



# L10 – Statistical Mechanics 2

Monte Carlo algorithms, sampling techniques, and rates

# Monte Carlo Methods

**Monte Carlo methods** are computational algorithms that rely on repeated random sampling to obtain numerical results.

The underlying concept is to use randomness to solve problems, even deterministic ones.

Monte Carlo methods are mainly used in three types of problems:

- integration
- optimization
- draw from a probability distribution

**Monte Carlo** methods follow the pattern:

1. Define a domain of possible inputs
2. Generate inputs randomly from a probability distribution over the domain
3. Perform a deterministic computation on the input
4. Aggregate the results

# Monte Carlo integration

Consider the function

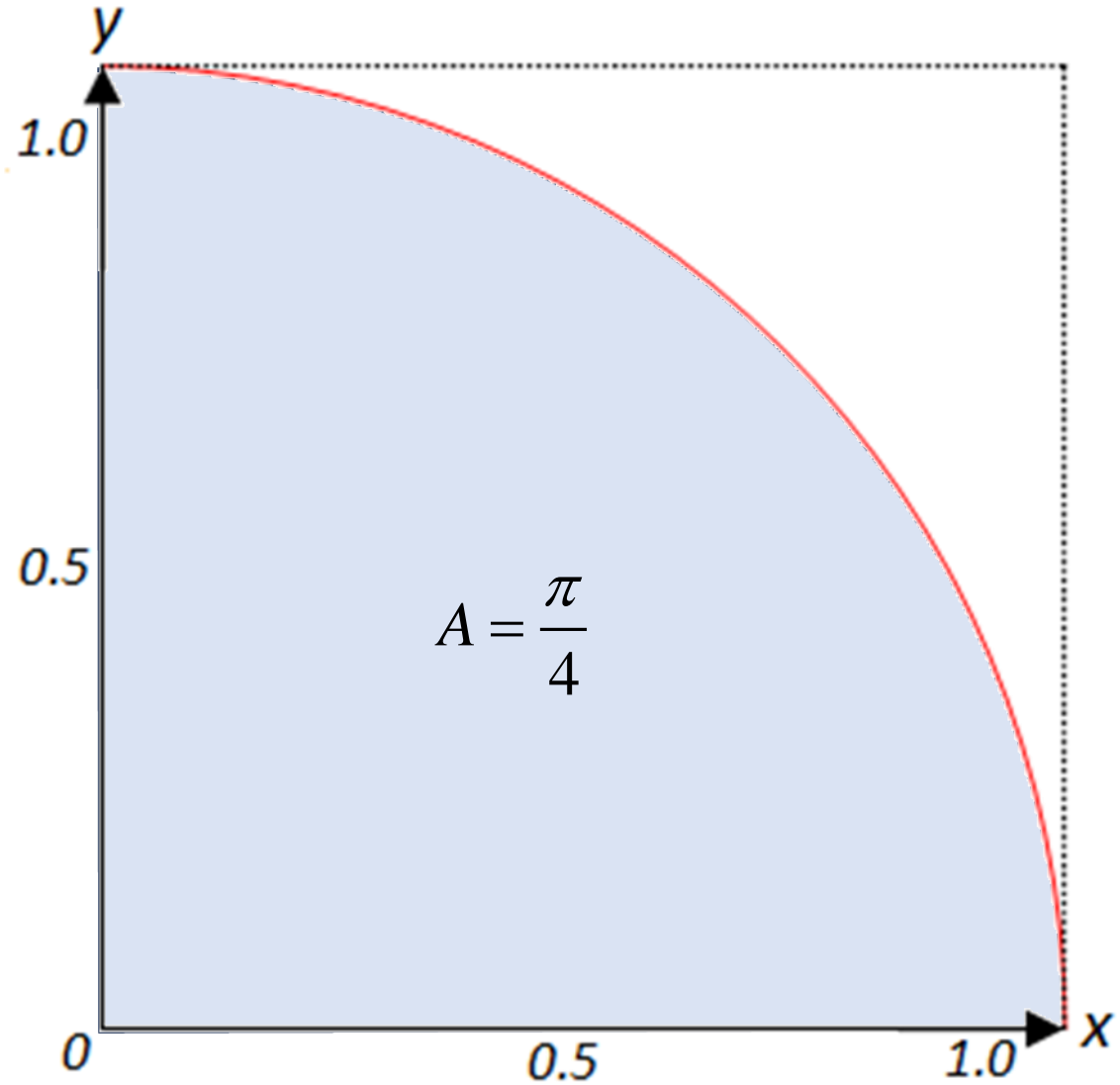
$$y = \sqrt{1-x^2}$$

Calculate

$$A = \int_0^1 \sqrt{1-x^2} dx$$

The result is

$$\begin{aligned} A &= \frac{1}{2} \left( x\sqrt{1-x^2} + \arcsin(x) \right) \Big|_0^1 \\ &= \frac{\pi}{4} \end{aligned}$$



Consider the function

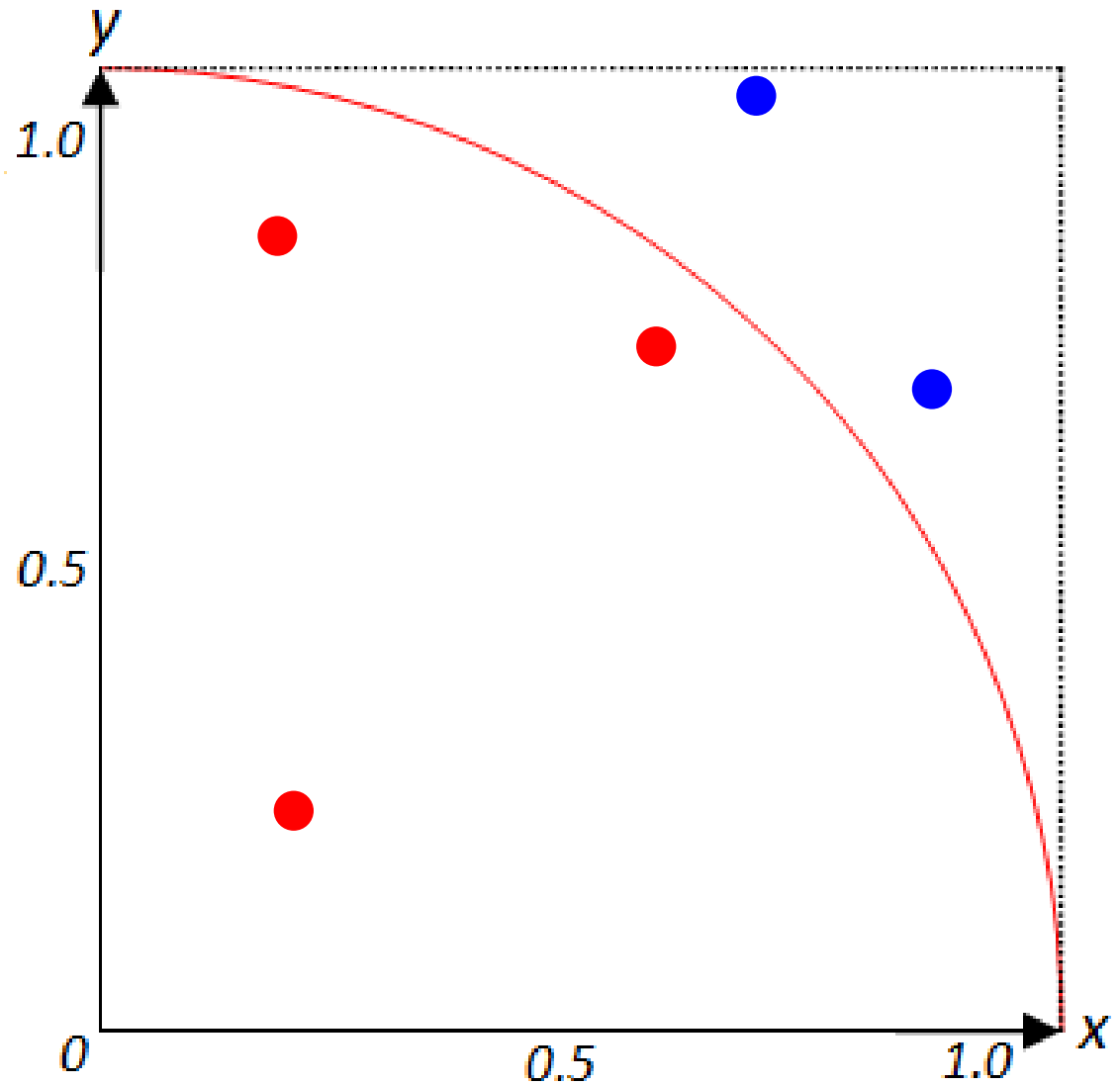
$$y = \sqrt{1-x^2}$$

Calculate

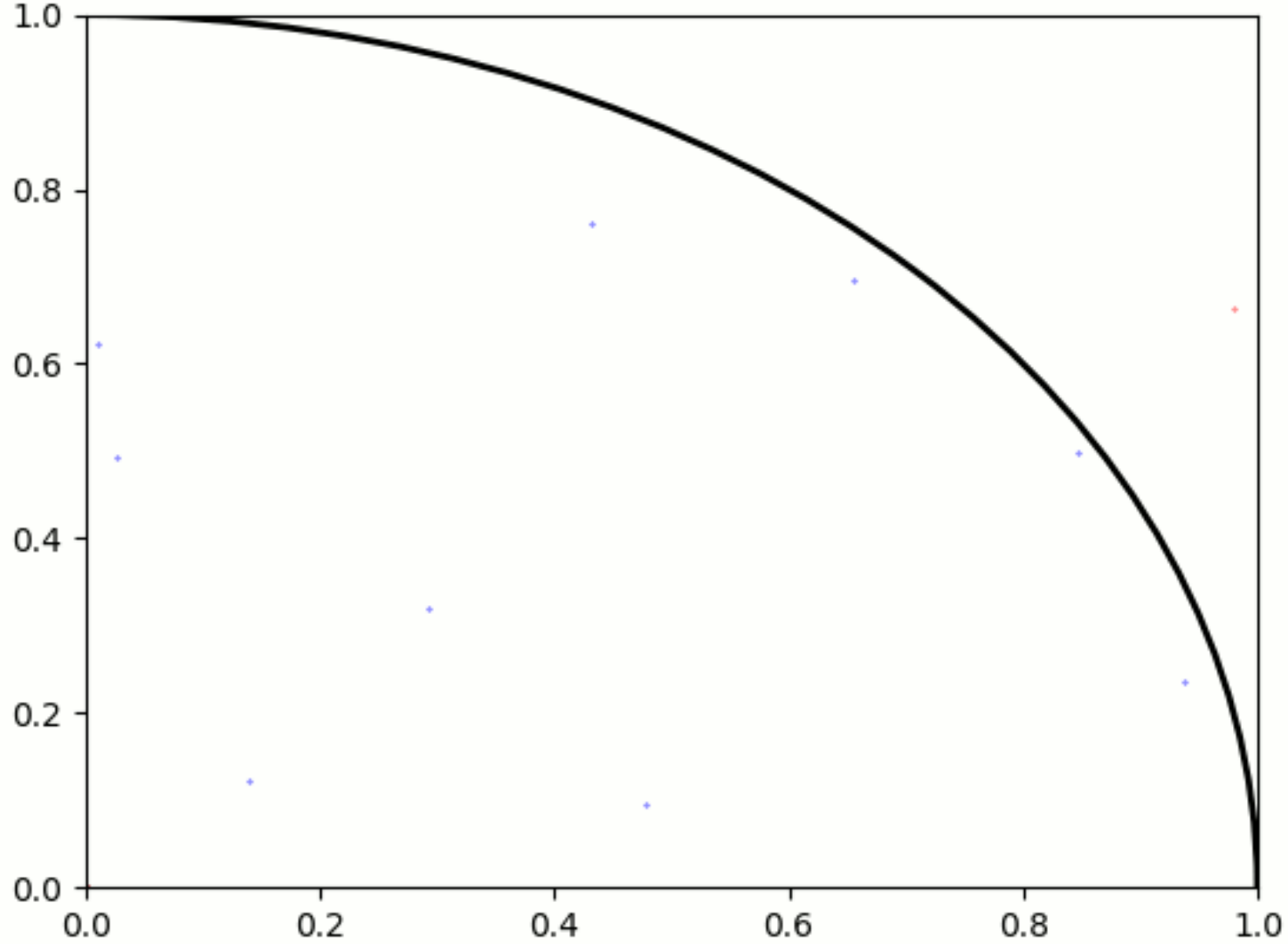
$$A = \int_0^1 \sqrt{1-x^2} dx$$

$$\frac{A}{A_{total}} \approx \frac{N_{red}}{N_{total}}$$

$$A \approx \frac{N_{red}}{N_{total}} A_{total}$$



Approx. Value of Pi: 3.6 | Percent Error : 14.6%





Such a procedure is known as **Monte Carlo integration**.

```
def myfunction(x):                                # Function to integrate
    return math.sqrt(1 - x**2)

(xmin, xmax) = (0.0, 1.0)                        # Define box
(ymin, ymax) = (0.0, 1.0)

Ntot = 10000                                     # Number of points

Atot = (xmax - xmin) * (ymax - ymin)            # Total area

Naccept = 0

for i in range(1, Ntot + 1):                     # Loop over all random points
    rx = random.uniform(xmin, xmax)
    ry = random.uniform(ymin, ymax)
    myfunction_result = myfunction(rx)

    if ry < myfunction_result:                  # Accept or reject
        Naccept += 1

Integral = Atot * Naccept / Ntot                # Estimate the integral
print("Integral = ", Integral)
```

Monte Carlo are stochastic methods following the pattern:

1. Define a domain of possible inputs

[Draw  $y(x)$  from  $x = 0$  to  $1$ ]

2. Generate inputs randomly from a probability distribution over the domain

[Sample random points within  $(x=[0..1], y=[0..1])$  domain]

3. Perform a deterministic computation on the input

[Check whether each point is smaller or larger than  $y(x)$ .]

4. Aggregate the results

[Compute the value  $A_{total} \cdot N_{accept} / N_{total}$ ]

Monte Carlo integration is advantageous:

- Complex or high-dimensional integrals
- Irregular or non-analytic functions
- Random or stochastic processes
- Parallel or distributed computing

The chief use of Monte Carlo integration in physical modelling of molecular systems is to evaluate the mean value of an observable  $A$ :

$$\langle A \rangle = \frac{\int d\mathbf{r}^N A(\mathbf{r}^N) \exp\left(-\frac{E(\mathbf{r}^N)}{k_B T}\right)}{\int d\mathbf{r}^N \exp\left(-\frac{E(\mathbf{r}^N)}{k_B T}\right)}$$

where

$$\mathbf{r}^N = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)$$

But are not limited to Boltzmann distributions. Any distribution  $P(\mathbf{r}^N)$  can be used:

$$\langle A \rangle = \frac{\int d\mathbf{r}^N A(\mathbf{r}^N) P(\mathbf{r}^N)}{\int d\mathbf{r}^N P(\mathbf{r}^N)}$$

where

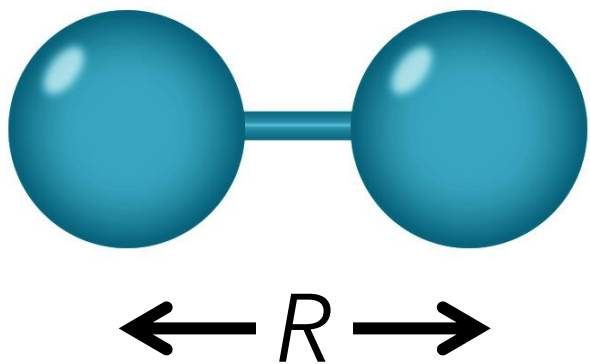
$$\mathbf{r}^N = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)$$

Quantum Monte Carlo is one of the most accurate quantum chemical electronic structure methods available. It is based on Monte Carlo integration of

$$\langle \Psi_{trial} | \hat{H}_e | \Psi_{trial} \rangle = \frac{\int d\mathbf{r}^N E_{trial}(\mathbf{r}^N) |\Psi_{trial}(\mathbf{r}^N)|^2}{\int d\mathbf{r}^N |\Psi_{trial}(\mathbf{r}^N)|^2}$$

# Monte Carlo sampling



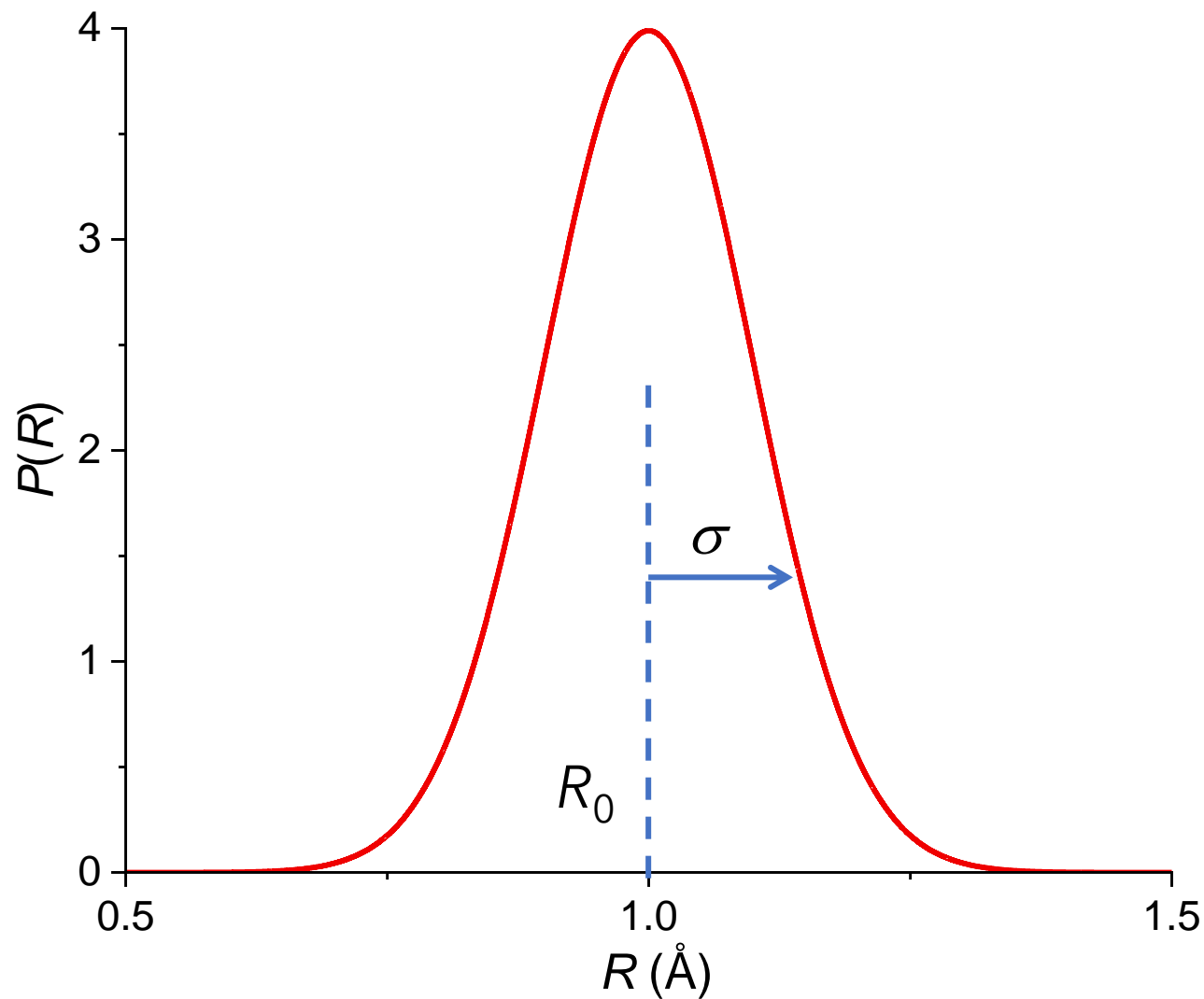


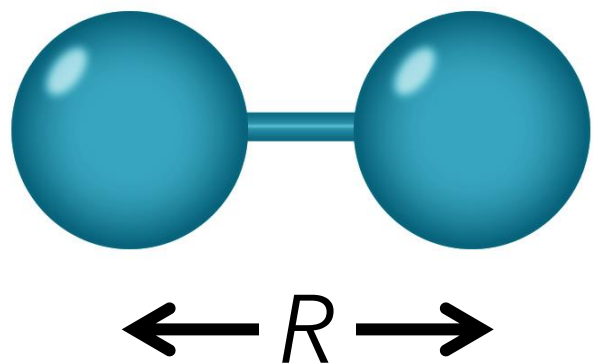
Suppose geometry is distributed as

$$P(R) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{(R - R_0)^2}{\sigma^2}\right)$$

$$R_0 = 1.0 \text{ \AA}$$

$$\sigma = 0.1 \text{ \AA}$$



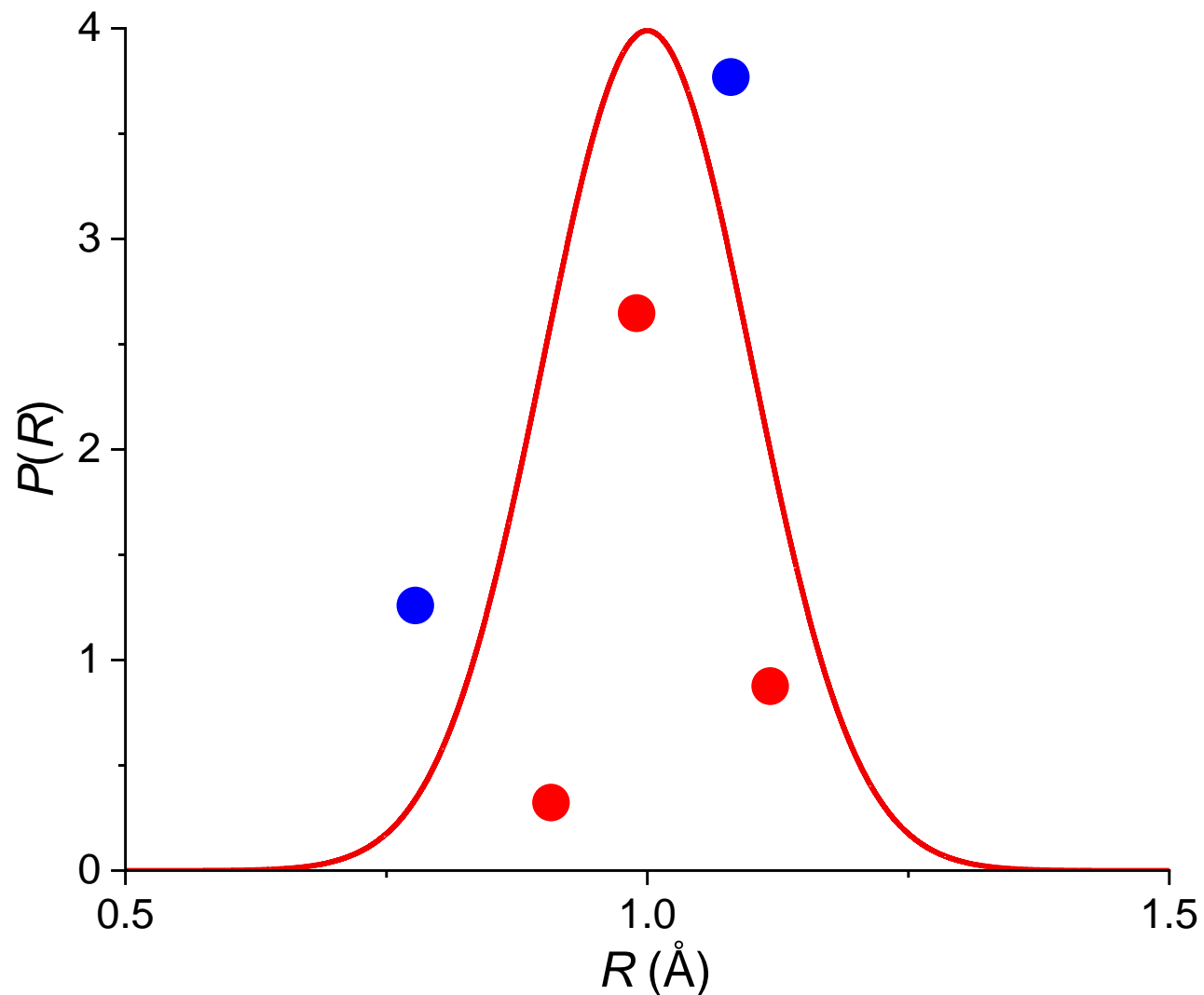


Sample random number in  $[x_{min}, x_{max}]$

Sample random number in  $[y_{min}, y_{max}]$

Compute  $P(\text{random } x)$

- accept if  $\text{random } y < P(\text{random } x)$
- reject if  $\text{random } y > P(\text{random } x)$



```
def gaussian(x, R0, sigma):                                     # Define probability distribution function
    return math.exp(-(x-R0)**2 / (2*sigma**2)) / (sigma*math.sqrt(2*math.pi))

R0 =1.0
sigma = 0.1
Nmax = 10000                                                # Number of points

(xmin, xmax) = (R0 - 3 * sigma, R0 + 3 * sigma)             # Define box
(ymin, ymax) = (0, gaussian(R0, R0, sigma))

for i in range(1, Nmax + 1):                                  # Loop over all random points
    random_x = random.uniform(xmin, xmax)
    random_y = random.uniform(ymin, ymax)
    Probability = gaussian(random_x, R0, sigma)               # Compute probability

    if random_y < Probability:                                 # Accept or reject
        print("Geometry is ", random_x)
```

# $N_{at}$ molecule: Wigner distribution

$$P_W(\mathbf{Q}, \mathbf{P}) = \prod_{i=1}^{N_F} \frac{1}{\sigma_{qi} \sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{Q_i^2}{\sigma_{qi}^2}\right) \times \frac{1}{\sigma_{Pi} \sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{P_i^2}{\sigma_{Pi}^2}\right)$$

$$\sigma_{qi} = \sqrt{\frac{\hbar}{2\alpha_i \mu_i \omega_i}} \quad \sigma_{Pi} = \sqrt{\frac{\hbar \omega_i \mu_i}{2\alpha_i}} \quad \alpha_i = \tanh\left(\frac{\hbar \omega_i}{2k_B T}\right)$$

$Q_i$  - normal mode

$P_i$  - momentum of the normal mode

$\omega_i$  - angular harmonic frequency

$\mu_i$  - reduced mass

$N_F = 3N_{at} - 6$  - number of normal modes

## $N_{at}$ molecule: Wigner distribution

$$P_W(\mathbf{Q}, \mathbf{P}) = \prod_{i=1}^{N_F} \frac{1}{\sigma_{qi} \sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{Q_i^2}{\sigma_{Qi}^2}\right) \times \frac{1}{\sigma_{Pi} \sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{P_i^2}{\sigma_{Pi}^2}\right)$$

1. For each normal mode  $i$ , make a Gaussian sampling of  $Q_i$  and  $P_i$ .
2. Take the set  $(Q_1, Q_2, \dots, Q_{N_F})$  and transform into Cartesian coordinate  $\mathbf{R}$ .
3. Take the set  $(P_1, P_2, \dots, P_{N_F})$  and transform into Cartesian momentum  $\mathbf{P}$ .
4. Repeat this procedure for as many points as you need.

You can use Newton-X for this procedure.

Harmonic Wigner sampling is a standard method to populate the phase space, including zero-point vibrational energy.

It can be used to generate initial conditions for QM molecular dynamics.

It only works well for rigid systems where harmonic approximation holds.

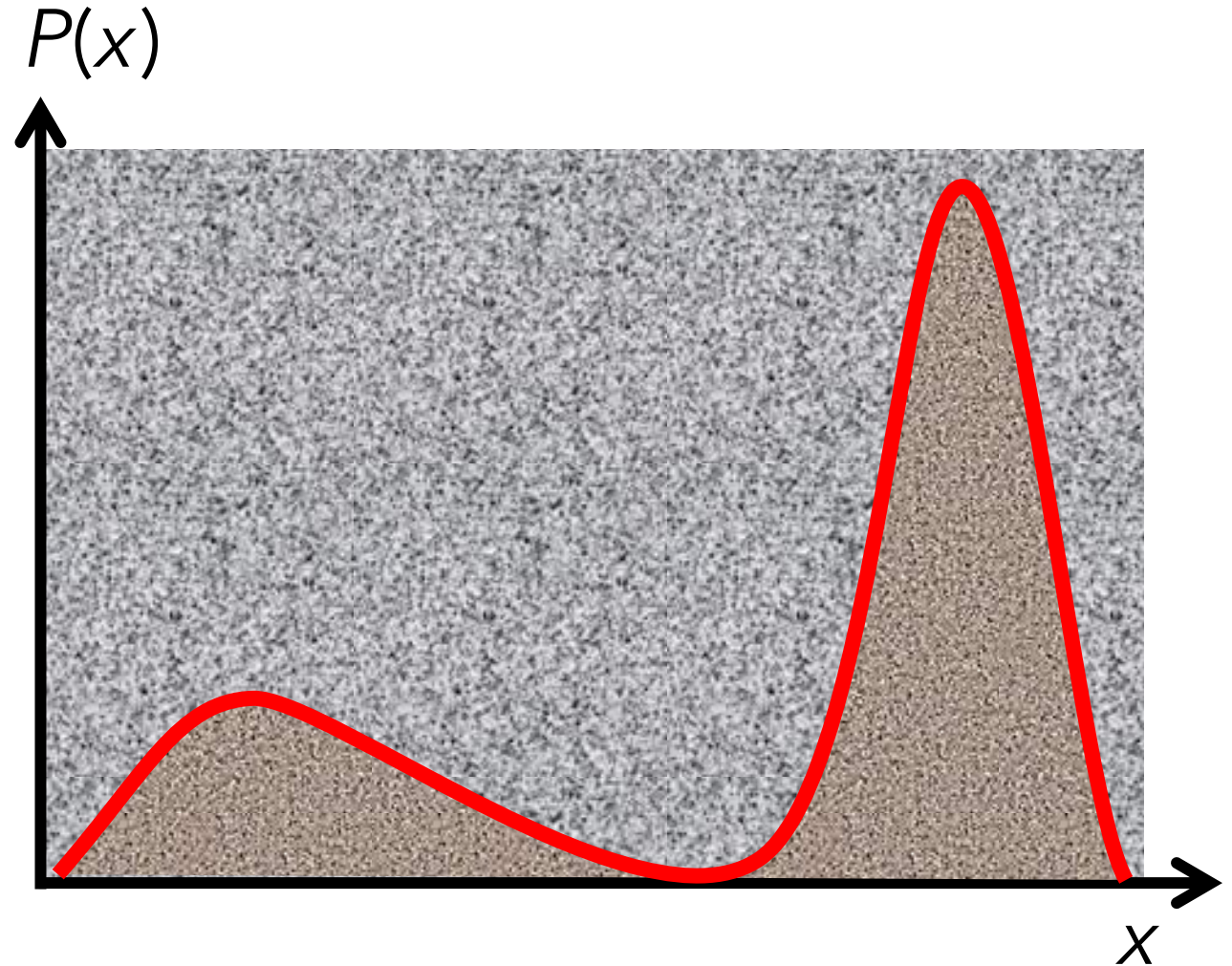
# Metropolis-Hastings algorithm

Suppose we want to sample  $P(x)$ .

We may waste too much time sampling above the curve (rejected points).

Metropolis and Metropolis-Hastings algorithms improve acceptance rate.

They also don't require defining the box.



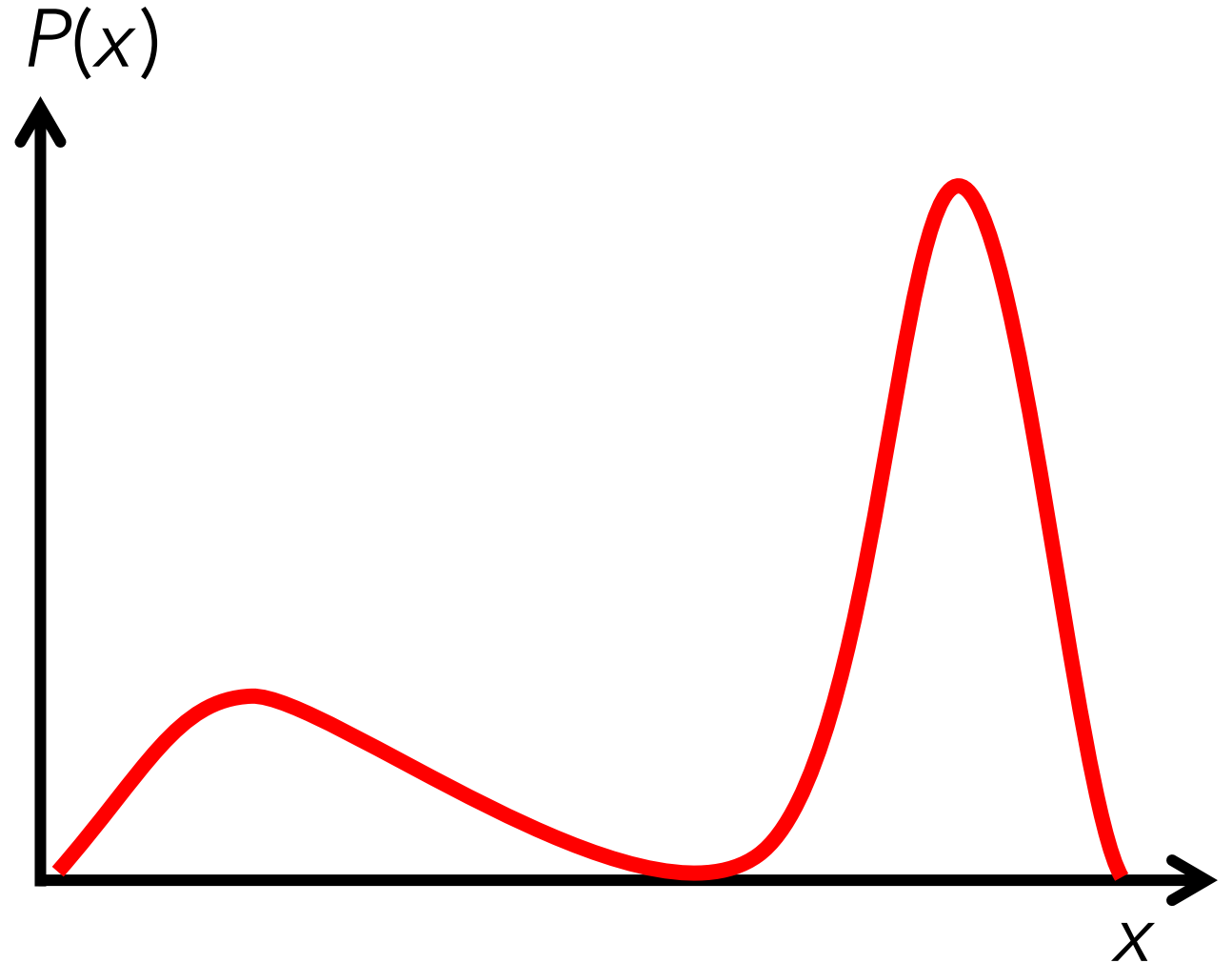
- [en.wikipedia.org/wiki/Metropolis-Hastings\\_algorithm](https://en.wikipedia.org/wiki/Metropolis-Hastings_algorithm)
- Metropolis *et al.* *J Chem Phys* **1953**, 21, 1087



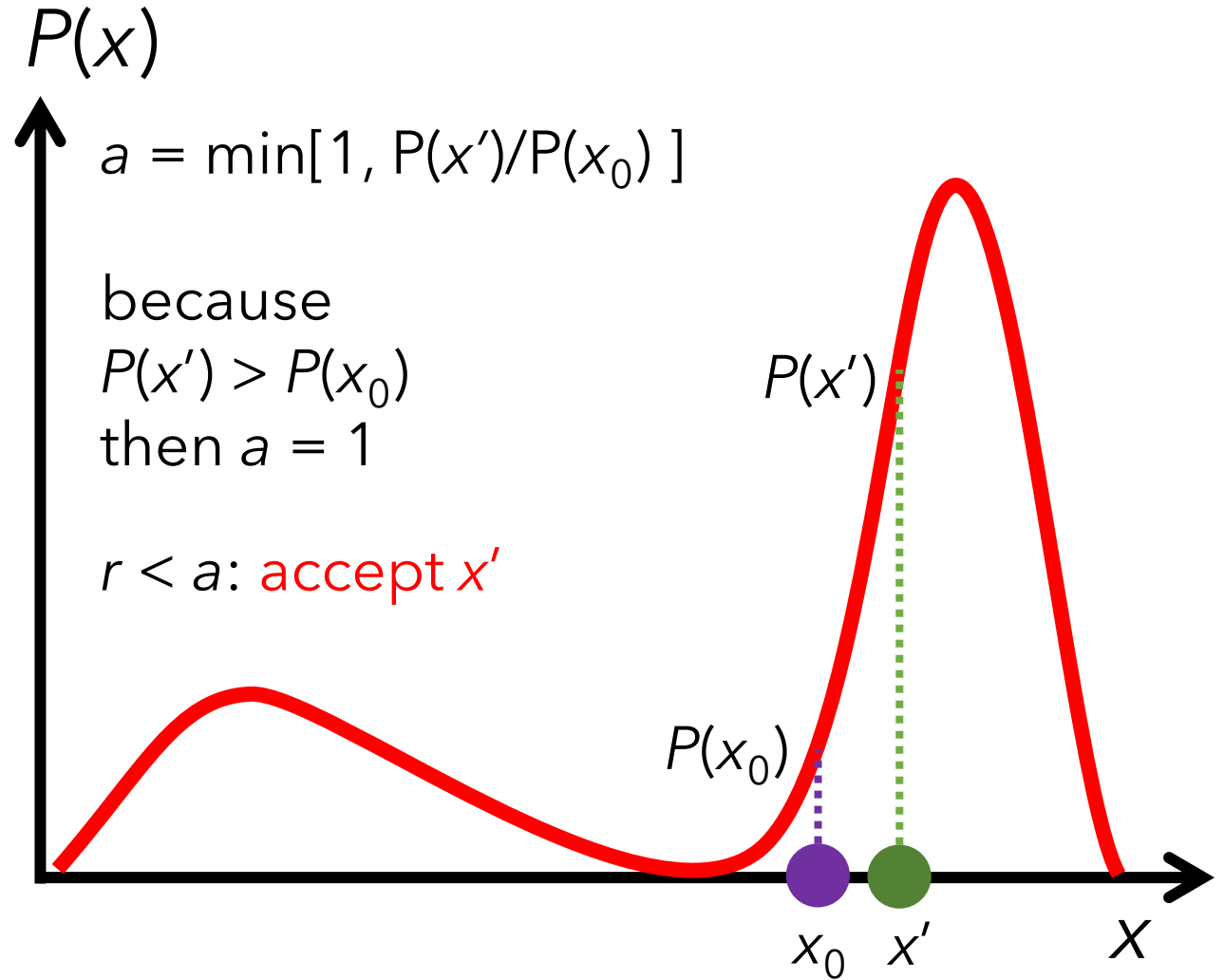
Before starting, choose a **proposal distribution**  $g(x|x_n)$ .

For example, a Gaussian distribution centred at  $x_n$ :

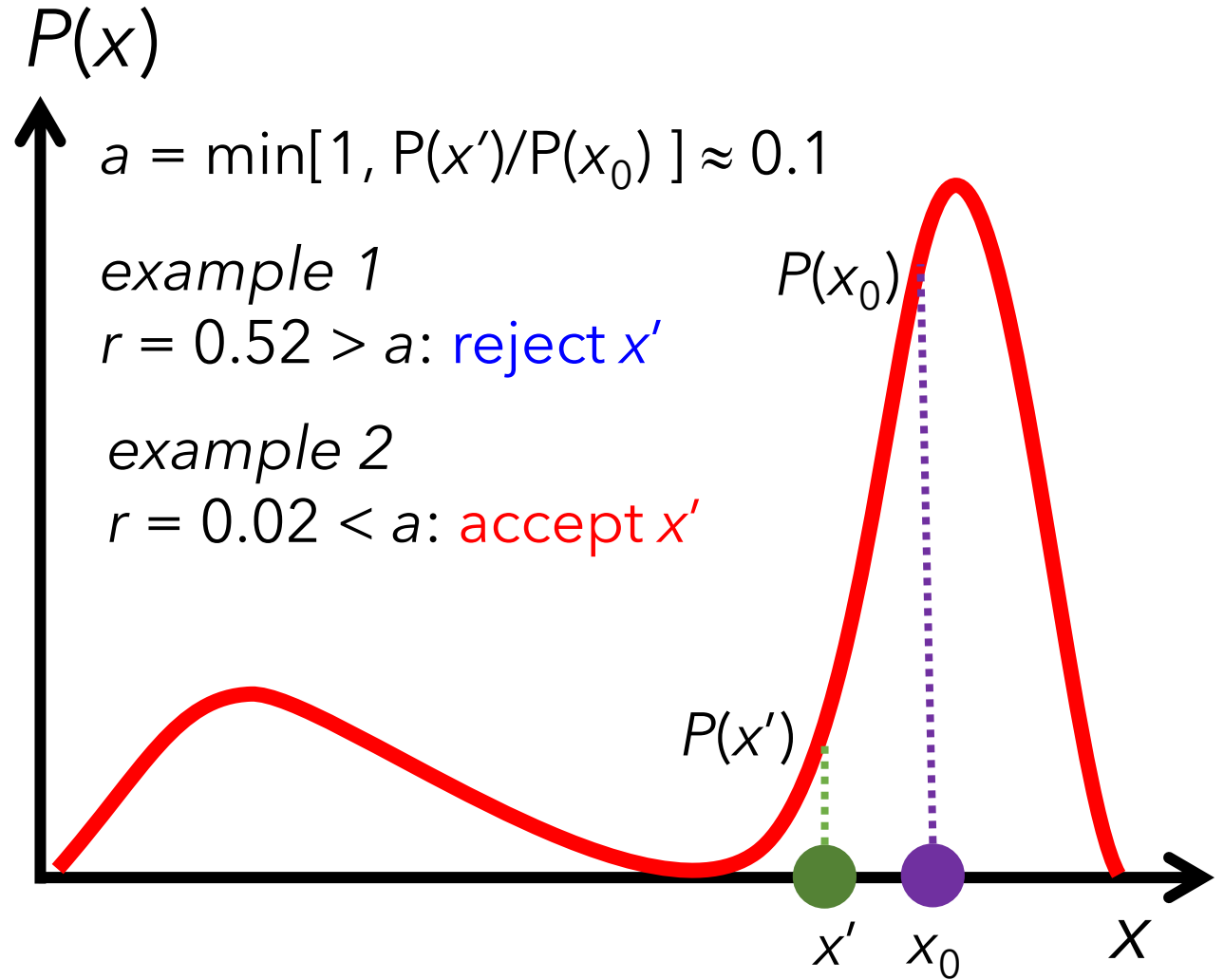
$$g(x|x_n) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{(x-x_n)^2}{\sigma^2}\right)$$



1. Sample a random point  $x_0$ .
2. Sample a second random point  $x'$  using  $g(x|x_0)$ .
3. Compute the jump probability  $a = \min[1, P(x')/P(x_0)]$
4. Sample a uniform random  $r$  number in  $[0,1]$ .
5. If  $r \leq a$ :  
accept  $x' \rightarrow x_1 = x'$   
If  $r > a$ :  
reject  $x' \rightarrow x_1 = x_0$
6. Go to step 2 and repeat.



1. Sample a random point  $x_0$ .
2. Sample a second random point  $x'$  using  $g(x|x_0)$ .
3. Compute the jump probability  $a = \min[1, P(x')/P(x_0)]$
4. Sample a uniform random  $r$  number in  $[0,1]$ .
5. If  $r \leq a$ :  
accept  $x' \rightarrow x_1 = x'$   
If  $r > a$ :  
reject  $x' \rightarrow x_1 = x_0$
6. Go to step 2 and repeat.



```

def P(x, b, mu):
    probability = np.exp(-abs(x - mu) / b) / (2 * b)
    return probability
# Define distribution (e.g., Laplace PDF)

(b, mu) = (1.0, 0.0)
(initial_state, proposal_std) = (0, 0.5)
num_samples = 10000 # Number of points
def metropolis_sampler(initial_state, num_samples, proposal_std, b, mu):
    samples = [initial_state]
    for _ in range(num_samples): # Loop over all points
        proposed_state = np.random.normal(samples[-1], proposal_std) # Proposal distribution is a Gaussian
        acceptance_ratio = min(1, P(proposed_state, b, mu) / P(samples[-1], b, mu))
        if np.random.rand() < acceptance_ratio: # Accept or reject
            samples.append(proposed_state)
        else:
            samples.append(samples[-1])
    return np.array(samples[1:])

samples = metropolis_sampler(initial_state, num_samples, proposal_std, b, mu)
print("First 5 values: ", samples[:5])

```

For exploring the PES, it may be helpful to take the *proposal distribution*  $g(x|x_n)$  as the Boltzmann distribution

$$g(x|x_n) = \exp\left(-\frac{E(x) - E(x_n)}{k_B T}\right)$$

If  $g(x|x_n) = g(x_n|x)$

the algorithm is called Metropolis.

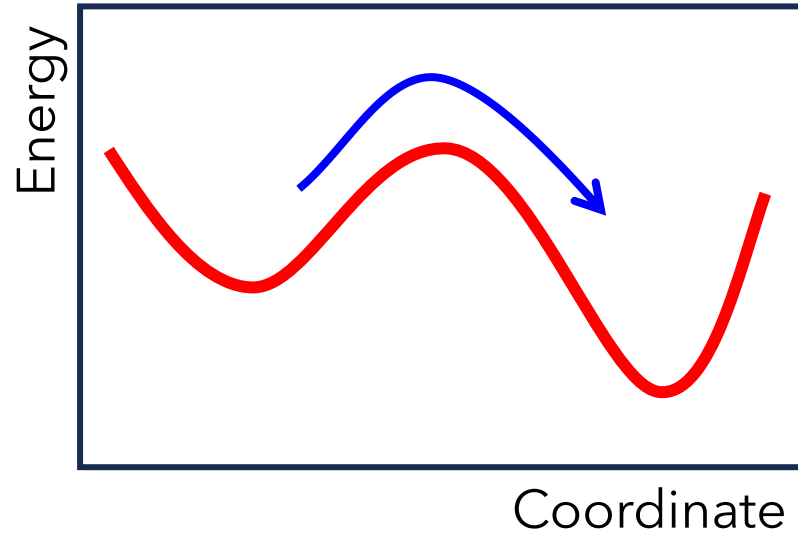
If  $g(x|x_n) \neq g(x_n|x)$

the algorithm is called Metropolis-Hastings.

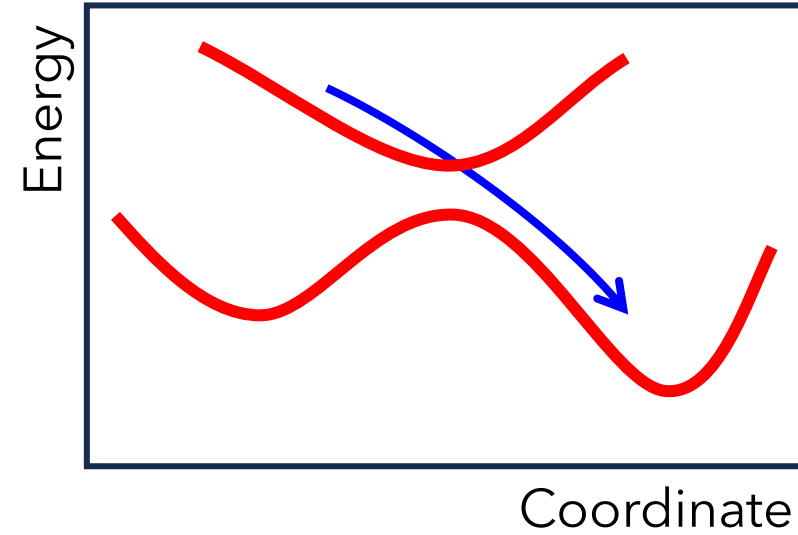
# Rate theory

In diverse situations, stochastic treatment is advantageous over dynamics.

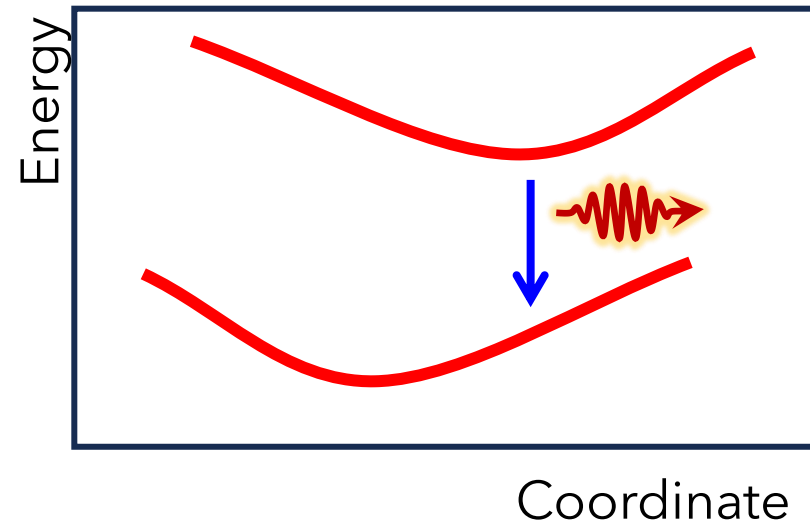
Reaction



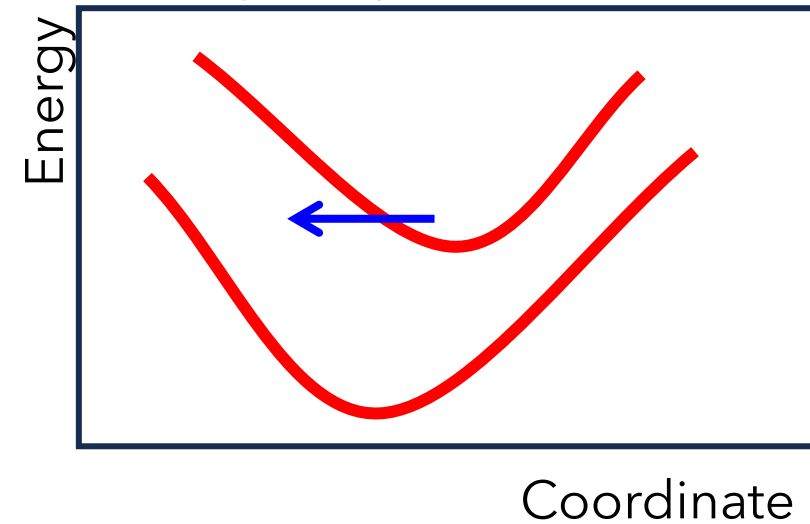
Internal conversion



Spontaneous emission



Weakly-coupled transition

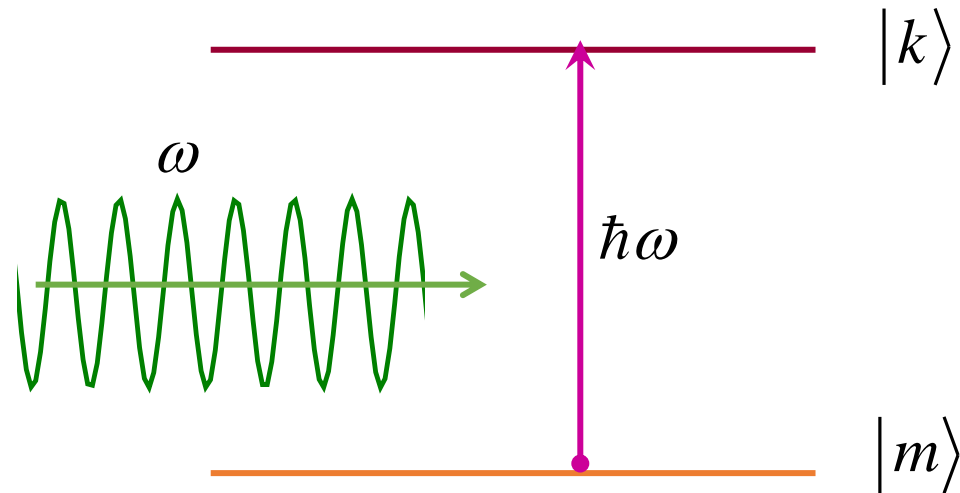




We want to define these processes' probability per unity of time (rate).

This is done with **Fermi's golden rule**.

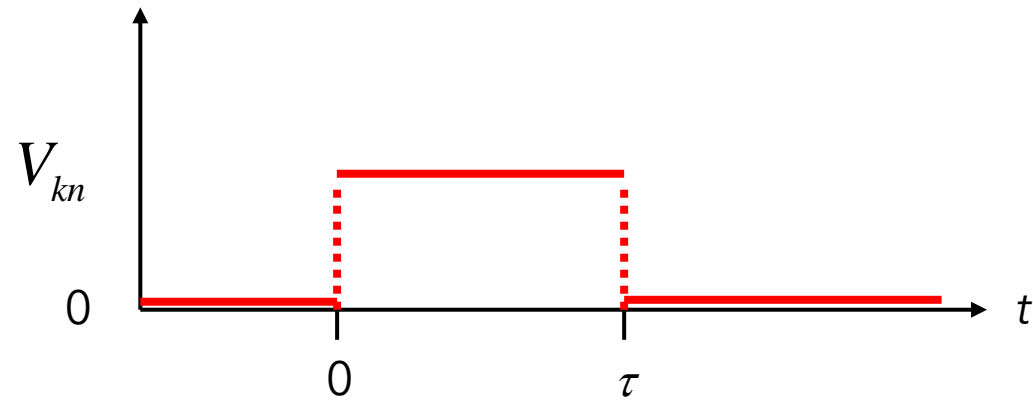
$$W_k = \frac{2\pi}{\hbar} |V_{mk}|^2 \delta(\Delta E_{mk} + \hbar\omega)$$



See derivation in the appendix to this presentation.

# Fermi's Golden Rule

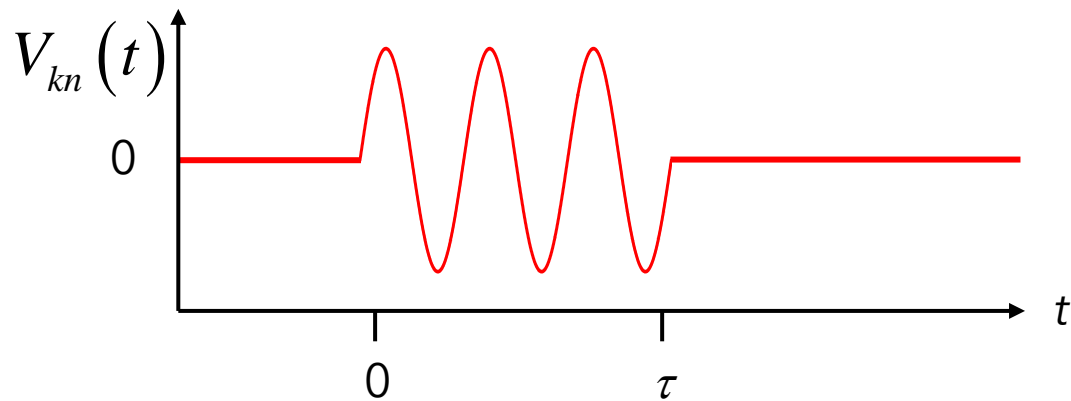
Constant perturbation



$$W_{km} = \frac{2\pi}{\hbar} |V_{km}|^2 \delta(\varepsilon_k - \varepsilon_m)$$

$P_{mk}$  is max for  $\varepsilon_k - \varepsilon_m = 0$   
(Internal conversion, collision)

Harmonic perturbation



$$W_{km} = \frac{2\pi}{\hbar} |V_{km}|^2 \delta(\varepsilon_k - \varepsilon_m \pm \hbar\omega)$$

$P_{mk}$  is max for  $\hbar\omega = \mp(\varepsilon_m - \varepsilon_k)$   
(photon absorption, emission)

The rate has units of time inverse

$$W_{km} = \frac{2\pi}{\hbar} |V_{km}|^2 \delta(\varepsilon_k - \varepsilon_m)$$

$$\begin{aligned} [W] &= \frac{1}{[Et]} [E]^2 \times \frac{1}{[E]} \\ &= \frac{1}{[t]} \end{aligned}$$

The lifetime of the process is the inverse of the rate:

$$\tau_{km} = \frac{1}{W_{km}}$$

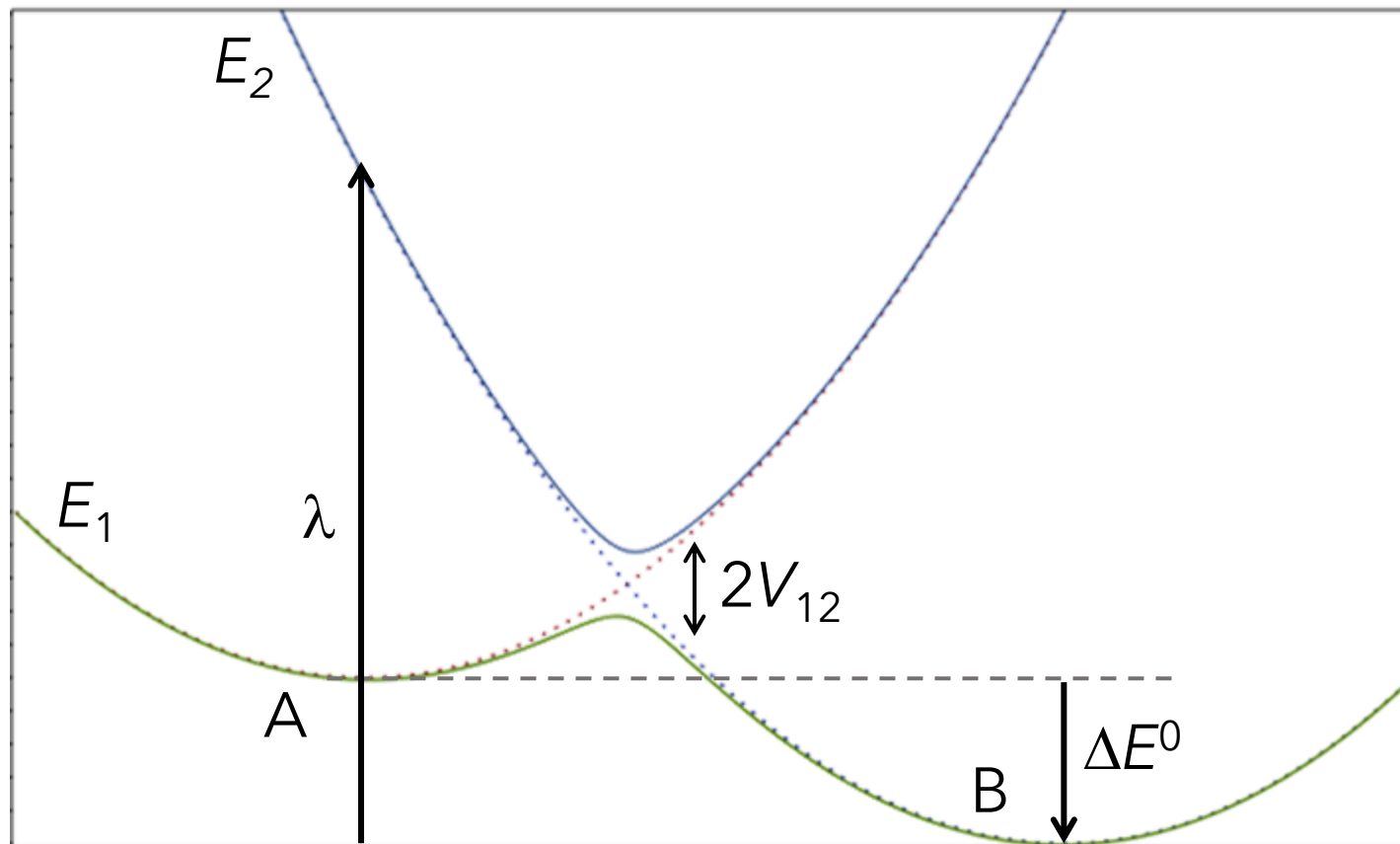
Fermi's golden rule is used to derive:

- Marcus rate (unimolecular reactions, internal conversion, charge transfer)
- Spontaneous emission rate
- Absorption and stimulated emission rates
- Landau-Zener rate (internal conversion)
- ...

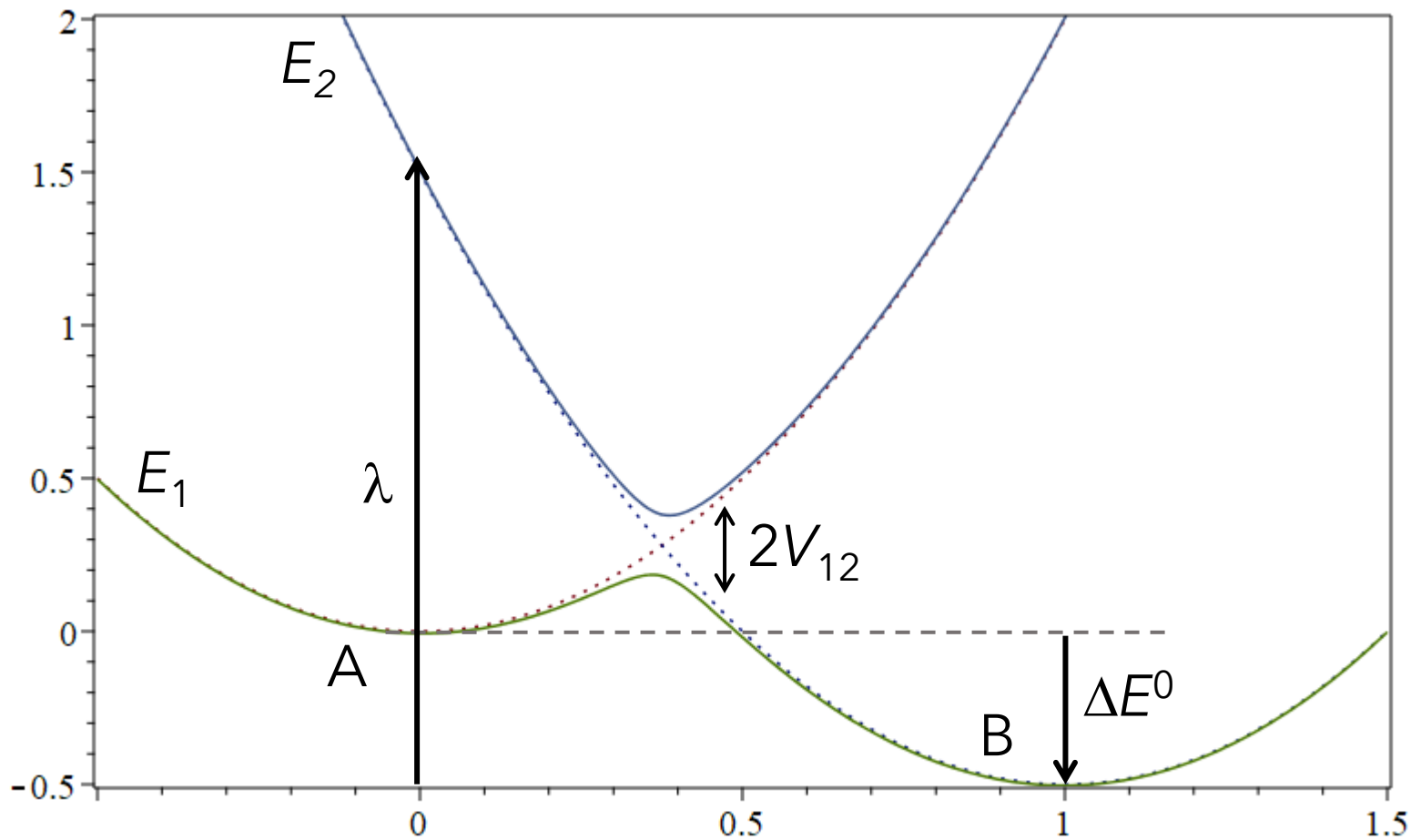
**Marcus rate**

# Marcus rate

$$W_{AB}^{(Marcus)} = \frac{2\pi}{\hbar} |V_{12}|^2 \frac{1}{\sqrt{4\pi\lambda k_B T}} \exp\left(-\frac{(\lambda + \Delta E^0)^2}{4\lambda k_B T}\right)$$



$$W_{AB}^{(Marcus)} = \frac{2\pi}{\hbar} |V_{12}|^2 \frac{1}{\sqrt{4\pi\lambda k_B T}} \exp\left(-\frac{(\lambda + \Delta E^0)^2}{4\lambda k_B T}\right)$$

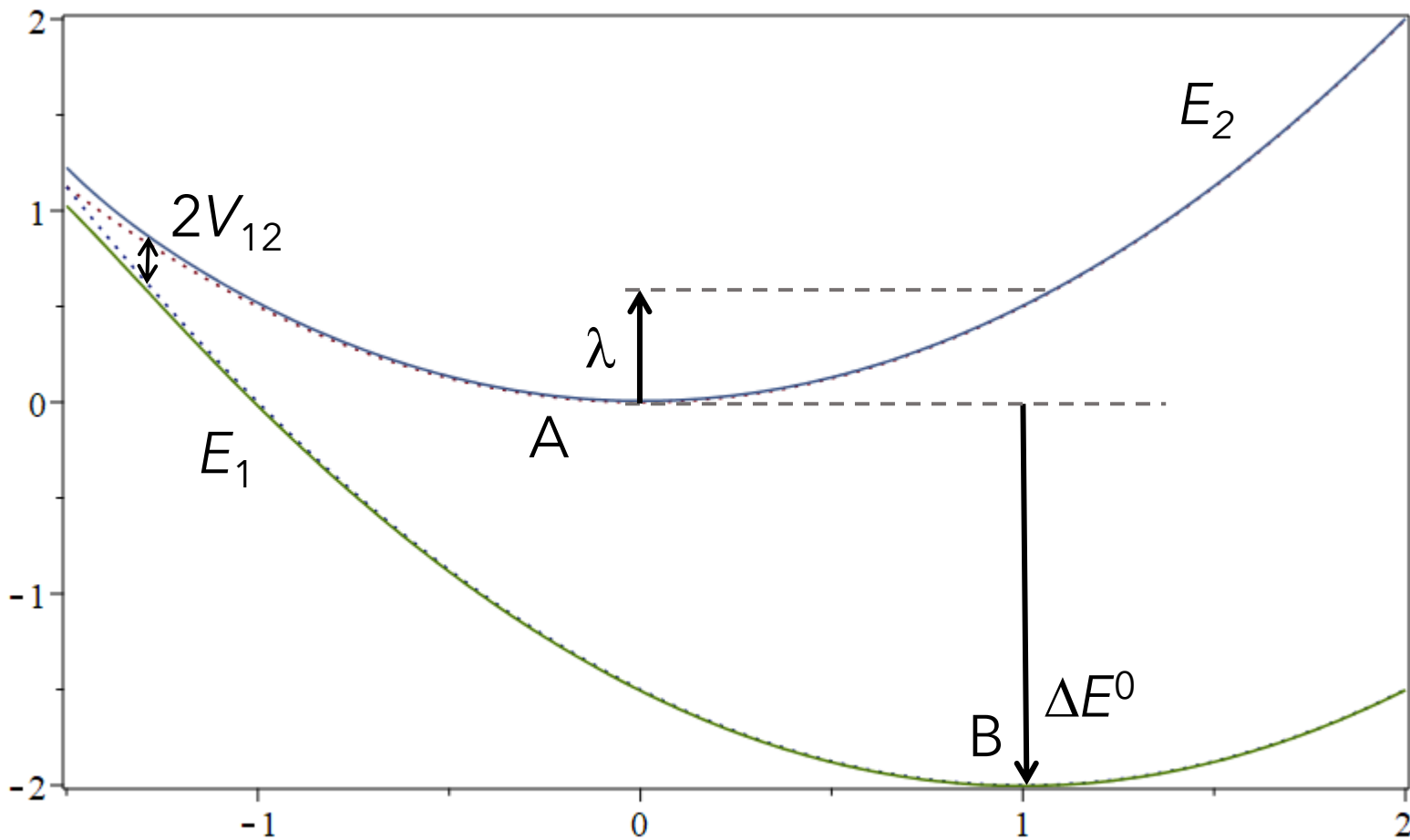


- Adiabatic energy gap  $\Delta E^0 = -1.0$  eV
- Reorganization energy  $\lambda = 1.5$  eV
- Diabatic coupling  $V_{12} = 0.1$  eV

Rate (300 K):  
 $W_{A \rightarrow B} = 9.4 \times 10^{11} \text{ s}^{-1}$   
 Lifetime:  
 $\tau_{A \rightarrow B} = 1.1 \times 10^{-12} \text{ s}$



$$W_{AB}^{(Marcus)} = \frac{2\pi}{\hbar} |V_{12}|^2 \frac{1}{\sqrt{4\pi\lambda k_B T}} \exp\left(-\frac{(\lambda + \Delta E^0)^2}{4\lambda k_B T}\right)$$



- Adiabatic energy gap  
 $\Delta E^0 = -2.0$  eV
- Reorganization energy  
 $\lambda = 0.5$  eV
- Diabatic coupling  
 $V_{12} = 0.1$  eV

Rate (300 K):

$$W_{A \rightarrow B} = 2.7 \times 10^{13} \text{ s}^{-1}$$

Lifetime:

$$\tau_{A \rightarrow B} = 3.7 \times 10^{-14} \text{ s}$$

Marcus rate will work well if:

- The system is harmonic
- Both states share the same vibrational frequency
- The frequency is small compared to  $k_B T$  ( $< 200 \text{ cm}^{-1}$  at 300 K)

For high frequencies, the Marcus-Levich-Jortner model is an alternative

$$W_{A \rightarrow B}^{(MLJ)} = \frac{2\pi}{\hbar} |V_{12}|^2 \frac{1}{\sqrt{4\pi\lambda k_B T}} e^{-S_c} \sum_{n=0}^{\infty} \frac{S_c^n}{n!} \exp\left(-\frac{(\lambda + \Delta E^0 + n\hbar\omega_c)^2}{4\lambda k_B T}\right)$$

$S_c$  is the electron-phonon coupling (Huang-Rhys factor)

$S_c$  is implemented in MOMAP

Bixon; Jortner, In *Electron transfer from isolated molecules to biomolecules*, **1999**

Bozzi; Rocha. *J Chem Theory Comput* **2023**, 19, 2316

**Spontaneous emission rate**

# Spontaneous emission rate

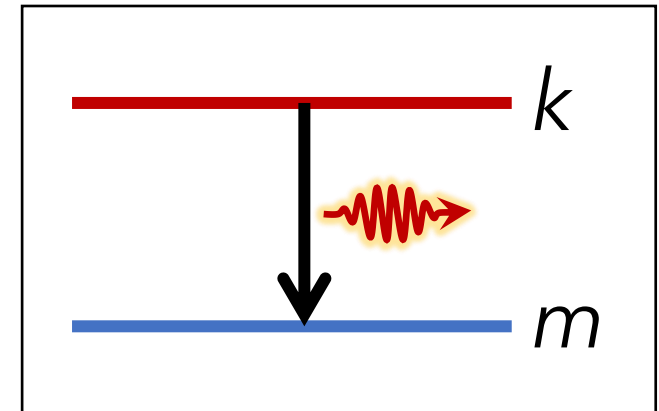
$$W_{km}^{(Sp.Em.)} = \frac{1}{\tau_{km}^{(S.Em.)}} = \frac{4}{3} \frac{\alpha}{\hbar^3 c^2 e^2} \Delta E_{km}^3 \mu_{km}^2$$

$\mu_{km}^2 = |\boldsymbol{\mu}_{km}|^2$  is the transition dipole moment

$\alpha = \frac{e^2}{4\pi\epsilon_0 \hbar c} \approx \frac{1}{137}$  is the fine structure constant

In terms of the oscillator strength  $f_{km} = \frac{2m_e}{3\hbar^2 e^2} \Delta E_{km} \mu_{km}^2$

$$W_{km}^{(Sp.Em.)} = \frac{1}{\tau_{km}^{(S.Em.)}} = \frac{2\alpha}{\hbar m_e c^2} \Delta E_{km}^2 f_{km}$$



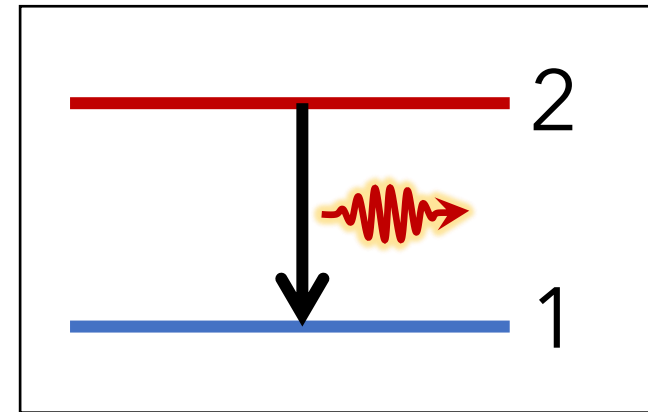
$$W_{km}^{(Sp.Em.)} = \frac{1}{\tau_{km}^{(Sp.Em.)}} = \frac{2\alpha}{\hbar m_e c^2} \Delta E_{km}^2 f_{km}$$

$$\Delta E_{21} = 2.0 \text{ eV}$$

$$f_{21} = 1.0$$

$$W_{21}^{(Sp.Em.)} = 1.7 \times 10^8 \text{ s}^{-1}$$

$$\tau_{21}^{(Sp.Em.)} = 5.8 \times 10^{-9} \text{ s}$$



To know more:

Monte Carlo & Metropolis-Hastings

[en.wikipedia.org/wiki/Monte\\_Carlo\\_method](https://en.wikipedia.org/wiki/Monte_Carlo_method)

[en.wikipedia.org/wiki/Metropolis-Hastings\\_algorithm](https://en.wikipedia.org/wiki/Metropolis-Hastings_algorithm)

Einstein Coefficients, cross-sections, oscillator strengths, dipole-moments

- Hilborn. *Am J Phys* **1982**, 50, 982

Papers available for download at:

[amubox.univ-amu.fr/s/xXAiMZrDPb9RMRX](https://amubox.univ-amu.fr/s/xXAiMZrDPb9RMRX) (Ask me for the password)

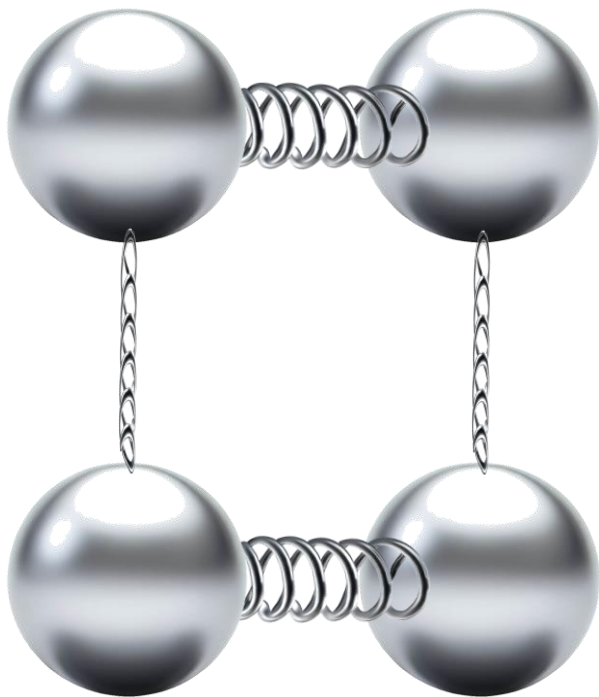
# **Appendix**

## **Zero-point energy spiling**



# Classical coupled oscillators

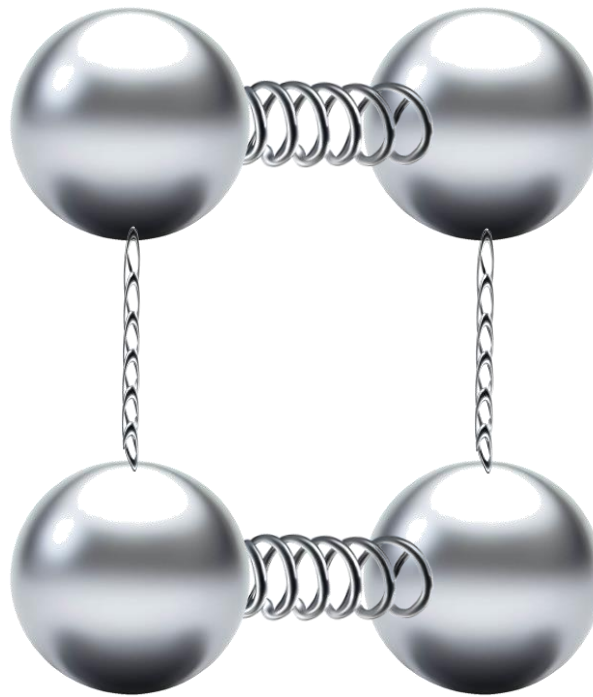
$$E_{top} = \varepsilon_0$$



$$E_{bot} = 0$$

Time 

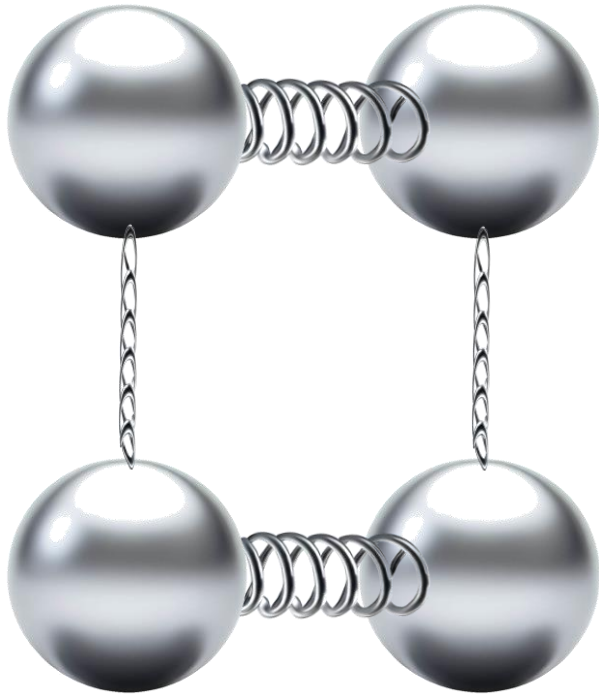
$$E_{top} = 0$$



$$E_{bot} = \varepsilon_0$$

# Quantum coupled oscillators

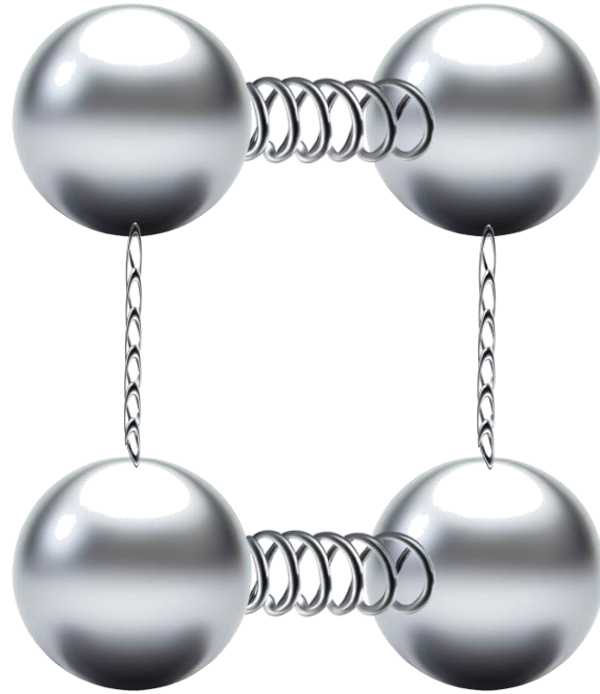
$$E_{top} = \varepsilon_0 + \Delta\varepsilon$$



$$E_{bot} = \varepsilon_0$$

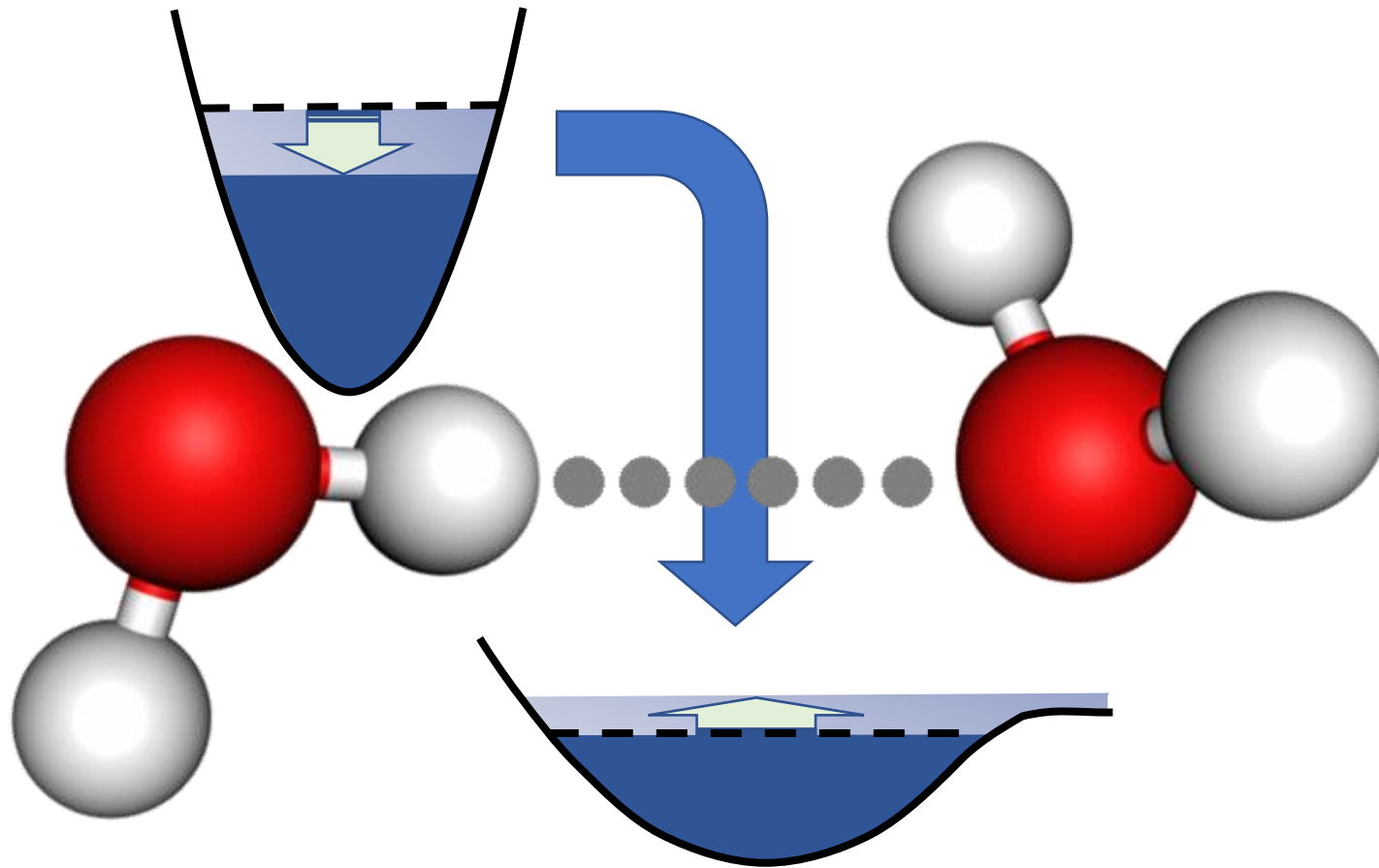
Time 

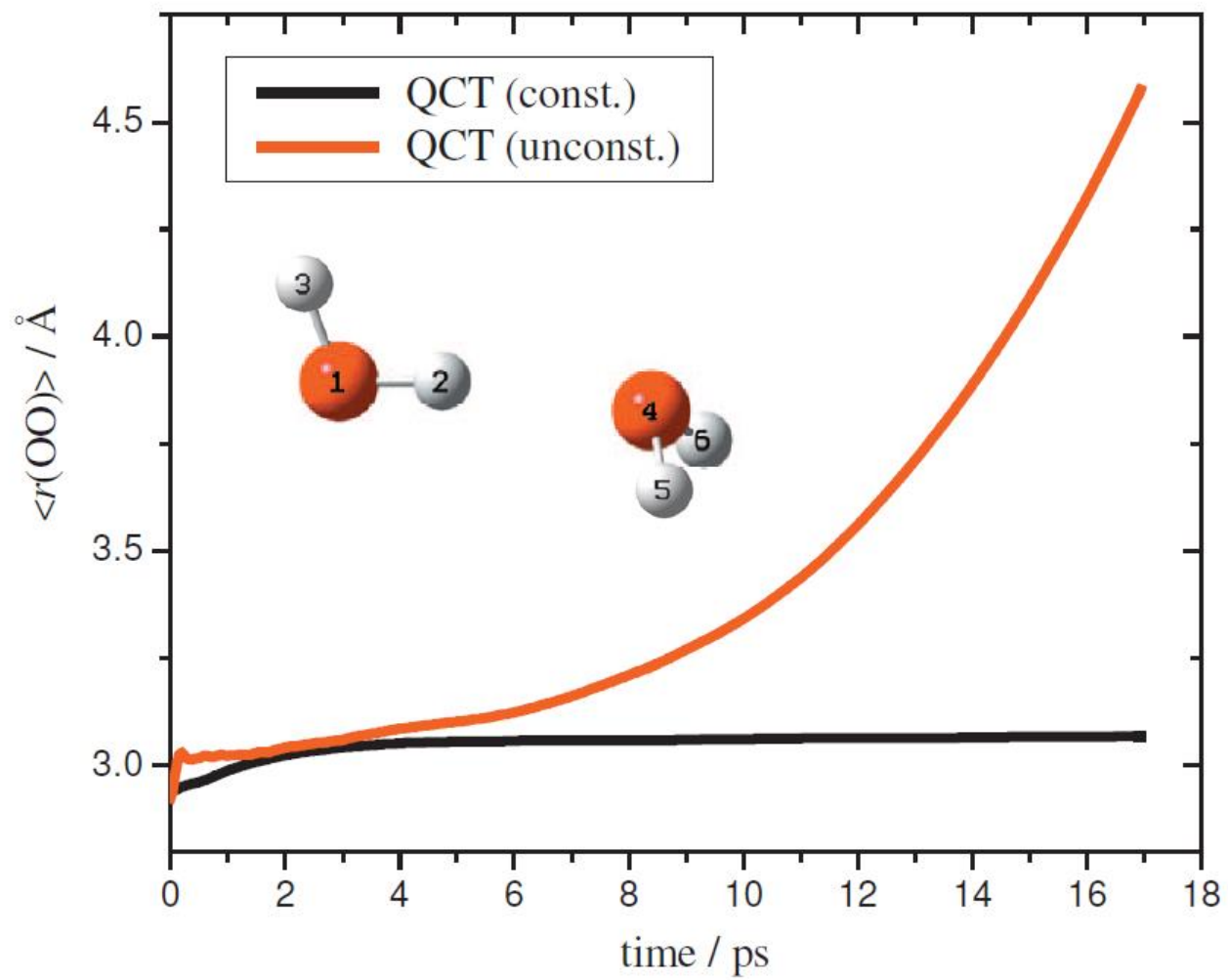
$$E_{top} = \varepsilon_0$$



$$E_{bot} = \varepsilon_0 + \Delta\varepsilon$$

Classical molecular dynamics can artificially transfer too much energy between vibrational modes because it has no zero-point energy constraint.





There are many ZPE-corrections proposed.

But they either require **energy Hessians**  
or introduce **statistical biases**.

Miller; Hase; Darling. *J Chem Phys* **1989**, 91, 2863

Bowman; Gazdy; Sun. *J Chem Phys* **1989**, 91, 2859

Varandas; Marques. *J Chem Phys* **1994**, 100, 1908

Guo; Thompson; Sewell. *J Chem Phys* **1996**, 104, 576

....

Ceriotti; Bussi; Parrinello. *Phys Rev Lett* **2009**, 103, 030603

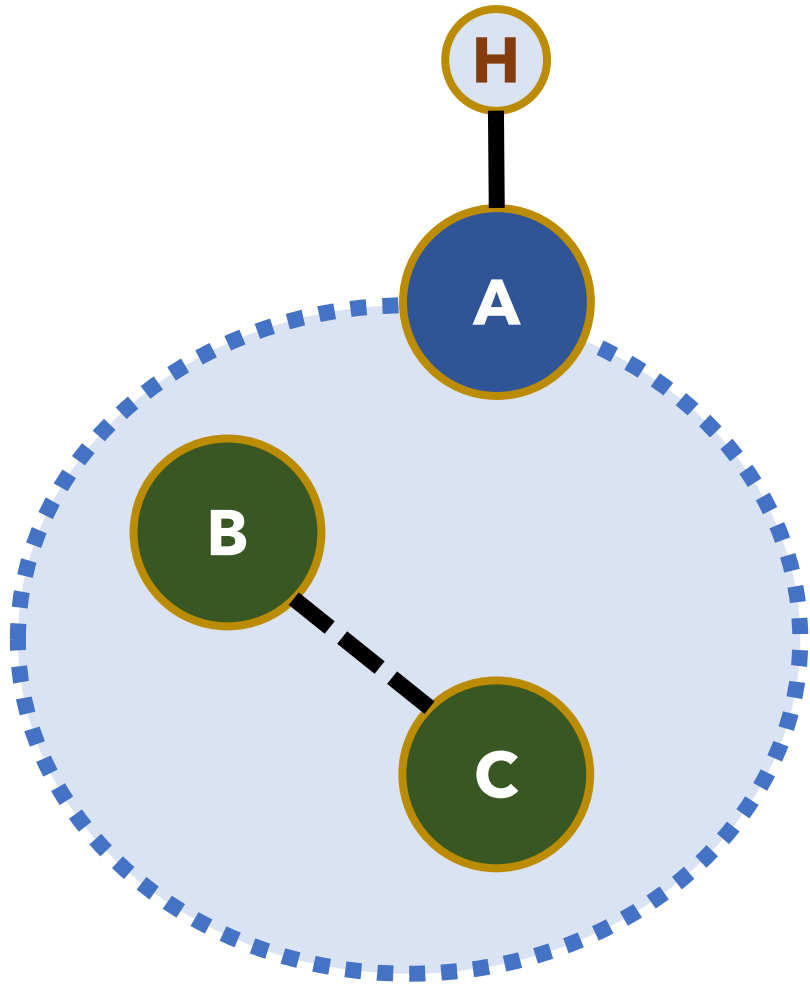
Czakó; Kaledin; Bowman. *J Chem Phys* **2010**, 132, 164103

# Local-Pair ZPE correction

LP-ZPE monitors the energy in pairs of atoms.  
For this reason, it doesn't need the Hessian.

It works like a thermostat, transferring kinetic energy from the slow vibrations into fast vibrations, whenever a ZPE drop is detected.

The correction conserves total energy, total momentum, and angular momentum of the pairs.



1. Define the mean kinetic energy the A-H pair must have in the internuclear direction

$$\bar{\mathcal{E}}_k^0$$

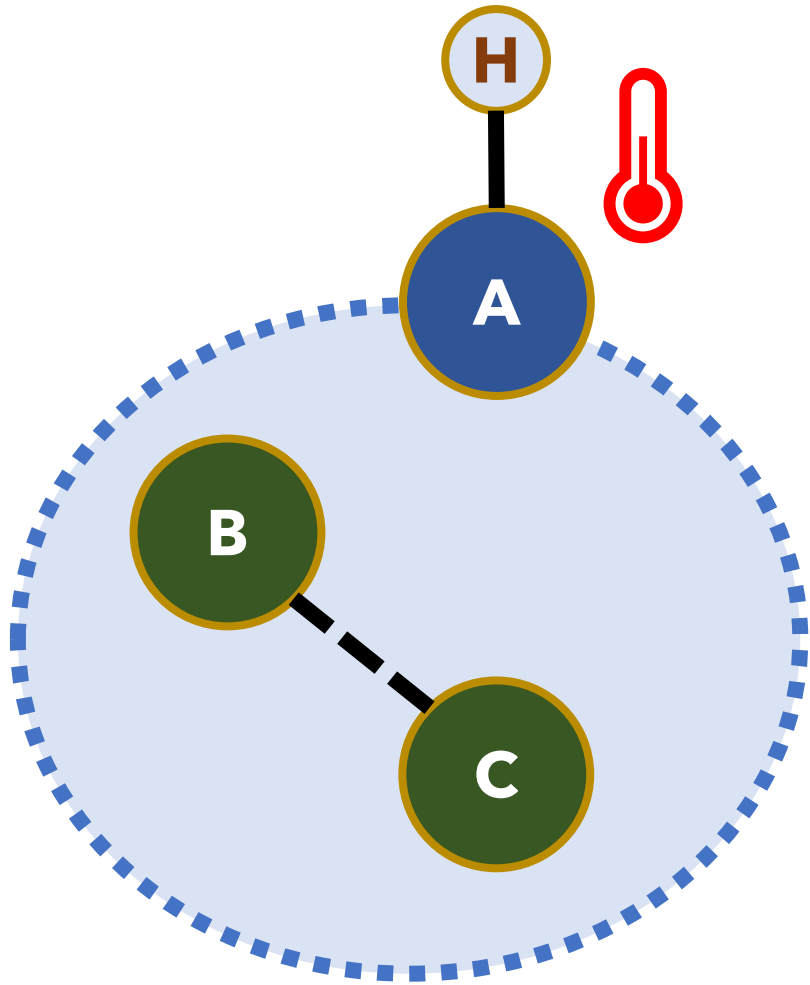
2. During dynamics, periodically check this mean kinetic energy

$$\bar{\mathcal{E}}_k(\tau)$$

3. If it drops

$$\bar{\mathcal{E}}_k(\tau) > \bar{\mathcal{E}}_k^0$$

apply LP-ZPE



4. Add the missing energy to the A-H internuclear direction

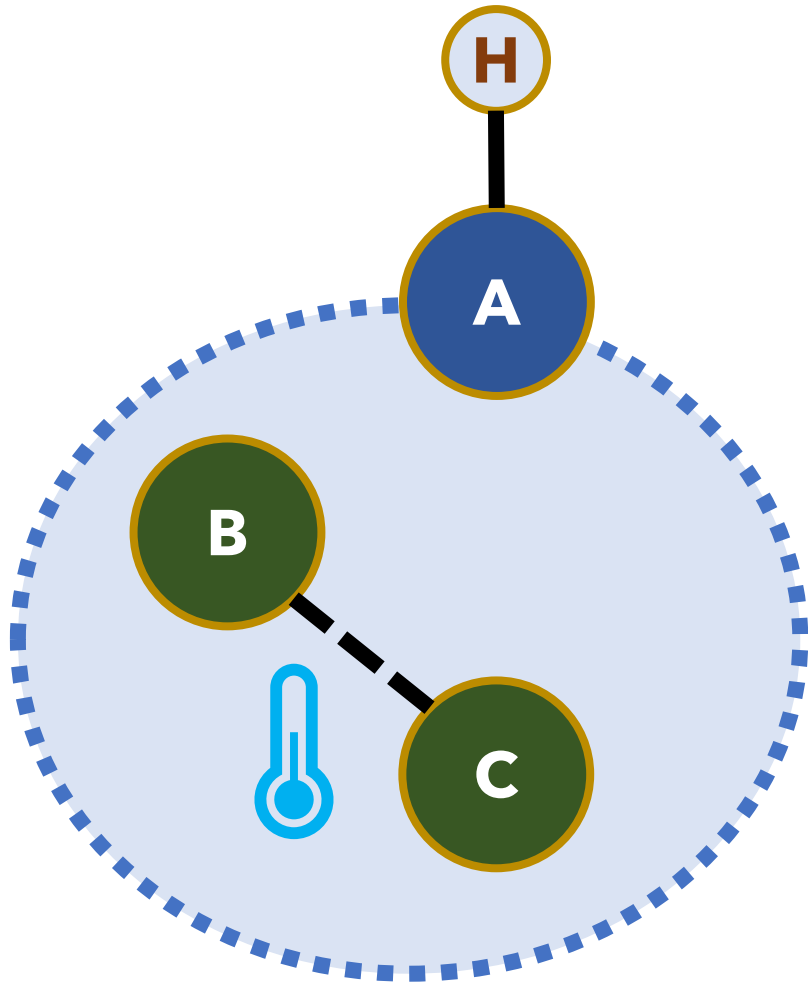
$$\mathbf{v}_{H_{new}} = \mathbf{v}_{H_{old}} + \frac{\mu_{AH}}{m_H} \Delta_{AH} \hat{\mathbf{u}}_{AH}$$

$$\mathbf{v}_{A_{new}} = \mathbf{v}_{A_{old}} - \frac{\mu_{AH}}{m_A} \Delta_{AH} \hat{\mathbf{u}}_{AH}$$

$$\Delta_{AH} = \left[ \left( v_{H_{old}}^{\parallel} - v_{A_{old}}^{\parallel} \right)^2 + \left( \frac{2\delta\varepsilon}{\mu_{AH}} \right)^2 \right]^{1/2} - \left( v_{H_{old}}^{\parallel} - v_{A_{old}}^{\parallel} \right)$$

$$\hat{\mathbf{u}}_{AH} = \frac{\mathbf{R}_A - \mathbf{R}_H}{\|\mathbf{R}_A - \mathbf{R}_H\|}$$





5. Remove this same amount from all B-C pairs

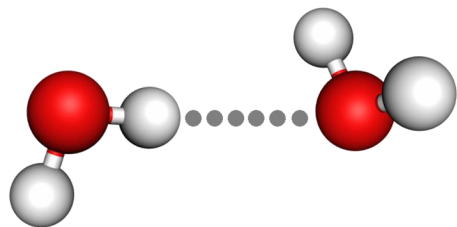
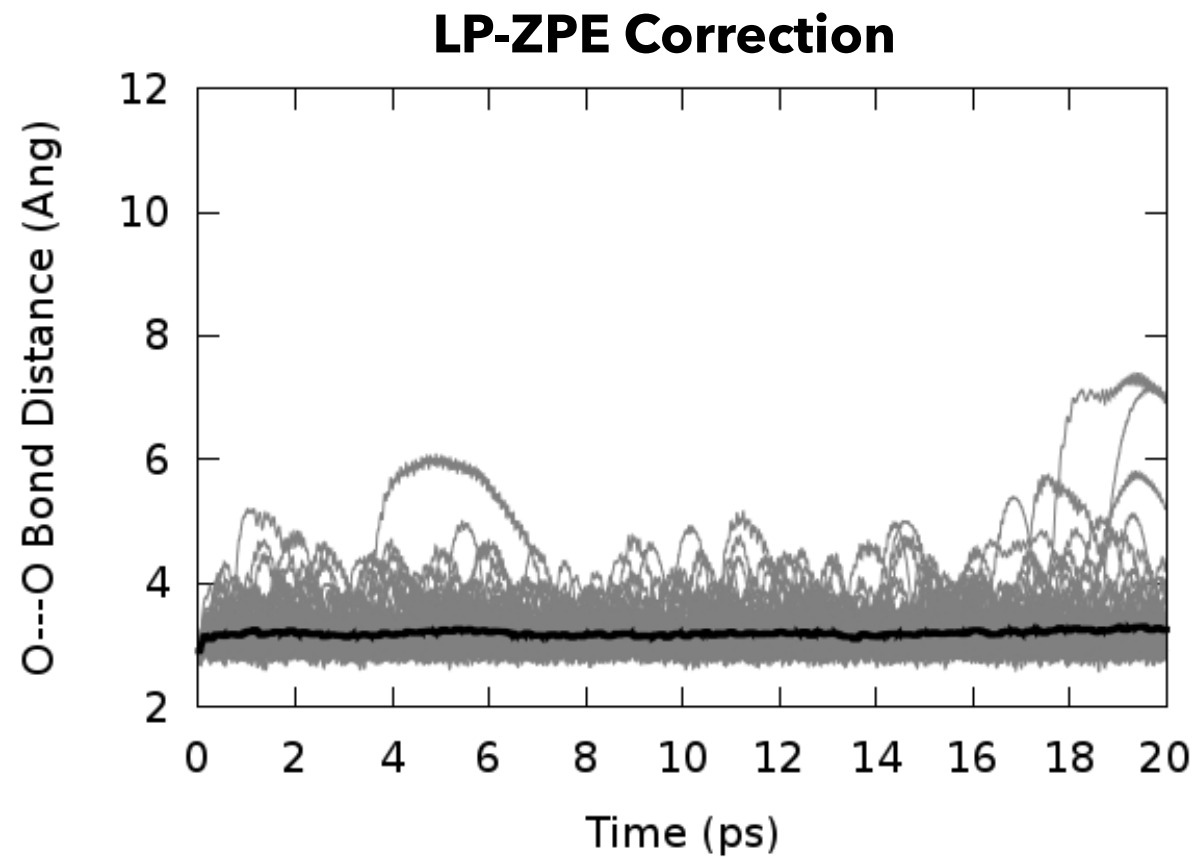
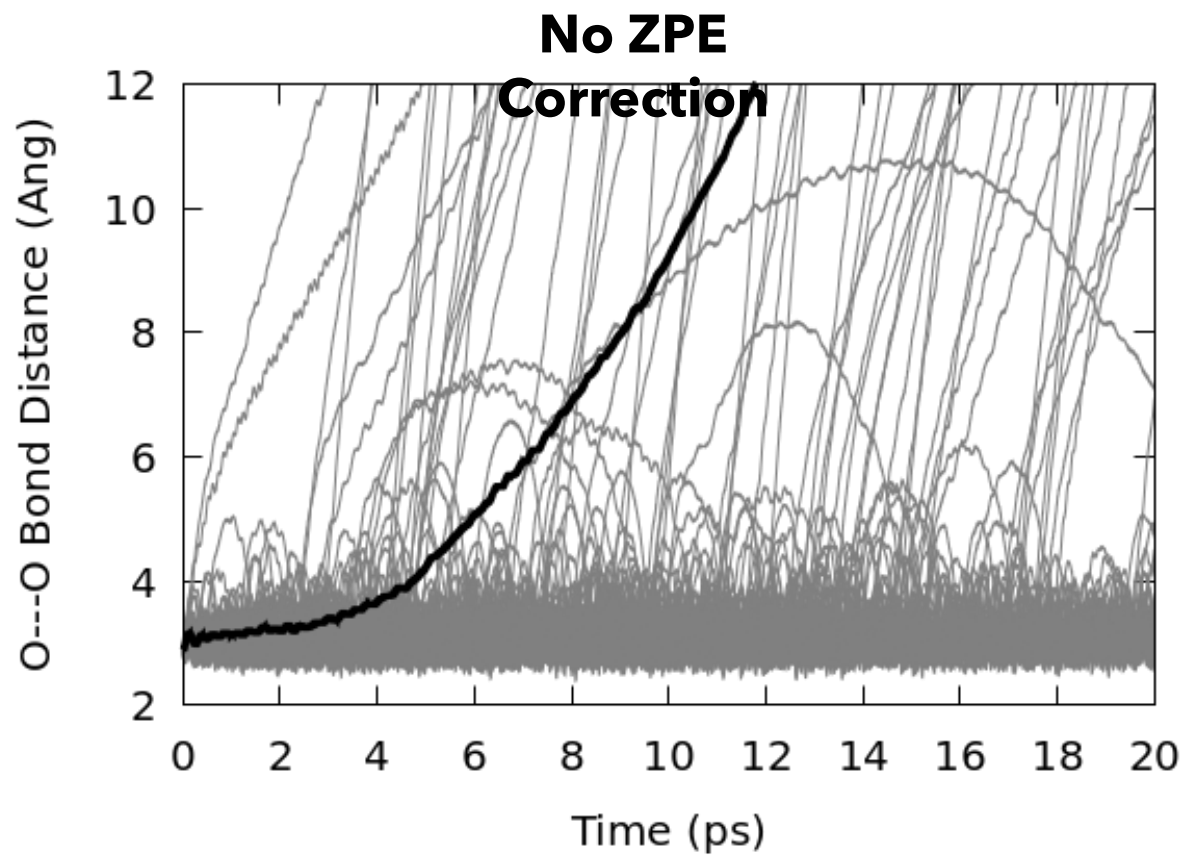
$$\mathbf{v}_{B_{new}} = \mathbf{v}_{B_{old}} + \frac{\mu_{BC}}{m_B} \Delta_{BC} \hat{\mathbf{u}}_{BC}$$

$$\mathbf{v}_{C_{new}} = \mathbf{v}_{C_{old}} - \frac{\mu_{BC}}{m_C} \Delta_{BC} \hat{\mathbf{u}}_{BC}$$

$$\Delta_{BC} = \left[ \left( \mathbf{v}_{B_{old}}^{\parallel} - \mathbf{v}_{C_{old}}^{\parallel} \right)^2 - \left( \frac{2\delta\epsilon}{\mu_{BC}} f_{BC} \right) \right]^{1/2} - \left( \mathbf{v}_{B_{old}}^{\parallel} - \mathbf{v}_{C_{old}}^{\parallel} \right)$$

Each B-C pair contributes proportionally to the energy they have

$$f_{BC} = \frac{\frac{1}{2} \mu_{BC} \left( \mathbf{v}_{B_{old}}^{\parallel} - \mathbf{v}_{C_{old}}^{\parallel} \right)^2}{\sum_{B'C'} \frac{1}{2} \mu_{B'C'} \left( \mathbf{v}_{B'_{old}}^{\parallel} - \mathbf{v}_{C'_{old}}^{\parallel} \right)^2}$$



- $S_0$  MP2/aug-cc-pVDZ
- 100 trajs; 20 ps;  $\Delta t = 0.5$  fs
- ZPE correction every 10 fs

# **Appendix**

## **Derivation of the Fermi's Golden Rule**

Suppose we have a quantum system with states  $|\phi_i\rangle$ , which are solutions of

$$H_0 |\phi_i\rangle = \varepsilon_i |\phi_i\rangle$$

This system is initially in  $|\phi_k\rangle$  when it is disturbed by an interaction  $V(t)$ . The perturbed system is described by

$$i\hbar \frac{\partial |\psi\rangle}{\partial t} = (H_0 + V) |\psi\rangle$$

expand  $|\psi(t)\rangle = \sum_n c_n(t) e^{-i\varepsilon_n t/\hbar} |\phi_n\rangle \quad |\psi(0)\rangle = |\phi_m\rangle \quad c_n(0) = \delta_{nm}$

replace  $i\hbar \frac{\partial}{\partial t} \sum_n c_n(t) e^{-i\varepsilon_n t/\hbar} |\phi_n\rangle = (H_0 + V(t)) \sum_n c_n(t) e^{-i\varepsilon_n t/\hbar} |\phi_n\rangle$

Left side

$$i\hbar \sum_n \left( \frac{d}{dt} c_n(t) \right) e^{-i\varepsilon_n t/\hbar} |\phi_n\rangle + i\hbar \sum_n c_n(t) \left( \frac{d}{dt} e^{-i\varepsilon_n t/\hbar} \right) |\phi_n\rangle$$

$$= i\hbar \sum_n \frac{dc_n(t)}{dt} e^{-i\varepsilon_n t/\hbar} |\phi_n\rangle + i\hbar \sum_n c_n(t) \left( -\frac{i\varepsilon_n}{\hbar} \right) e^{-i\varepsilon_n t/\hbar} |\phi_n\rangle$$

$$= i\hbar \sum_n \frac{dc_n(t)}{dt} e^{-i\varepsilon_n t/\hbar} |\phi_n\rangle + \sum_n c_n(t) \varepsilon_n e^{-i\varepsilon_n t/\hbar} |\phi_n\rangle$$

Right side

$$(H_0 + V(t)) \sum_n c_n(t) e^{-i\varepsilon_n t/\hbar} |\phi_n\rangle$$

$$= \sum_n c_n(t) \varepsilon_n e^{-i\varepsilon_n t/\hbar} |\phi_n\rangle + \sum_n c_n(t) V(t) e^{-i\varepsilon_n t/\hbar} |\phi_n\rangle$$

expand

$$|\psi(t)\rangle = \sum_n c_n(t) e^{-i\varepsilon_n t/\hbar} |\phi_n\rangle \quad |\psi(0)\rangle = |\phi_m\rangle \quad c_n(0) = \delta_{nm}$$

replace

$$i\hbar \sum_n \frac{dc_n(t)}{dt} e^{-i\varepsilon_n t/\hbar} |\phi_n\rangle + \sum_n c_n(t) \varepsilon_n e^{-i\varepsilon_n t/\hbar} |\phi_n\rangle =$$
$$\sum_n c_n(t) \varepsilon_n e^{-i\varepsilon_n t/\hbar} |\phi_n\rangle + \sum_n c_n(t) V(t) e^{-i\varepsilon_n t/\hbar} |\phi_n\rangle$$

project

$$i\hbar \sum_n \frac{dc_n(t)}{dt} e^{-i\varepsilon_n t/\hbar} \langle \phi_k | \phi_n \rangle + \sum_n c_n(t) \cancel{\varepsilon_n} e^{-i\varepsilon_n t/\hbar} \langle \phi_k | \phi_n \rangle =$$
$$\sum_n c_n(t) \cancel{\varepsilon_n} e^{-i\varepsilon_n t/\hbar} \langle \phi_k | \phi_n \rangle + \sum_n c_n(t) e^{-i\varepsilon_n t/\hbar} \langle \phi_k | V(t) | \phi_n \rangle$$

$$\langle \phi_k | \phi_n \rangle = \delta_{kn} \quad \langle \phi_k | V(t) | \phi_n \rangle = V_{kn}(t)$$

$$i\hbar \frac{dc_k(t)}{dt} e^{-i\varepsilon_k t/\hbar} = \sum_n c_n(t) e^{-i\varepsilon_n t/\hbar} V_{kn}(t)$$

expand

$$|\psi(t)\rangle = \sum_n c_n(t) e^{-i\varepsilon_n t/\hbar} |\phi_n\rangle \quad |\psi(0)\rangle = |\phi_m\rangle \quad c_n(0) = \delta_{nm}$$

replace

$$i\hbar \sum_n \frac{dc_n(t)}{dt} e^{-i\varepsilon_n t/\hbar} |\phi_n\rangle + \sum_n c_n(t) \varepsilon_n e^{-i\varepsilon_n t/\hbar} |\phi_n\rangle =$$
$$\sum_n c_n(t) \varepsilon_n e^{-i\varepsilon_n t/\hbar} |\phi_n\rangle + \sum_n c_n(t) V(t) e^{-i\varepsilon_n t/\hbar} |\phi_n\rangle$$

project

$$\frac{dc_k(t)}{dt} = -\frac{i}{\hbar} \sum_n c_n(t) e^{-i(\varepsilon_n - \varepsilon_k)t/\hbar} V_{kn}(t)$$

# First-order perturbation solution

$$\frac{dc_k(t)}{dt} = -\frac{i}{\hbar} \sum_n c_n(t) e^{-i\omega_{nk}t} V_{kn}(t) \quad \omega_{nk} = \frac{\varepsilon_n - \varepsilon_k}{\hbar} \quad c_n(0) = \delta_{nm}$$

$$\begin{aligned} \frac{dc_k(t)}{dt} &\approx -\frac{i}{\hbar} \sum_n c_n(0) e^{-i\omega_{nk}t} V_{kn}(t) \\ &= -\frac{i}{\hbar} \sum_n \delta_{nm} e^{-i\omega_{nk}t} V_{kn}(t) = -\frac{i}{\hbar} e^{-i\omega_{mk}t} V_{km}(t) \end{aligned}$$

$$c_k(t) \approx -\frac{i}{\hbar} \int_0^t e^{-i\omega_{mk}t'} V_{km}(t') dt'$$

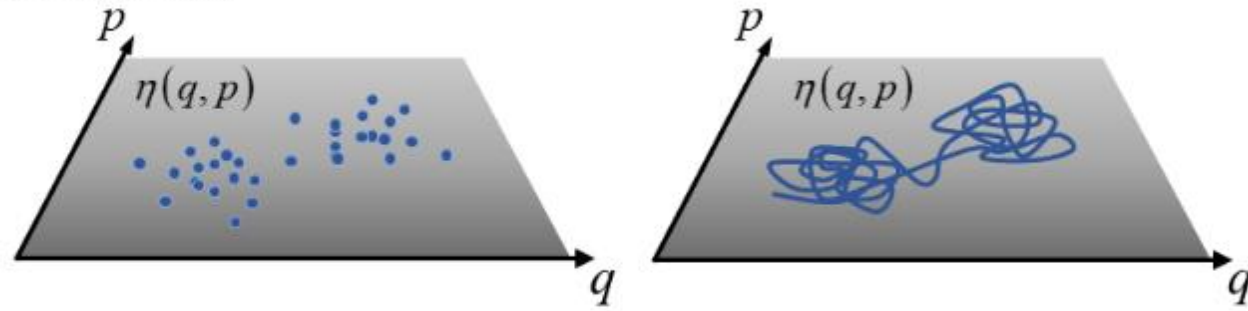
$$P_k(t) = |c_k(t)|^2 = \frac{1}{\hbar^2} \left| \int_0^t e^{-i\omega_{mk}t'} V_{km}(t') dt' \right|^2$$



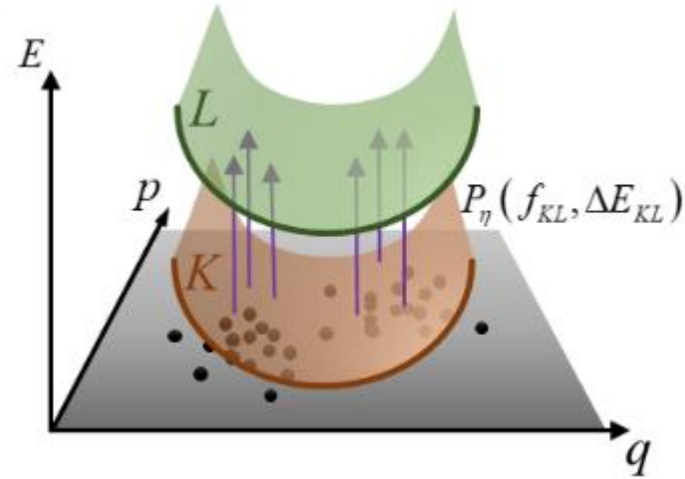
# **Appendix**

## **Wigner x trajectory samplings**

### 1. Sampling

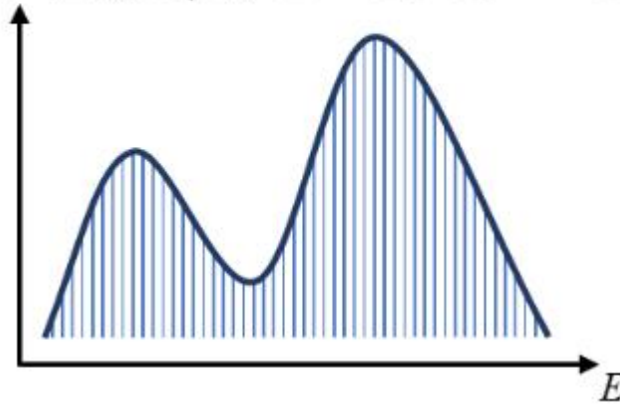


### 2. Transition probabilities

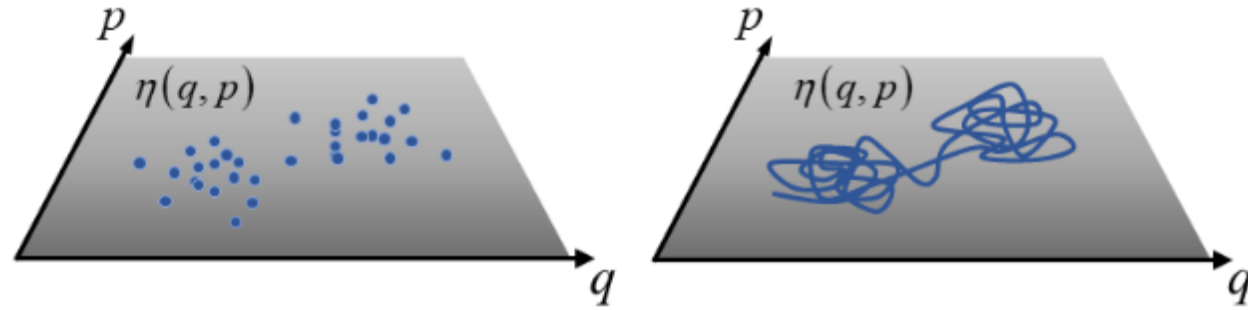


### 3. Spectrum

$$\sigma(E) = \sum_{KL} \sum_{\eta} P_\eta(f_{KL}, \Delta E_{KL}) w_s(E - \Delta E_{KL})$$



## 1. Sampling

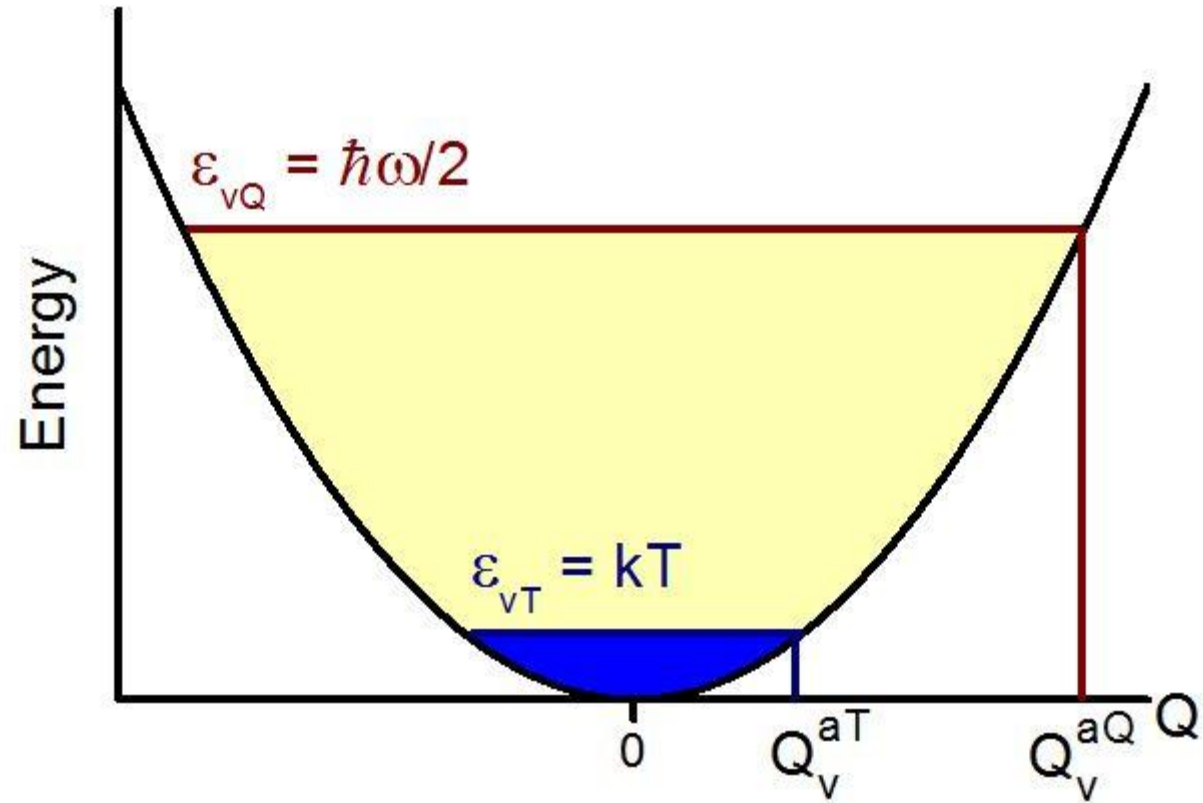


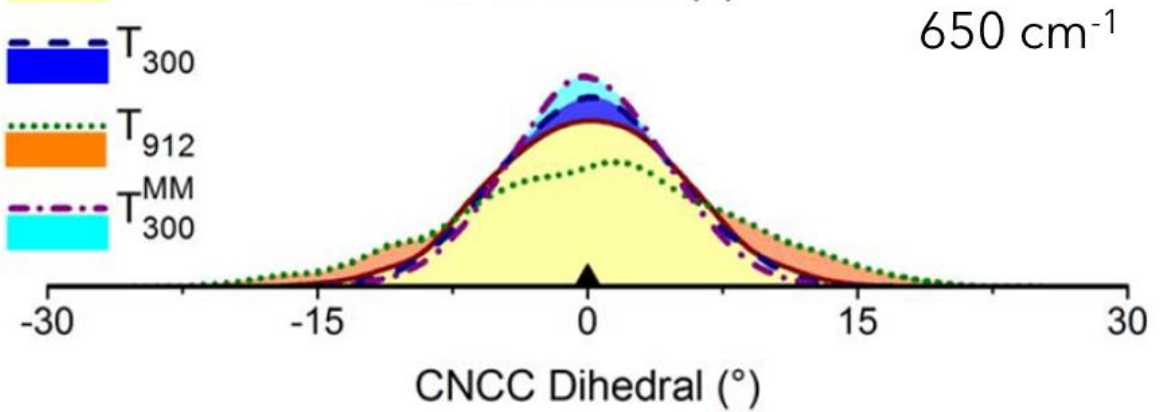
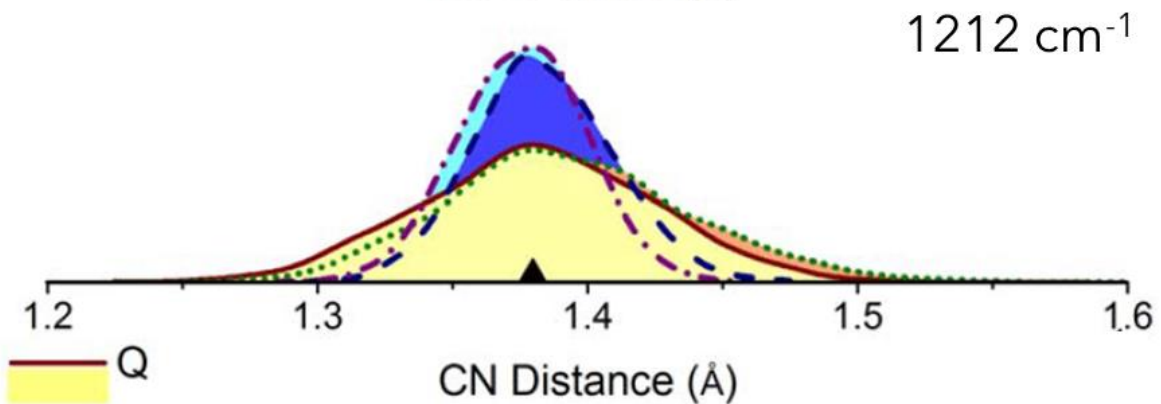
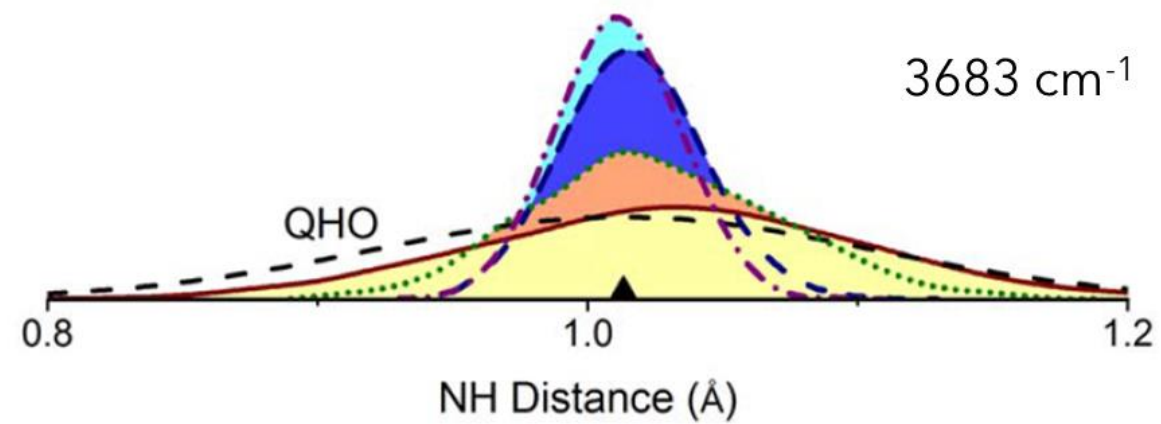
Wigner distribution  
in the ground state

Long trajectory  
in the ground state

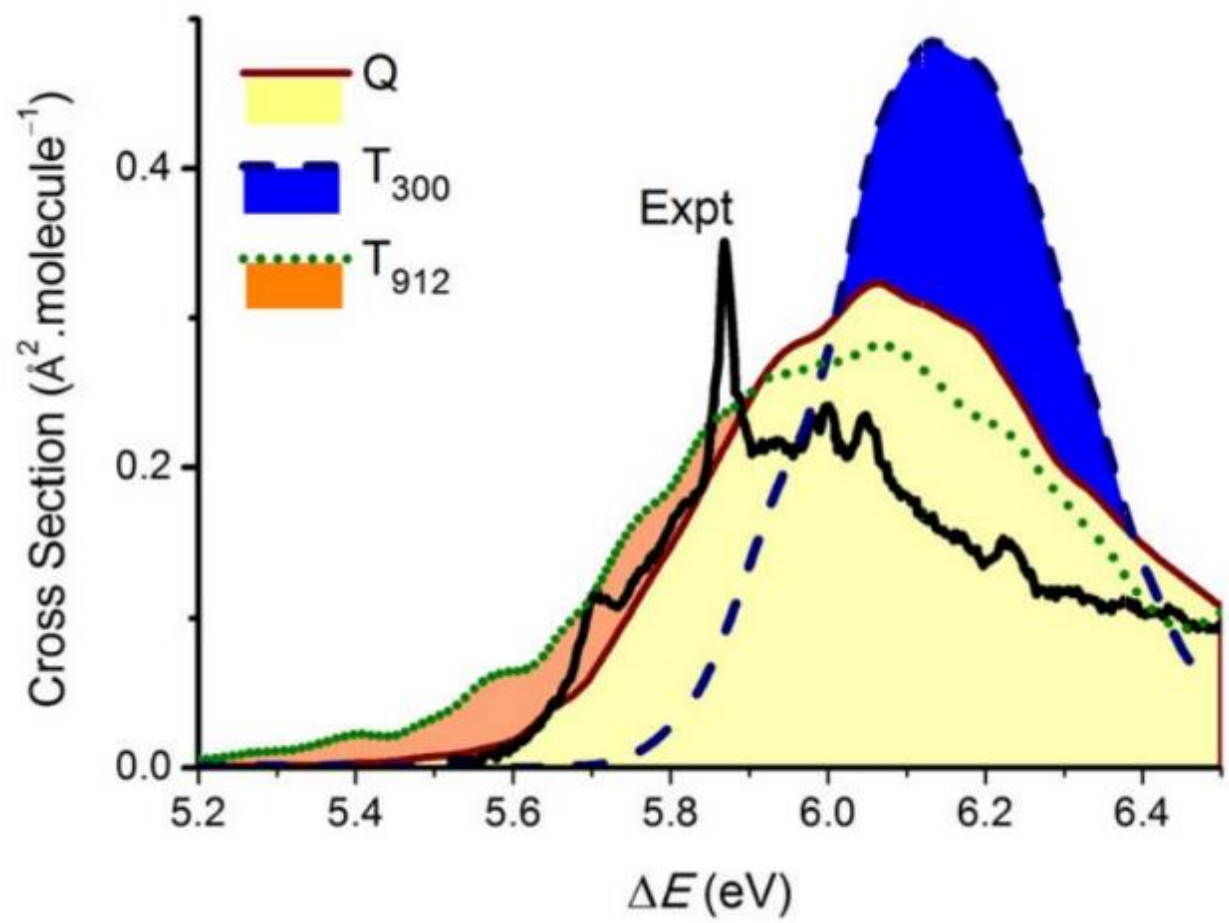
**They do NOT yield the same results**

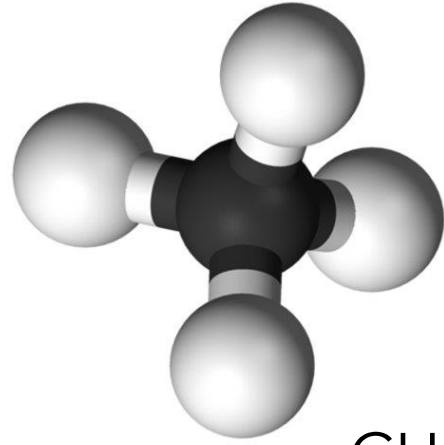
# Energy per degree of freedom





The difference is bigger in low frequency modes

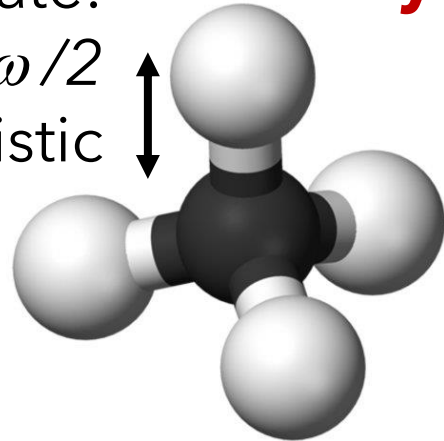




CH<sub>4</sub>...CH<sub>4</sub> (intermolecular coordinate)  
 $\varepsilon = k_B T$

CH (intramolecular) coordinate:  
 $\varepsilon = \hbar \omega / 2$

**Wigner sampling** is more realistic



**Dynamics** is more realistic

Hybrid sampling:

- Ruckenbauer; Barbatti; Müller; Lischka. *J Phys Chem A* **2013**, 117, 2790