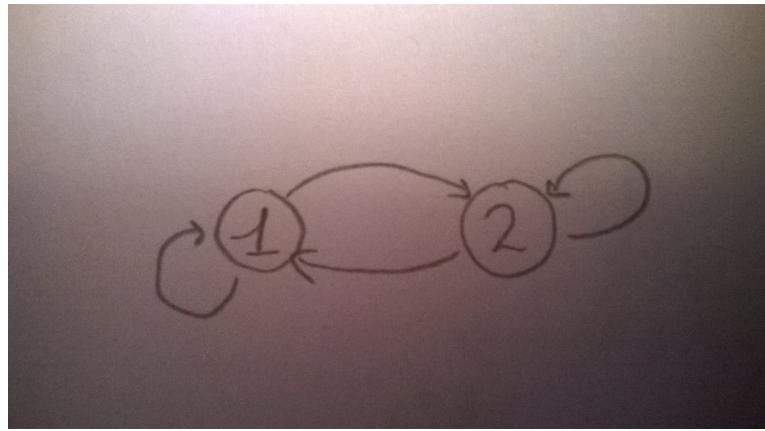
we have

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}$$

since π must be a stochastic matrix:

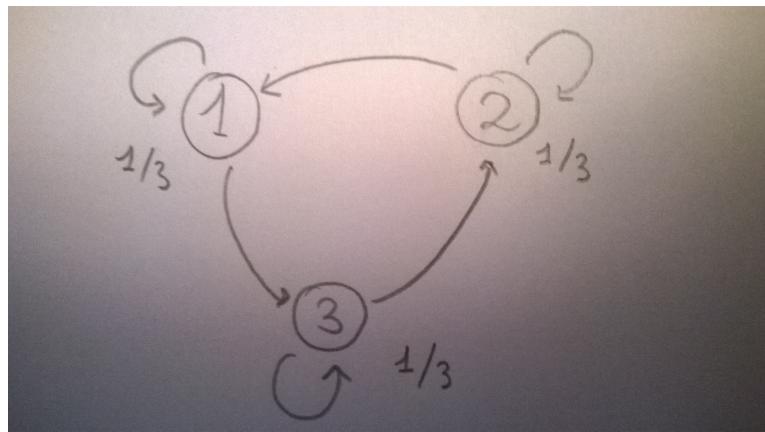
$$\begin{pmatrix} 1-c & b \\ c & 1-b \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}$$

that gives $cx = by$, that is: $\frac{c}{b} = \frac{y}{x}$

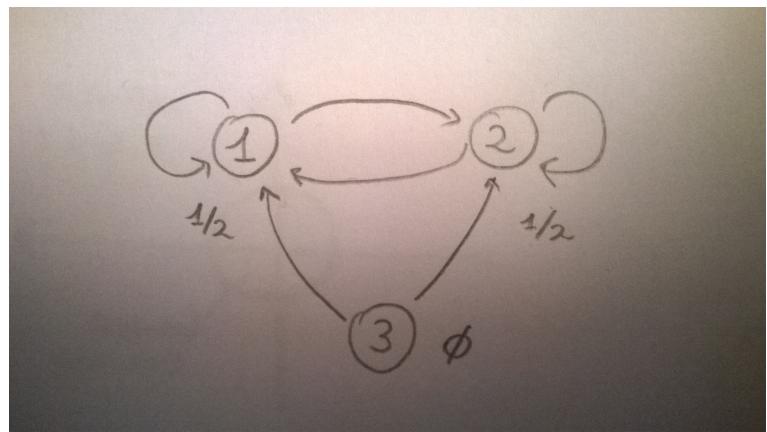


All arrows entering compensate with arrows exiting. In this case balance implies detailed balance (actually it's the same thing, since there is no sum to do).

- **three states system:**

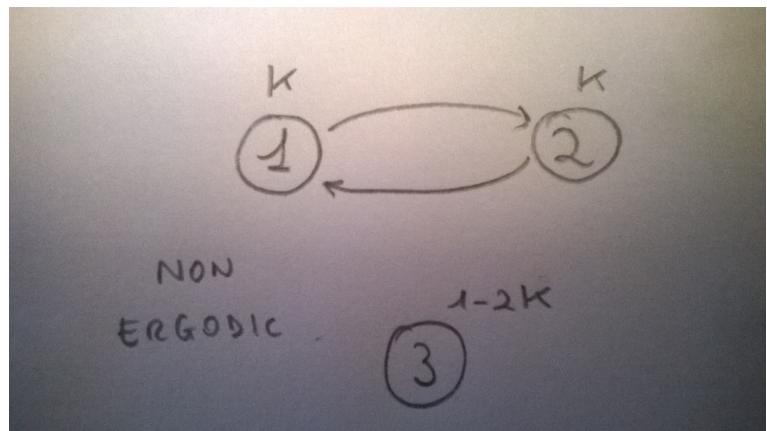


In this case detailed balance isn't satisfied, but balance is. There is a net current different from zero.



$$\pi = \begin{pmatrix} 1-x & x & x \\ x & 1-x & x \\ 0 & 0 & 1-2x \end{pmatrix}$$

The system in its entirety is not ergodic, but if we consider only 1 and 2 it is. Actually the ergodicity we're interested in is the condition in which the choice of initial state doesn't influence the final state of the system.



$$\pi = \begin{pmatrix} 1-x & x & 0 \\ x & 1-x & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

This is a non-ergodic system. It's like putting together two systems that don't see each other. This is a nasty case: if I start a Monte Carlo simulation from state 3 I will have convergence, but also if I start from state 1 or 2, however the results will be different. This is a very difficult mistake to find and we have to be very careful.

Physical meaning: **detailed balance means there is no net current and describes equilibrium, while balance means there is a net current and describes non equilibrium**

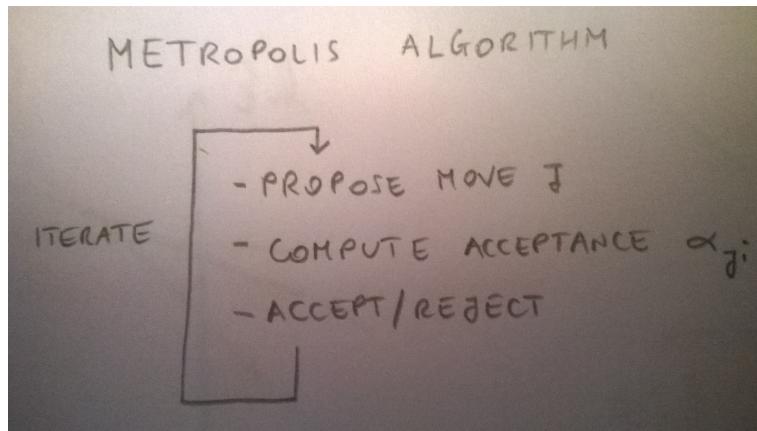
brium. In other words, when there isn't detailed balance, it means that something non conservative is happening.

Since:

$$\mathcal{P}(p, q) \propto \mathcal{P}(q)\mathcal{P}(p)$$

Monte Carlo algorithms are typically performed on position only, since q and p distribution are independent.

2.7.2 Metropolis algorithm and Pseudocode



$$\pi_{ji} = (\text{probability of proposing move } j) \cdot (\text{probability of accepting it}) = M_{ji} \cdot \alpha_{ji}$$

$$\frac{M_{ji}\alpha_{ji}}{M_{ij}\alpha_{ij}} = \frac{\bar{P}_j}{\bar{P}_i} = \text{detailed balance} \rightarrow \frac{\alpha_{ji}}{\alpha_{ij}} = \frac{\bar{P}_j M_{ij}}{\bar{P}_i M_{ji}} = e^{-\beta(H_j - H_i)} \frac{M_{ij}}{M_{ji}}$$

The choice of Metropolis is to take:

$$\alpha_{ji} = \min \left\{ 1, \frac{\bar{P}_j M_{ij}}{\bar{P}_i M_{ji}} \right\}$$

```

//initialize
i = ...

for (int istep = 0; istep < nsteps; istep++) {
    //for integer i, j
    if(rand() > 0.5) {
        j = i + 1
    } else {
        j = i - 1
    }

    /*for real i, j
    if(rand() > 0.5) {
        j = i + (rand() - 0.5)*delta
    }*/
}

```

```

/*not time reversible
if(rand() > 2/3)           {
    j = i + 1
} else {
    j = i - 1
}

in this case I have to compute a different alpha in each branch, since
Mij and Mji are different*/

alpha = exp(-beta*(h(j)-h(i)))

if(alpha > rand())          {
    i = j
}

/*this condition can be refined:
if(alpha > 1 || (alpha < 1 && alpha > rand() ))
in this case i generate only the random numbers that I need and the
code is optimized*/

print i
}

/*
Notice that other choices for the move could lead to non ergodicity, for
example:
Case of integers i,j:
- j = i + 2: parity of i is conserved. Every time the system is not ergodic,
there is a conserved quantity.
- case of real i, j: j = i + rand()
This can happen even if P is stationary.
*/

```

If X_i were independent, for $N \rightarrow \infty$ we would make an error $\sim \frac{\sigma}{\sqrt{N}}$, where σ is the standard deviation.

We don't have independent X_i , each point is close to the previous one, so we have a larger error:

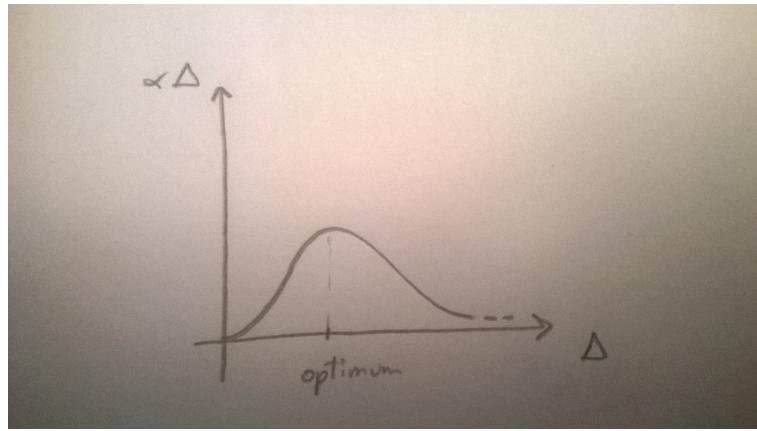
$$\frac{\sigma}{\sqrt{\frac{N}{N_{correlation}}}}$$

We can implement Markov Chain Monte Carlo using a different rule: **symmetric rule**. In this case probability of acceptance is given by:

$$\alpha = \frac{\mathcal{X}}{1 + \mathcal{X}} = \frac{\bar{P}_j}{\bar{P}_i + \bar{P}_j} \text{ where } \mathcal{X} = \frac{\bar{P}_j M_{ij}}{\bar{P}_i M_{ji}}$$

In this case we look at relative probability to make a choice. By the way Metropolis algorithm is considered the best choice, since it has the highest acceptance rate and so the least waste of calculations. If we'd try to increase $\alpha_{Metropolis}$, we'd end up with an $\alpha > 1$ for the reverse move, so this is the best we can get.

Actually one should choose α in order to optimize the product $\alpha \cdot \Delta$, where Δ is the typical size of moves and the plot goes like this:



A smart move should be to move all particles together, but if all particles are moved randomly, since $\Delta H \propto \sqrt{N}$, the acceptance drops. There are algorithms designed to create moves of group of particles in a smart way, they are more efficient, the tricky part is to compute $\frac{M_{x \leftarrow x_{TRY}}}{M_{x_{TRY} \leftarrow x}}$.

Grand Canonical Monte Carlo

Since injecting a particle raises the energy, Grand Canonical Monte Carlo are designed to have algorithms that inject particles where density is low, in order not to vary too much total energy.

```
//initialize
x[natoms]

for (int istep = 0; istep < nsteps; istep++) {
    - Choose particle i randomly
    - Move i
    - Compute energy (we should have previous value saved somewhere)
    - Accept/Reject
}
```

This algorithm is time reversible, but usually we don't choose particle i randomly:

```
//initialize
x[natoms]

for (int istep = 0; istep < nsteps; istep++) {
    for (int i = 0; i < natoms; i++) {
        - Move i
        - Compute energy (we should have previous value saved somewhere)
        - Accept/Reject
    }
}
```

The a step of the inner loop is usually called a *Monte Carlo Sweep*. In this case, the algorithm is not time reversible, because I know the order of the particles moving: if I see particle 10 moving before particle 9, I know the time is reversed. This breaks detailed balance, but it's not an issue, since balance is still satisfied.

2.7.3 Hybrid Monte Carlo

This method uses integrators to propose a new move at H very similar to the previous one. For example we can use Velocity Verlet from molecular dynamics in order to generate an attempt set of p, q and then accept or reject this set. If energy is almost conserved, acceptance is almost equal to one.

In some sense Monte Carlo corrects imperfections of molecular dynamics: if proposed move is decent enough, with one calculation we have moved all the particles.

Detailed balance is not satisfied, but it's satisfied a condition called **generalized detailed balance**:

$$P_B \pi_{BA} = \pi_{A^* B^*} P_{A^*}$$

Notice that this condition should be on M_{ij}, M_{ji} , but if E is conserved, the move will be approved and these quantities are the same (further discussion on this: *Stochastic Methods: A Handbook for the Natural and Social Sciences*, C. Gardiner).

Actually this condition is the same as:

$$J_{B \leftarrow A} = J_{A^* \leftarrow B^*} = 1$$

since one is the inverse of the other.

If we use an integrator that is time reversible and conserves energy and density, we have $\pi_{AB} = \pi_{A^* B^*}$, $P_B = P_{A^*}$. Generalized detailed balance is satisfied.

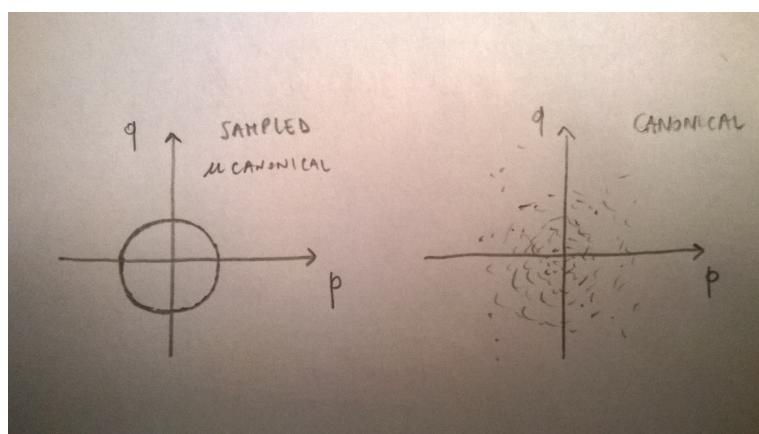
If we use an integrator as Velocity Verlet, we find approximate solutions: energy is almost (but not!) conserved. We won't be sure that generalized detailed balance will be satisfied:

$$\frac{M_{A^* B^*} P_{B^*}}{M_{B A} P_A} = e^{-\beta(\tilde{H}_B - \tilde{H}_A)} \neq 1$$

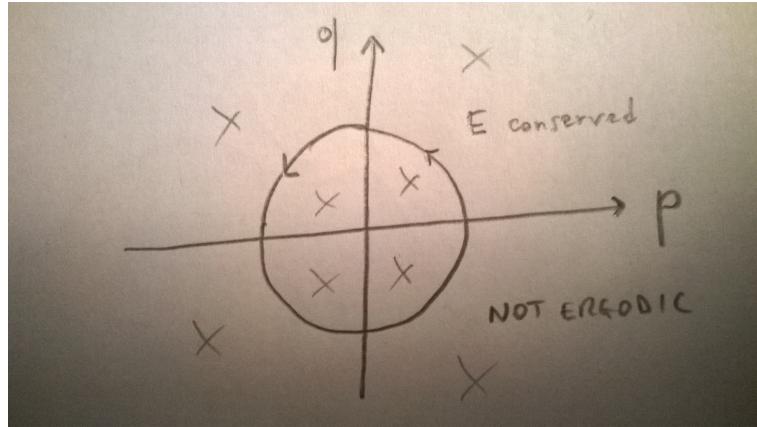
If energy is almost conserved, we can say that this quantity is close to one and ignore the *accept/reject step*. This leads to a more efficient but less accurate algorithm.

If we want to conserve energy, we can perform one step of Velocity Verlet, then change v in order to have energy equal to the previous one: we have constant energy by construction. Nobody uses this: it's not more accurate, constant energy doesn't mean I'm sampling micro-canonical ensemble.

An inconsistency arises: we have gone from sampling a canonical ensemble, to sampling a micro-canonical one.



The problem is this algorithm is **not ergodic** in the canonical ensemble. We said that if a system is not ergodic there is a conserved quantity, in this case is evident that's the energy.



We can tackle this problem by using a larger Δt , energy won't be exactly conserved. There is another problem: internal points are accepted, if a move is rejected, the system gets stuck in a point. Given p, q , the computation of H gives always the same result, so the algorithm will propose always the same move, that's already been rejected. To solve this problem, p are randomized at the beginning of each step.

```
//initialize
x[natoms]

for (int istep = 0; istep < nsteps; istep++) {
    randomize p //to not get stuck in a point. Canonical distribution is
    left stationary by this step
    (q', p') = velocity_verlet(q, p, deltat, natoms)
    /*this deltat is typically higher than one used in traditional Monte
     Carlo
    n is usually 10-20*/
    alpha = min{1, e^(-beta[H(q', p') - H(q, p)])}

    if (alpha > rand())      (q, p) = (q', p') //if you forget this and
        timestep too large, simulation explodes
}
```

However there's a big gain when we can exploit trick to compute energy on all the particles in a simple way. Otherwise the gain is reduced by the fact that recomputing energy for moves of fewer particles is much less expensive.

Another issue is how much energy should vary: it should vary of fraction of $k_B T$, since if $\Delta E \sim 0.01 k_B \rightarrow \alpha \simeq 1$. Fluctuations depend on N , this is the main reason because hybrid Monte Carlo is not very used: it doesn't scale with N . It's obvious: more atoms mean that a hit between two of them is more likely, so it's more likely the energy will vary a lot. Increasing N we have to use a shorter timestep.

Another issue to discuss is why we don't perform an accept/reject after randomizing p : if the move is correct, it should always be accepted, since we should have $M_{AB} = \bar{P}_A$, $M_{BA} = \bar{P}_B$, so $\alpha = \min \left\{ 1, \frac{P_B M_{AB}}{P_A M_{BA}} \right\} = 1$.

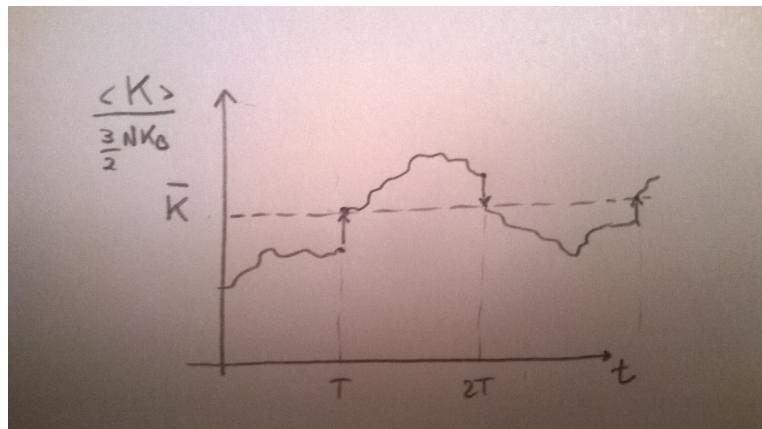
Notice that if we implemented an algorithm with a stochastic term in Verlet velocity and without randomizing p at every step, we would satisfy generalized detailed balance, so we wouldn't need to modify the code.

2.8 Thermostats

We usually relate simulations at fixed energy with experiments at fixed temperature. In μ canonical ensemble we can define T as:

$$T = \frac{\langle K \rangle}{\frac{3}{2}Nk_B}$$

So we can run simulation at some initial condition at constant energy and then find T . If we want higher T we increase initial energy.



2.8.1 Velocity rescaling, Berendsen Thermostat and Pseudocode

If we want to keep temperature almost constant, we can rescale velocity at some step multiplying p by $\sqrt{\frac{K_{TARGET}}{K}} \rightarrow p_{NEW} = p_{OLD} \sqrt{\frac{K_{TARGET}}{K}}$. New kinetic energy will be equal to the target one. The frequency of this rescaling defines the strength of the thermostat: doing it rarely gives a more accurate trajectory but less constant temperature, while doing it often keeps temperature almost constant, but trajectory is more affected.

In this algorithm not all the steps are equal, this is not pleasing because we can't calculate auto-correlation functions: there's something periodic happening. It would be nice to have something smoother, continuous.

We could compute a K_{TARGET} which is close to the constant, but not equal, using differential equations:

$$dK = \frac{\frac{3}{2}Nk_B T - K}{\tau} dt \rightarrow dK = (\bar{K} - K) \frac{dt}{\tau}$$

τ is an input parameter that gives the timescale of relaxation: it gives the intensity of the thermostat. If $\tau = 1$ and $dt = 10^{-3}$, in 10^3 steps kinetic energy will be relaxed.

$$K(t + \Delta t) = \bar{K} + (k - \bar{K})e^{-\frac{\Delta t}{\tau}} = K(t)e^{-\frac{\Delta t}{\tau}} + \bar{K} \left(1 - e^{-\frac{\Delta t}{\tau}}\right)$$

if $\tau = 0$, $K_{new} = \bar{K}$, new kinetic energy is the target one, while if $\tau \rightarrow \infty$, $K_{new} = K_{old}$ since there is no thermostat acting (it's like rescaling with infinite period, it never happens). We should notice that K is changing also because we're integrating the equations of motion, but we're just combining the effect of the equation (it's something similar to Trotter splitting). This is called **Berendsen Thermostat**.

```
//initialize
q[natoms]
p[natoms]

for (int istep = 0; istep < nsteps; istep++) {
    q, p = velocity_verlet(q, p, deltat, natoms)

    scaling_factor = sqrt( (K(p) e^(-deltat/tau) + K_bar (1-e^(-deltat/tau))
        ) / ( K(p) - K_actual ) ) //numerator is similar to a weighted
        average with two factors that add to one

    p = p*scaling_factor //all p are scaled
}
```

This works as a real thermostat: it measures T and correct it. This thermostat has a problem: it doesn't enforce anywhere canonical distribution: when $\tau \rightarrow 0$ this is not ergodic, K is going to be a fixed value. Fluctuations are never right for K with this algorithm. It's not possible to know in advance which distribution you will be sampling. It is used anyway because simulations equilibrate very quickly.

This is a **global thermostat**: you don't need to measure each particle, but K_{TOT} and correct it to a proper value. This doesn't work if a system is decoupled in two points. Typically systems get to thermic equilibrium quickly, if it's not a special case.

We can derive a differential equation corresponding to this thermostat:

$$\vec{p}_{NEW} = \vec{p}_{OLD} \sqrt{1 - \frac{\Delta t}{\tau} + \frac{\Delta t \bar{K}}{\tau}} = \vec{p}_{OLD} + \frac{\Delta t}{2\tau} \left(\frac{\bar{K}}{K} - 1 \right) \vec{p}_{OLD}$$

in the limit $\Delta t \rightarrow 0$. The change of p is proportional to p itself times a factor:

$$\dot{\vec{p}} = -\frac{1}{2\tau} \left(1 - \frac{\bar{K}}{K} \right) \vec{p} = -\gamma \vec{p}$$

γ is a friction coefficient, it's a variable (not an input parameter) and it's dependent on K . When K is large positive friction slows particles, while when K is small particles are accelerated. The set of non hamiltonian differential equations that describe Berendsen thermostat is:

$$\begin{cases} \dot{q} = \frac{p}{m} \\ \dot{p} = f - \gamma p \end{cases}$$

of course these aren't implemented in an algorithm, we implement a correction factor.

2.8.2 Andersen, Langevin Thermostat and Pseudocode

Instead of scaling all p with some factor, we could randomize them.

```

void randomize(p) {
    for( i = 0; i < natoms; i++) {
        for( j = 0; j < 3; j++) {
            p[i][j] = gasdev()*sqrt(m_i/beta) //gasdev returns a RN
            from gaussian distribution
        }
    }
}

//initialize
q[natoms]
p[natoms]

for (int istep = 0; istep < nsteps; istep++) {
    if(step%stride) p = randomize() //stride is the strength of the
                                     thermostat
    (q, p) = vv(q, p)
}

```

This thermostat is local: at every step, you affect each of the particles independently of the others. It's a **local thermostat**. This algorithm is disliked because there are steps different from others. We can play a trick similar to the one used before:

$$p_{NEW} = c_1 p_{OLD} + c_2 \sqrt{m_i k_B T} R$$

where $c_1^2 + c_2^2 = 1$, since p_{NEW} , p_{OLD} , R should be Gaussian and the linear combination of a gaussian is also gaussian.

$$c_1 = e^{-\frac{\Delta t}{\tau}}$$

$$c_2 = \sqrt{1 - e^{-2\frac{\Delta t}{\tau}}}$$

so

$$p_{NEW} = e^{-\frac{\Delta t}{\tau}} p_{OLD} + \sqrt{1 - e^{-2\frac{\Delta t}{\tau}}} \sqrt{m_i k_B T} R$$

if $\frac{\Delta t}{\tau} \ll 1$ we can linearize:

$$p_{NEW} = (1 - \frac{\Delta t}{\tau}) p_{OLD} + \sqrt{2 \frac{\Delta t}{\tau}} \sqrt{m_i k_B T} R$$

we can write:

$$\Delta p = -\frac{\Delta t}{\tau} p_{OLD} + \sqrt{\frac{2 \Delta t m_i k_B T}{\tau}} R$$

if we define $\gamma_p = \frac{1}{\tau}$:

$$\Delta p = -\gamma_p p \Delta t - \sqrt{2 m_i k_B T \gamma} \sqrt{\Delta t} R$$

We call this **integrator for Langevin equation**. The increment of the momentum is $\Delta p = F\Delta t$, so there is a force acting on particles:

$$F = -\gamma_p p \propto -p$$

It's a frictional force. The other term:

$$-\sqrt{2m_i k_B T \gamma} \sqrt{\Delta t} R$$

is a random force that describes thermic effects and it's called **Wiener noise**. in the limit of $\Delta t \rightarrow 0$ it goes to infinity.

$$\Delta p = -\gamma_p p \Delta t - \sqrt{2m_i k_B T \gamma} \sqrt{\Delta t} R$$

This is a stochastic differential equation and it's the **Langevin equation**. Using **Itô formalism** (which we will use from now on):

$$dp = -\gamma p \Delta t + \sqrt{2m k_B T \gamma} dW \text{ where } dW = \sqrt{\Delta t} R$$

using **Stratonovich formalism** we would write:

$$\dot{p} = -\gamma p + \sqrt{2m k_B T \gamma} \eta \text{ where } \eta = \frac{dW}{dt}$$

There are differences between the two ways of describing stochastic differential equations, so we have to be careful and check in which formalism we are working. The process of translating one formalism in the other is very error prone.

Coming back at the previous equation, we can write more in general:

$$\Delta p = A \Delta t + B \sqrt{\Delta t} R$$

It seems that first term is negligible with respect to the second in the limit of $\Delta t \rightarrow 0$, but it's not as it seems. If I double Δt , I get:

$$\Delta p = A(\Delta t + \Delta t) + \dots = 2A\Delta t + \dots$$

for the first part, so the first term grows linearly with Δt as I expected, for the second one, one would say:

$$\sqrt{\Delta t} + \sqrt{\Delta t} = \sqrt{2\Delta t} \neq \sqrt{2\Delta t}$$

but THIS IS A WRONG REASONING! The correct way is:

$$\sqrt{\Delta t} R_1 + \sqrt{\Delta t} R_2 = \sqrt{2\Delta t} R_3$$

since the sum of two gaussian random numbers is a gaussian random numbers multiplied by $\sqrt{c_1^2 + c_2^2}$, where c_i are the prefactors. **Both terms grow linearly with Δt** . Intuitively the bigness of the \sqrt is limited by the fact that increments sums up differently because multiplied by gaussian random numbers.

It can be demonstrated that the integrator is the exact solution (in probability) of Itô equation. This is just another step in the algorithm, which implements the thermostat, in fact, using Trotter splitting, we have the same code as before with two more lines.

```

    for (int istep = 0; istep < nsteps; istep++) {
        p = c1*p + c2*sqrt(m*kB*T)*gasdev() //loop over p_i - step 1
        (q, p) = vv(q, p) // step 2
        p = c1*p + c2*sqrt(m*kB*T)*gasdev() //loop over p_i - step 3
    }

    //c1 = exp(-gamma*deltat/2)
    //c2 = sqrt(1-exp(-gamma*deltat/2)*c1^2)

```

Balance is satisfied by this algorithm: if p is distributed as a gaussian of zero mean and unitary variance and $\text{gasdev}()$ too, their sum is again a gaussian of zero mean and unitary variance. It can be verified that:

$$\frac{\pi_{AB} P_B}{\pi_{BA} P_A} = 1$$

using $P(x'|x) = \frac{1}{\sqrt{2\pi c_2}} e^{-\frac{(x' - c_1 x)^2}{2c_2}}$ and $c_1^2 + c_2^2 = 1$.

We notice that energy changes in step 1 and 3 are obvious (since it's what we want), but we don't care since they satisfy detailed balance, while step 2 doesn't conserve energy perfectly and it could be a problem. What is **almost conserved** in this algorithm is **total energy - increments from thermostat**.

This algorithm can be used to implement Hybrid Monte Carlo. Another important thing is that in infinitesimal Itô, R can be distributed in any way as long as it has the correct variance (it can also be infinite, to implement jump processes!), while in the integrator R has to be gaussian.

Langevin thermostat is a local thermostat: each particle's momentum is changed and there is no calculation of total kinetic energy.

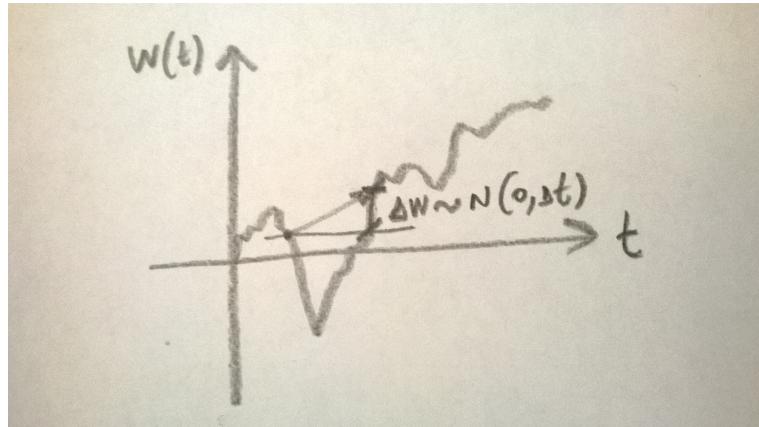
2.9 Stochastic differential equations

$$dx = A(x, t)dt + B(x, t)dW$$

$$\Delta x = A(x, t)\Delta t + B(x, t)\Delta W$$

with $\Delta W = \sqrt{\Delta t}R$, where R is a random number drawn from any distribution with zero mean and unitary variance. This ΔW can be seen as the increment of a random walk:

$$\Delta W = W(t + \Delta t) - W(t)$$



Choosing which of the RW we want to sample is the process of choosing the seed for the random number generator: it converts the process from stochastic to deterministic. The $\sqrt{\Delta t}$ has dramatic consequences when the increment is of $f(x)$ instead of x . Let's say $y = y(x)$:

$$\Delta y \neq \frac{\partial y}{\partial x} \Delta x$$

in fact:

$$\begin{aligned}\Delta y &= y' [A\Delta t + B\Delta W + 2AB\Delta t\Delta W] + \frac{y''}{2} [A^2\Delta t^2 + B^2\Delta W^2 + 2AB\Delta t\Delta W] + \dots = \\ &= y' A\Delta t + y' B\Delta W + \frac{y''}{2} B^2 \Delta W^2\end{aligned}$$

Obviously we can neglect $A^2\Delta t^2$, we can neglect $2y'AB\Delta t\Delta W$ and $2AB\Delta t\Delta W$ too, since $\Delta W \cdot \Delta t$ is negligible with respect to Δt . On the reason why to keep the second term we've already discussed. We now discuss why we need to keep into account the third term.

$$\Delta W^2 = \Delta t R^2$$

where R is a random number drawn from a distribution with zero mean and unitary variance. Notice that $\sigma^2 = \langle R^2 \rangle - \langle R \rangle^2 = \langle R^2 \rangle - 1$, since mean of R is zero, we can just rewrite:

$$\langle R^2 \rangle = \Delta t + \Delta t(R^2 - 1)$$

We analyze $(R^2 - 1)$ and see that it has zero mean (since we found $\langle R^2 \rangle = 1$) and finite variance, $\Delta t(R^2 - 1)$ is much smaller than Δt and $\Delta W \propto \Delta t$ and it can not be neglected. At the end of this calculations we find:

$$\begin{aligned}\Delta y &= y' (A\Delta t + B\Delta W) + \frac{y''}{2} B^2 \Delta t \\ dy &= y' (Adt + BdW) + \frac{y''}{2} B^2 dt\end{aligned}$$

from which:

$$dy = \left(y' A + \frac{y''}{2} B^2 \right) dt + y' BdW$$

which is **Itô Rule for chain derivatives**. We didn't expect $\frac{y''}{2} B^2 dt$, so we have to be careful when taking derivatives of stochastic equations. Notice that if y is purely stochastic ($A = 0$), we have:

$$dy = \frac{y''}{2} B^2 dt + y' BdW$$

y could be on average increasing or decreasing, depending on its convexity (for application of this to finance: "Stochastic Differential Equations and Financial Mathematics - Thomas Onskog").

If we have to implement an algorithm to solve a differential equation

$$x(t + \Delta t) = A(x, t)\Delta t + x(t)$$

we could ask if we should put $A(x + \Delta x, t + \Delta t)$ or $A(x(t), t)$. Actually it doesn't matter which choice we do, because the solution is the same for $\Delta t \rightarrow 0$, **this isn't true anymore if we have dW** .

Implementations of Itô and Stratonovich formalisms are different: Itô formalism modifies the chain rule, while Stratonovich doesn't, but in Stratonovich formalism we can have only implicit algorithms to solve stochastic differential equations, while in Itô's we can implement a simple Euler method.

Capitolo 3

Assignments

3.1 Assignment 1

Files for this assignment can be downloaded at: <https://github.com/GiovanniBussi/simplemd>

Preliminary comments First, we take a look at the input file:

```
|| inputfile crystal.xyz
|| outputfile output.xyz
|| temperature 0.1
|| tstep 0.05
|| friction 0.0
|| forcecutoff 2.5
|| listcutoff 3.0
|| nstep 2000
|| nconfig 10 trajectory.xyz
|| nstat 10 energies.dat
```

File `crystal.xyz` is an input file which contains the crystal structure of the system. Temperature is in $k_B T$ units. Timestep is chosen initializing `tstep` variable. Number of steps is chosen with `nstep`. `Forcecutoff` tells how far interactions are different from zero. List cutoff is a parameter for neighbors list computation optimization.

This standard input file makes sense, but 2000 is a very short simulation, we may want to increase this number.

Nerd stuff:

```
|| ln -s ..../cpp/simplemd.x
creates a shortcut to execute simplemd.x with
|| ./simplemd.x
```

Notice that our system has $3N - 3$ degrees of freedom, since velocity of center of mass is conserved, so:

$$T \neq \frac{\langle K \rangle}{\frac{3N_A}{2} K_B} \rightarrow T = \frac{\langle K \rangle}{\frac{3N_A-3}{2} K_B}$$

This is the temperature for a microcanonical ensemble. We can see how temperature varies with time with gnuplot writing in the terminal:

```

|| $gnuplot
>plot 'energies.dat' u 2:3
to update plot: push e

```

We see that temperature equilibrates around a value that is more or less 50% of the starting one. This is because we are initializing atoms in a configuration of minimal potential energy, it can't become lower, so when part of kinetic energy becomes potential energy, temperature decreases. Since around a half of kinetic energy becomes potential, the final value of temperature is about half the initial one.

We can automate simulations at different temperatures writing **bash scripts**:

```

#!/bin/bash

for T in 0.01 0.1 1 10
do

cat > input-$T << EOF
inputfile crystal.xyz
outputfile output.xyz
temperature $T
tstep 0.05
friction 0.0
forcecutoff 2.5
listcutoff 3.0
nstep 20000
nconfig 10 trajectory.xyz
nstat 10 energies-$T.dat
EOF

done

```

Then we get the permission to execute the script in terminal with

```

|| $chmod a+x script-filename

```

and we can execute it from terminal or by double-click.

Energy conservation - drift and fluctuations The proper way to evaluate drift would be to fit data with a straight line and take its angular coefficient. We can do it in a simpler and rough way by computing the difference between final and initial value and divide by the total step.

As before, we can write a bash script to automate different simulations at different time-step values, we define nstep in function of time-step in order to have the same number of samples:

```

#!/bin/bash

for tstep in 0.005 0.01 0.05 0.1 0.5
do

cat > input << EOF
inputfile crystal.xyz
outputfile output.xyz
temperature 0.722
tstep $tstep
friction 0.0
forcecutoff 2.5
listcutoff 3.0
nstep $((100000*tstep))
nconfig 10 trajectory.xyz
EOF

```

```

|| nstat    10 energies-$tstep.dat
EOF
done

```

We find that energy conservation is not as good using a higher time-step, but simulation ends up much faster. Slope of drift increases with increasing time-step. Traditional timesteps used for this system are: 0.01 and 0.005.

In some simulations we see something weird: the drift is going downwards. If the system is non-harmonic we should see system go up, but it's a matter of probability, in our case 4 on 5 simulations go up and it's reasonable.

As before, the proper way to compute fluctuations would be to fit with a straight line and then take the residuals. It can be done roughly by computing variance with another bash script and help of **awk**. A script to compute both drift and fluctuations:

```

#!/bin/bash

awk '{
    if(NR==2) {
        initial_time = $2 #because first value is our initialization and can
                           lead to spurious results.
        initial_value = $5
    }
}
sum+=$2
sum2+=$5*$5
END{
    sum2/=NR #NR is a predefined variable = numbers of rows
    sum/=NR
    print sum2-sum #variance
    print ($5-initial_value)/($2-initial_time)
},
energies-$tstep.dat
done

```

Nerd stuff:

```
|| plot [] [:0] '...'
```

doesn't show data with $y > 0$.

If we plot our data, we find that fluctuations and drift increase with time-step. With our value of drift we can estimate how long has to be the simulation to explode (sooner or later it will arrive to zero if angular coefficient > 0).

This is one of the reason why simulations at constant temperature are preferred to simulations at constant energy. Of course there are tricks to avoid this, for example, introducing a different error, you could enforce energy conservation by rescaling potential energy, in this way you lose time-reversibility, but recover stability of the algorithm.

Simulations at different temperatures - maximum allowed time-step Again we can automate the task of running simulations at different temperatures with a bash script.

What we would expect is that increasing temperature doesn't affect the maximum allowed time-step, since it depends only on the shortest period, which isn't function of T. What we see in our simulations instead is that maximum allowed time-step decrease with increasing temperature: this is because what we implement is not an harmonic oscillator, actually, the rule of fastest period gives an upper bound for the timestep.

Drift and fluctuations in relation to the size of the system We can write a bash script like:

```
#!/bin/bash

for n in 2 3 4 5
do

./lattice n > crystal-$((n*n*n*4)).xyz

cat > input-$T-$tstep << EOF
inputfile crystal.xyz
outputfile output.xyz
temperature 0.722
tstep 0.005
friction 0.0
forcecutoff 2.5
listcutoff 3.0
nstep 20000
nconfig 10 trajectory.xyz
nstat 10 energies-$((n*n*n*4)).dat
EOF

done
```

to generate several crystal structures and run simulations.

To check if we haven't done any mistake, we can plot temperature in function of time and see that it doesn't vary much among simulations, this is correct because it's an intensive quantity. The same is for energy per particle: it is similar among simulations, while if we plot total energies, they're different because depending on number of atoms.

Results of simulations show that drift and fluctuations increase with size of the system.

Drift increases with the size of the system. Fluctuations increase with the size of the system.

Bonus level Download vmd and learn how to use it. You can see your crystal structure and its dynamics.

An interesting fact is that you can see periodic boundary conditions: a particle going outside the cube reappears at the other side of it.

Simulations at constant energy are not very physical actually, we have a lot of approximation such as finite size effects and so on.