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

en.wikipedia.org/wiki/Monte_Carlo_method

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

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

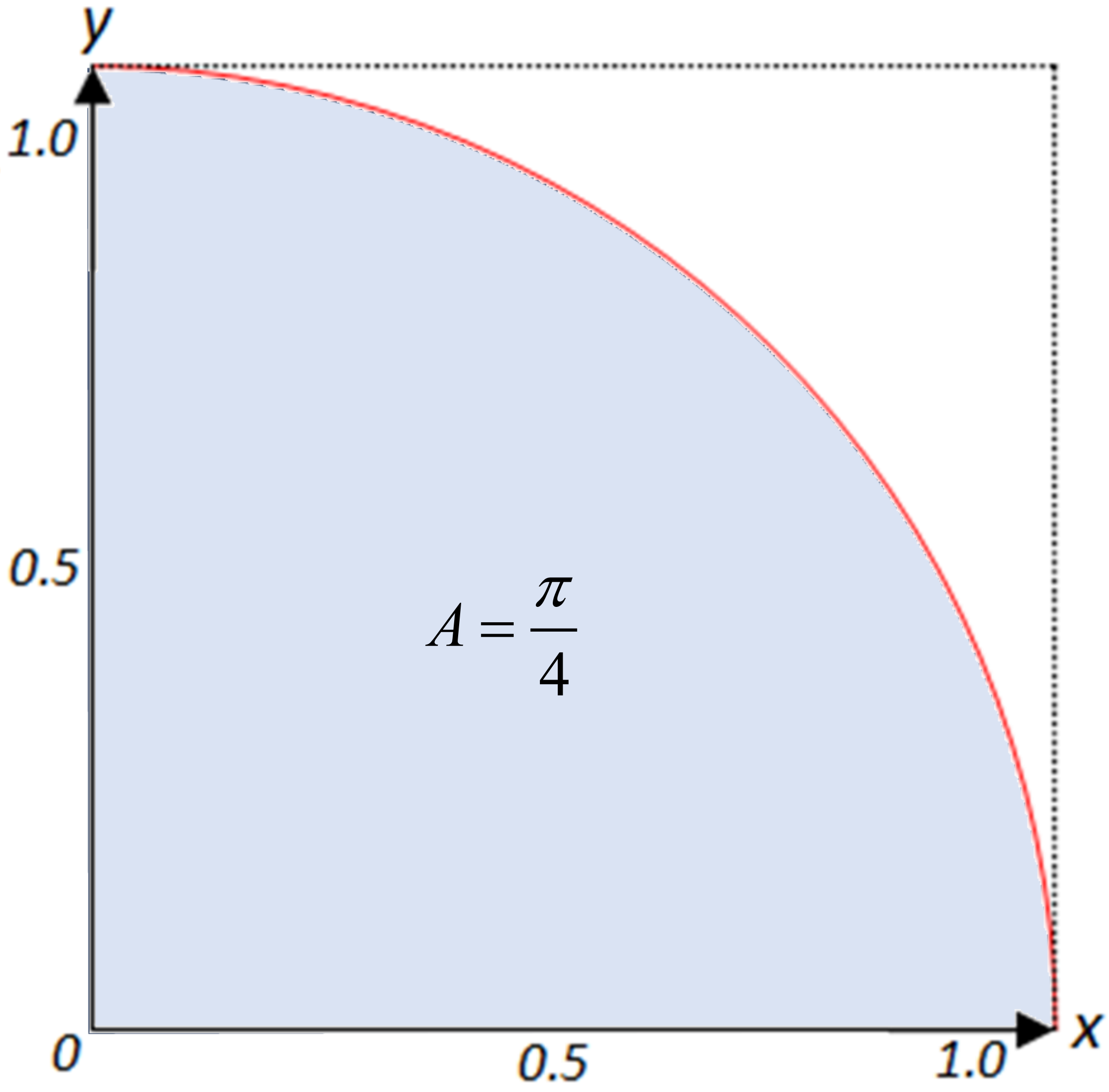
Calculate

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

The result is

$$A = \frac{1}{2}\left(x\sqrt{1 - x^2} + \arcsin\left(x\right) \right)\Big|_0^1$$

$$= \frac{\pi}{4}$$



Consider the function

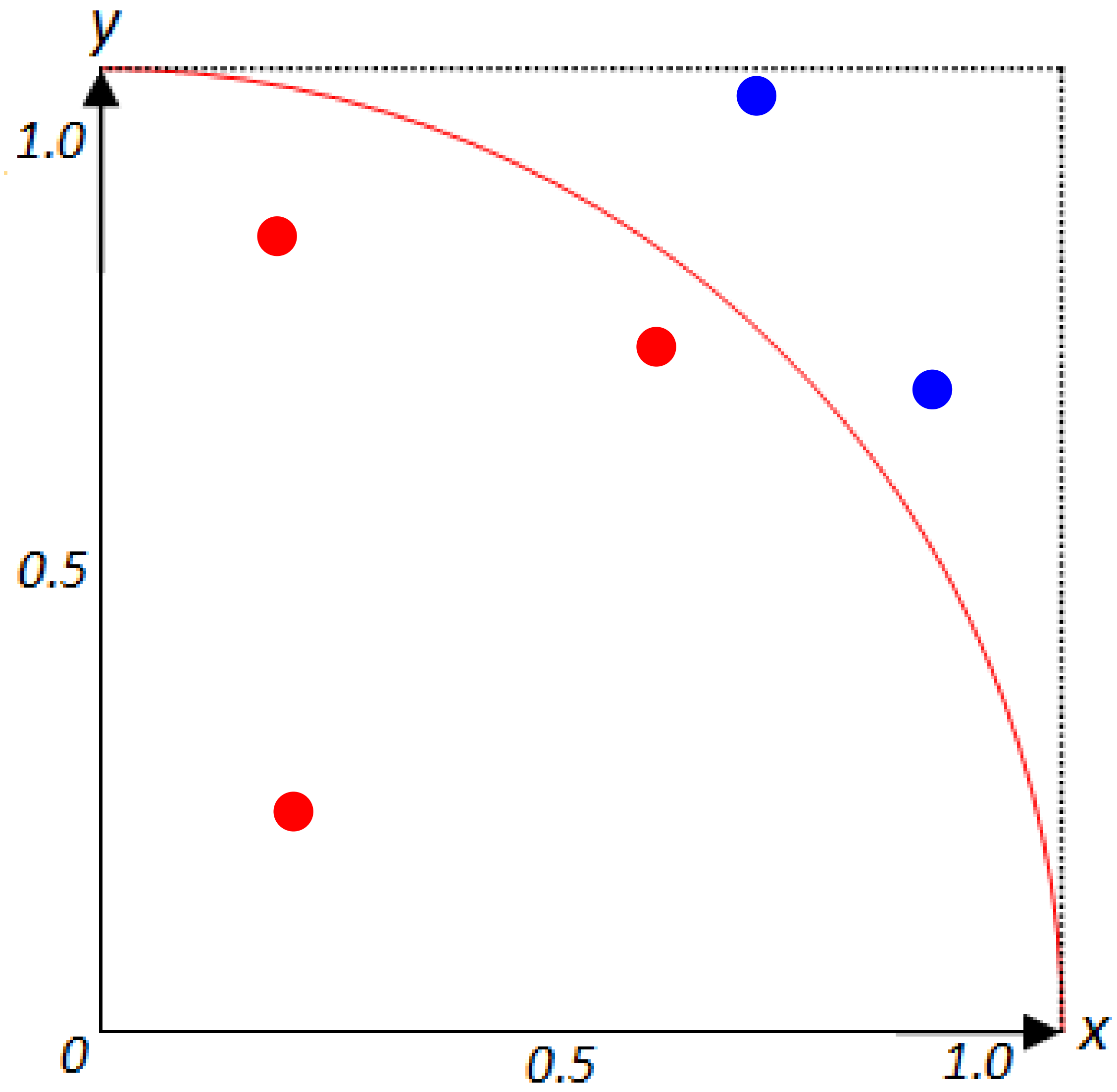
$$y = \sqrt{1-x^2}$$

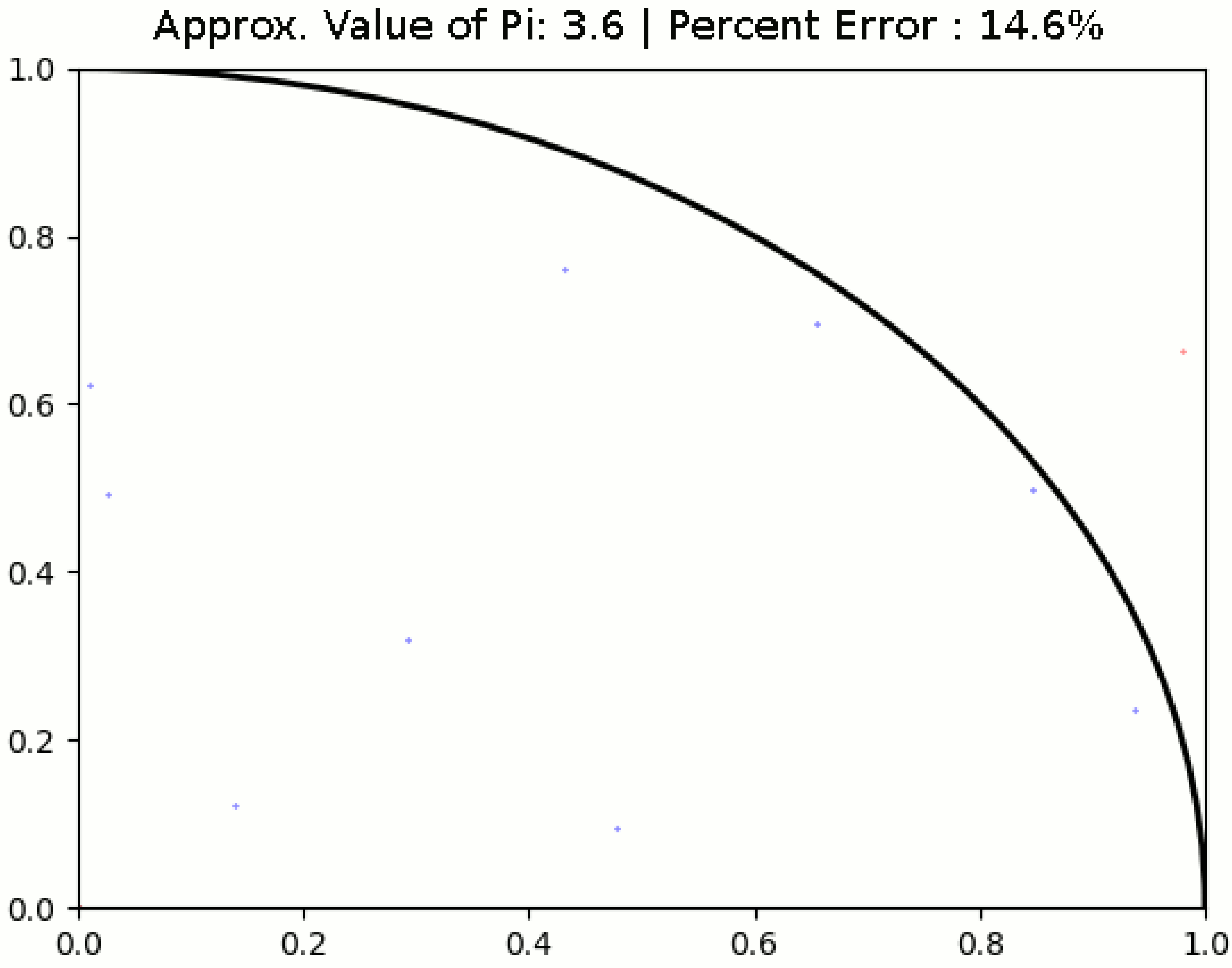
Calculate

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

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

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





github.com/IkeKap/StatsProject

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 $A_{total}.N_{accept}/N_{total}$]

en.wikipedia.org/wiki/Monte_Carlo_method

Monte Carlo integration is advantageous:

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

Luo; Sorella. *Frontiers in Materials* ***2015, 2***

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

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

where

$$\mathbf{r}^N = \left(\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_N\right)$$

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

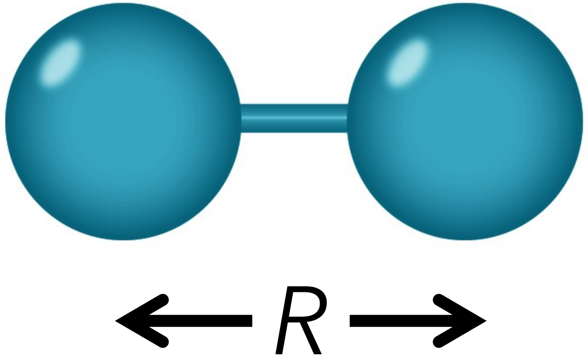
$$\mathbf{r}^N = (\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_N)$$

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

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

Luo; Sorella. *Frontiers in Materials* ***2015, 2***

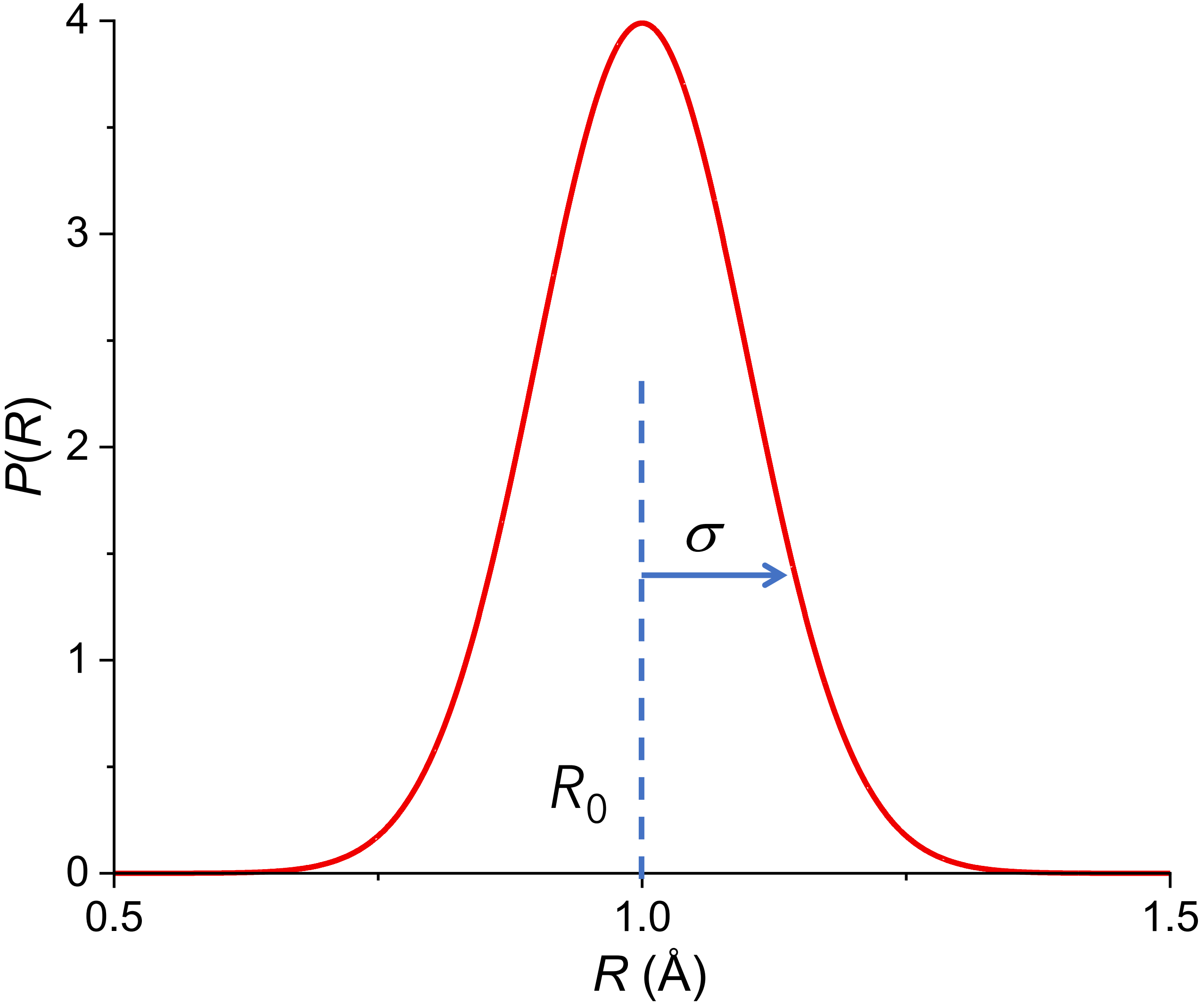
Monte Carlo sampling

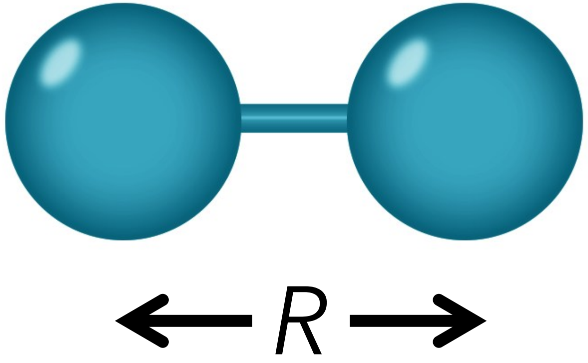


Suppose geometry is distributed as

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

R_0 = 1.0 Å
σ = 0.1 Å



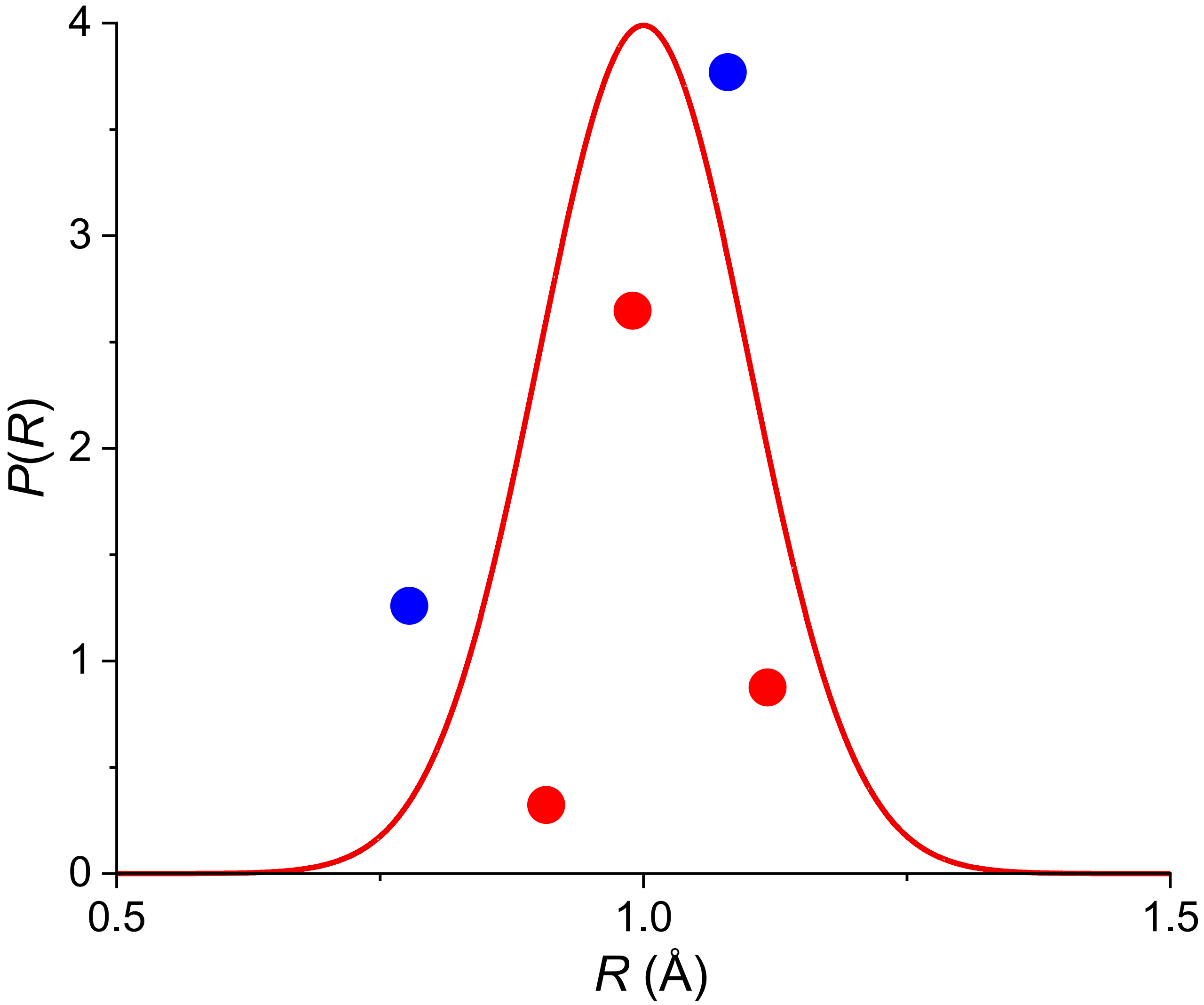


Sample random number in [x_{min}, x_{max}]

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

Compute *P*(*random x*)

- accept if *random y* < *P*(*random x*)
- reject if *random y* > *P*(*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                                                 # Mean position
sigma = 0.1                                             # Standard deviation
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

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

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

Q_i – normal mode
P_i – momentum of the normal mode
ω_i – angular harmonic frequency
μ_i – reduced mass
$N_F = 3N_{at} - 6$ – number of normal modes

Barbatti *et al. J Chem Theory Comput* ***2022,*** *18, 6851*

N_{at} molecule: Wigner distribution

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

1. For each normal mode *i*, make a Gaussian sampling of Q_i and P_i.

2. Take the set (Q_1, Q_2, ..., Q_{NF}) and transform into Cartesian coordinate **R**.

3. Take the set (P_1, P_2, ..., P_{NF}) and transform into Cartesian momentum **P**.

4. Repeat this procedure for as many points as you need.

You can use Newton-X for this procedure.

Barbatti *et al. J Chem Theory Comput* ***2022,*** *18, 6851*

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.

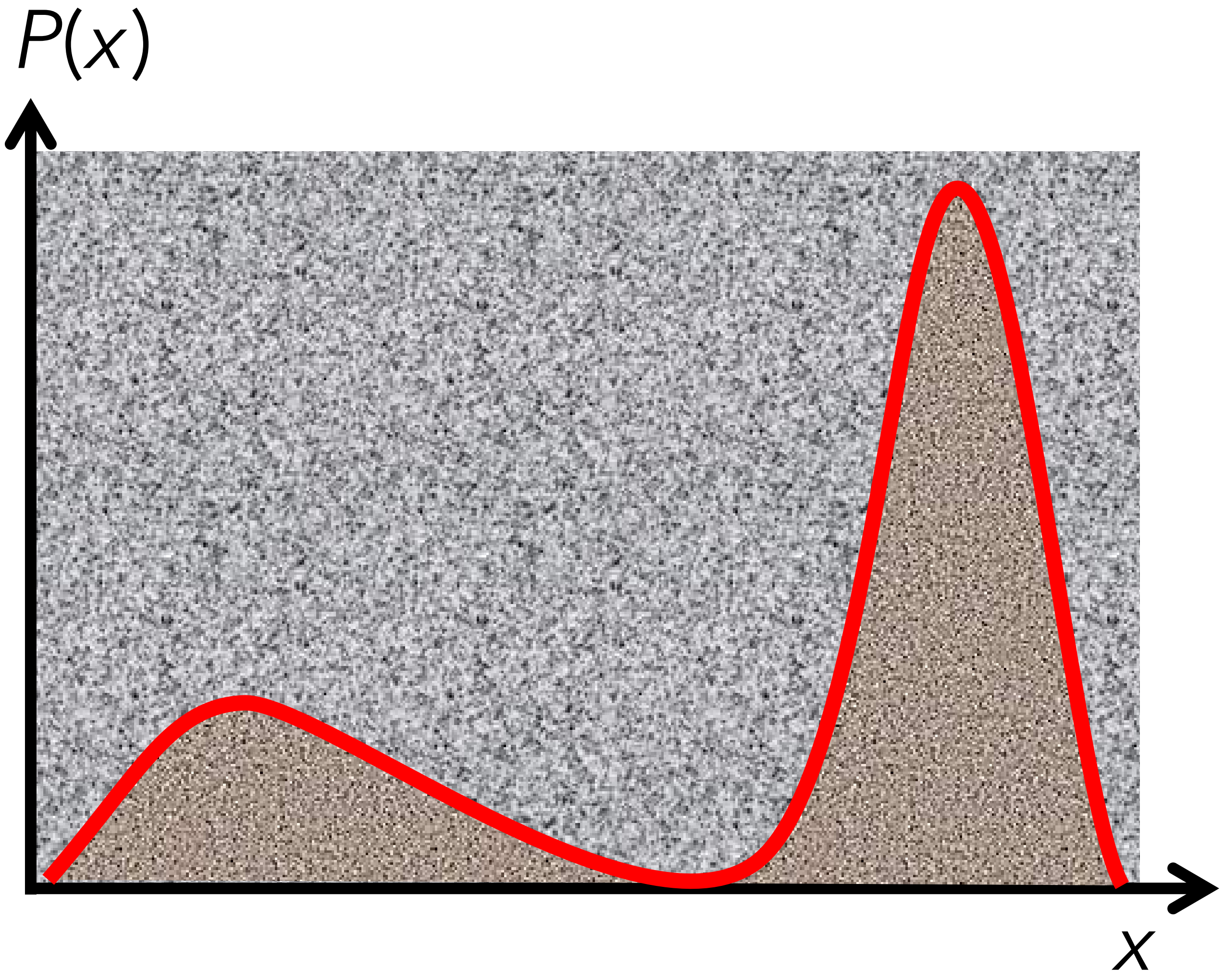
Metropolis-Hastings algorithm

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.

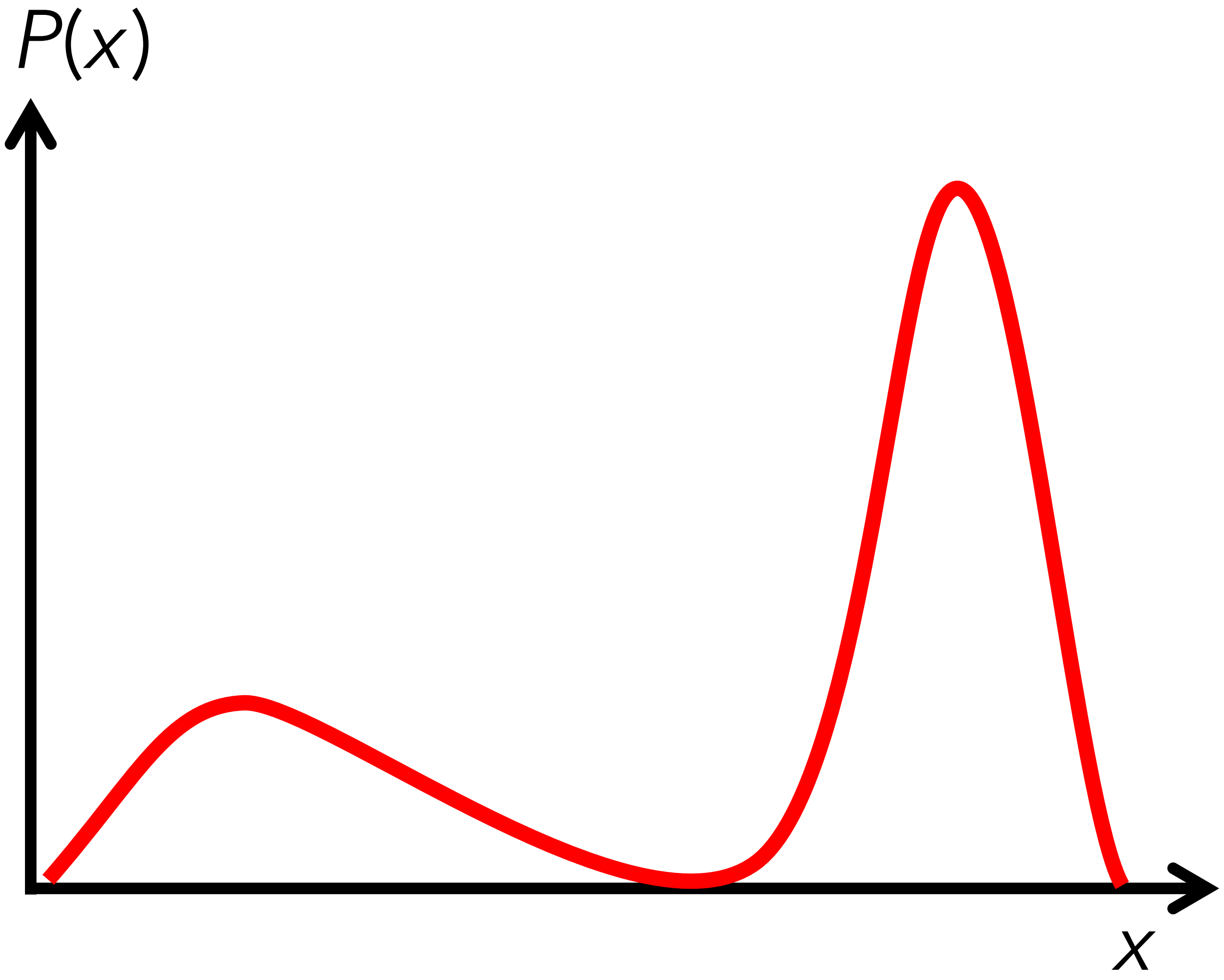


- en.wikipedia.org/wiki/Metropolis-Hastings_algorithm
- Metropolis *et al. J Chem Phys* **1953**, *21*, 1087

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.

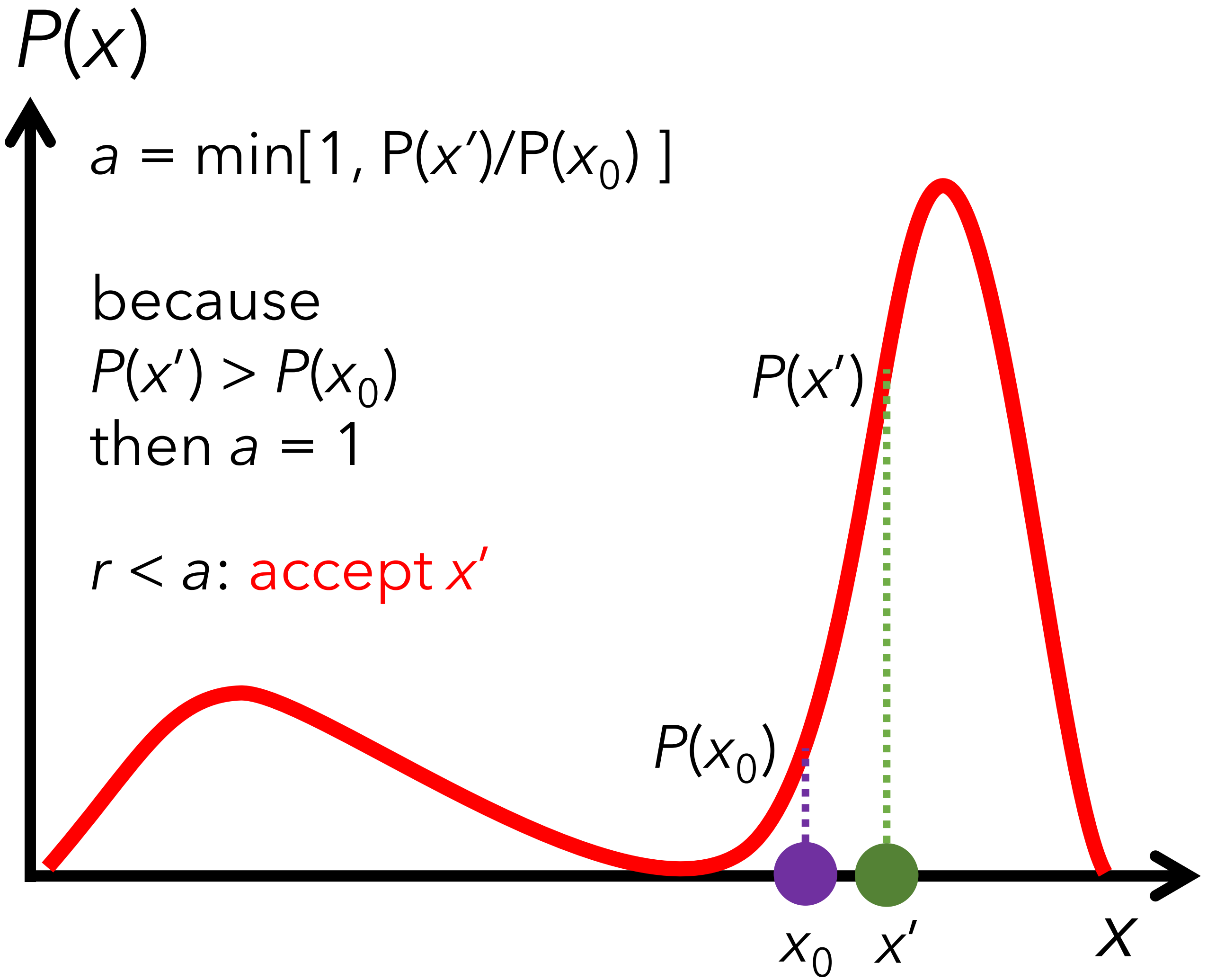
2. Sample a second random point x' using $g(x|x_0)$.

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.

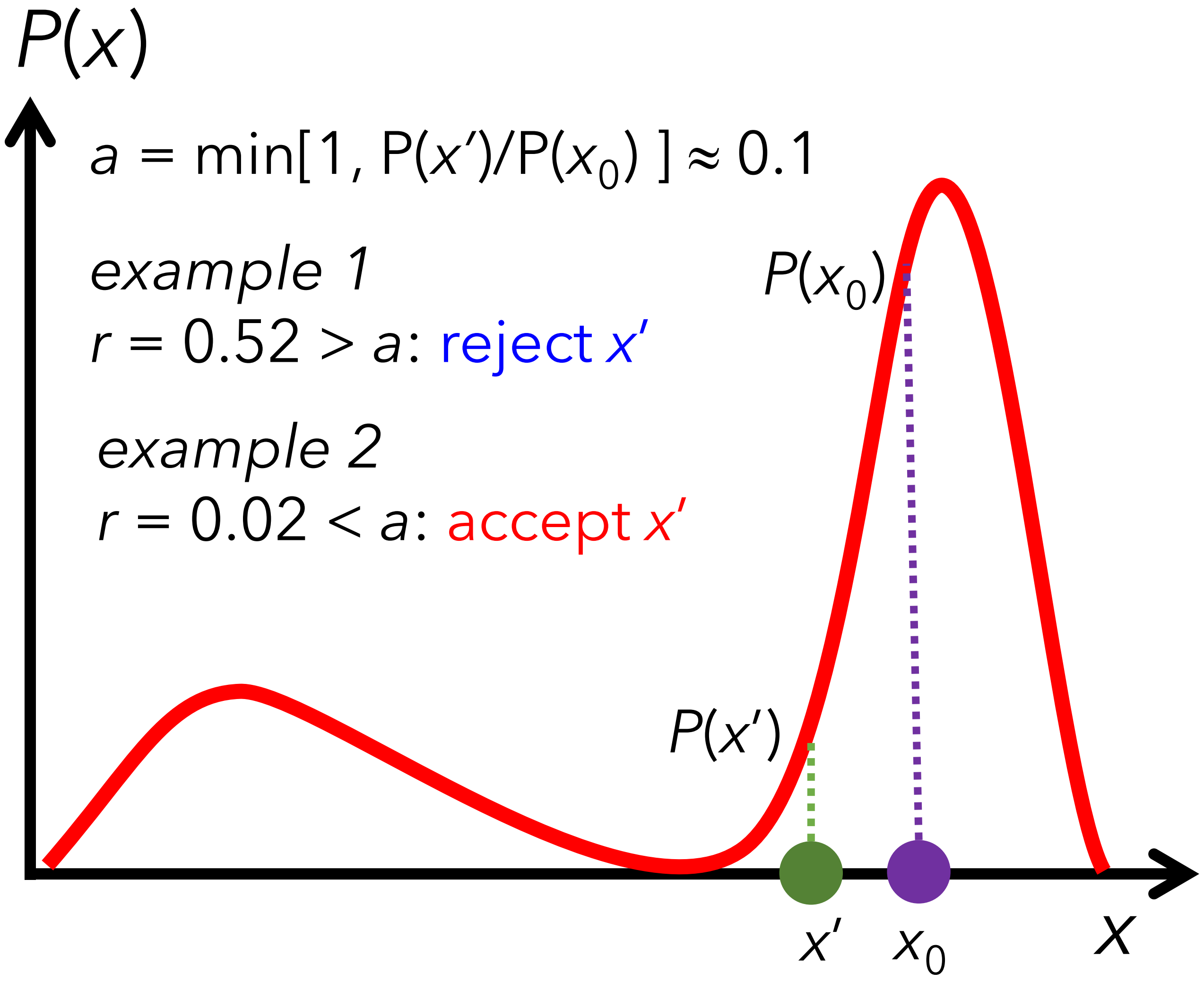


$a = \min[1, P(x')/P(x_0)\,]$

because
$P(x') > P(x_0)$
then $a = 1$

$r < a$: accept x'

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.



$a = \min[1, P(x')/P(x_0)\,] \approx 0.1$

example 1
$r = 0.52 > a$: reject x'

example 2
$r = 0.02 < a$: accept x'

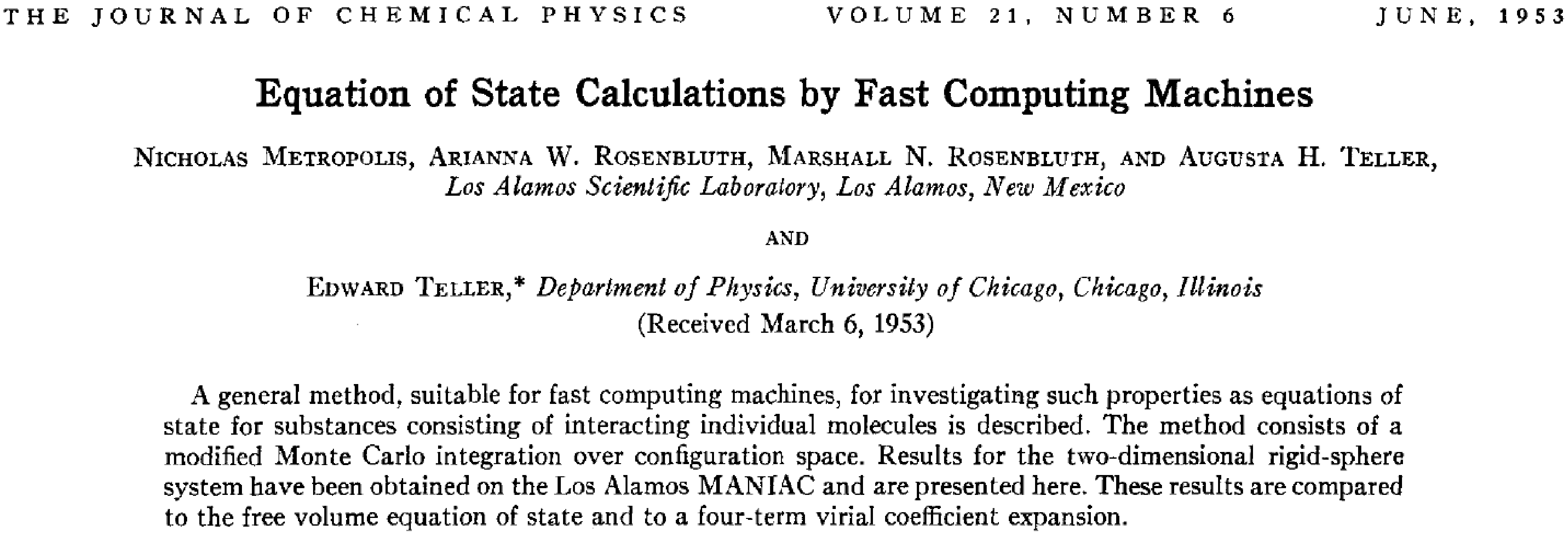
```python
def P(x, b, mu):                                                      # Define distribution (e.g., Laplace PDF)
    probability = np.exp(-abs(x - mu) / b) / (2 * b)
    return probability

(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)  # Poprosal 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

$$g\left(x \mid x_n\right) = \exp\left(-\frac{E\left(x\right) - E\left(x_{'n}\right)}{k_B T}\right)$$

If $g(x \mid x_n) = g(x_n \mid x)$
the algorithm is called Metropolis.

If $g(x \mid x_n) \neq g(x_n \mid x)$
the algorithm is called Metropolis-Hastings.



THE JOURNAL OF CHEMICAL PHYSICS VOLUME 21, NUMBER 6 JUNE, 1953

Equation of State Calculations by Fast Computing Machines

NICHOLAS METROPOLIS, ARIANNA W. ROSENBLUTH, MARSHALL N. ROSENBLUTH, AND AUGUSTA H. TELLER,
Los Alamos Scientific Laboratory, Los Alamos, New Mexico

AND

EDWARD TELLER,* *Department of Physics, University of Chicago, Chicago, Illinois*

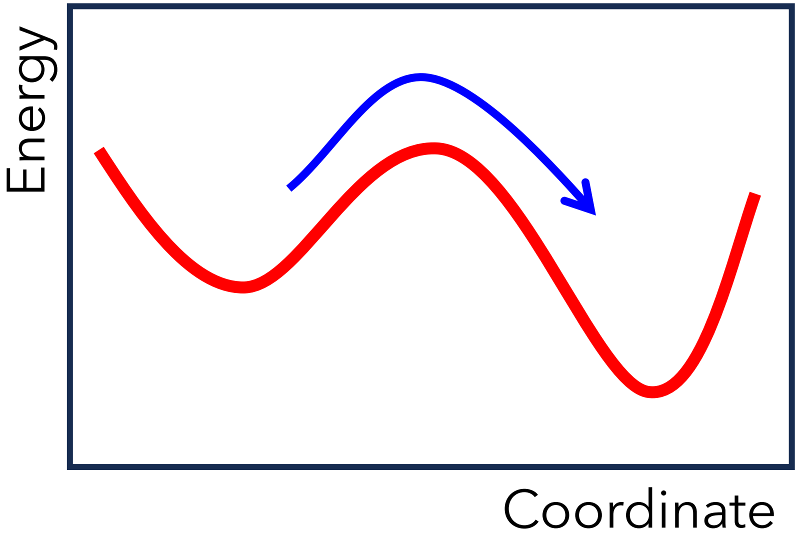
(Received March 6, 1953)

A general method, suitable for fast computing machines, for investigating such properties as equations of state for substances consisting of interacting individual molecules is described. The method consists of a modified Monte Carlo integration over configuration space. Results for the two-dimensional rigid-sphere system have been obtained on the Los Alamos MANIAC and are presented here. These results are compared to the free volume equation of state and to a four-term virial coefficient expansion.

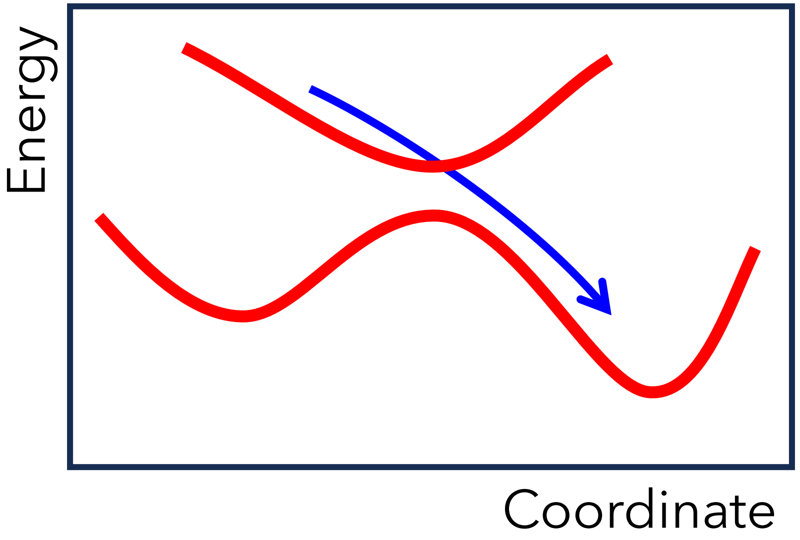
Rate theory

In diverse situations, stochastic treatment is advantageous over dynamics.

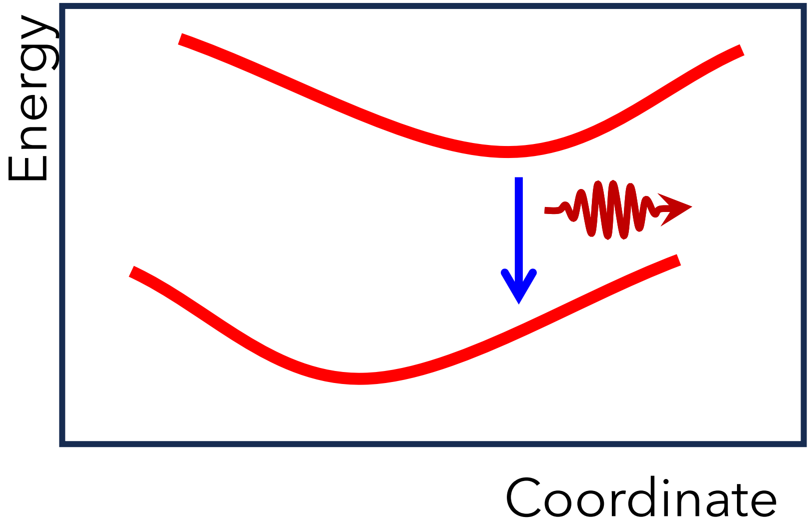
Reaction



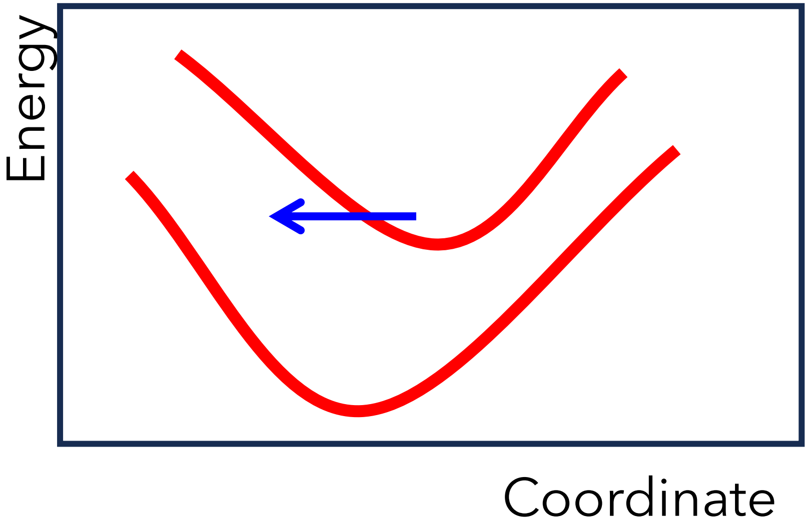
Internal conversion



Spontaneous emission



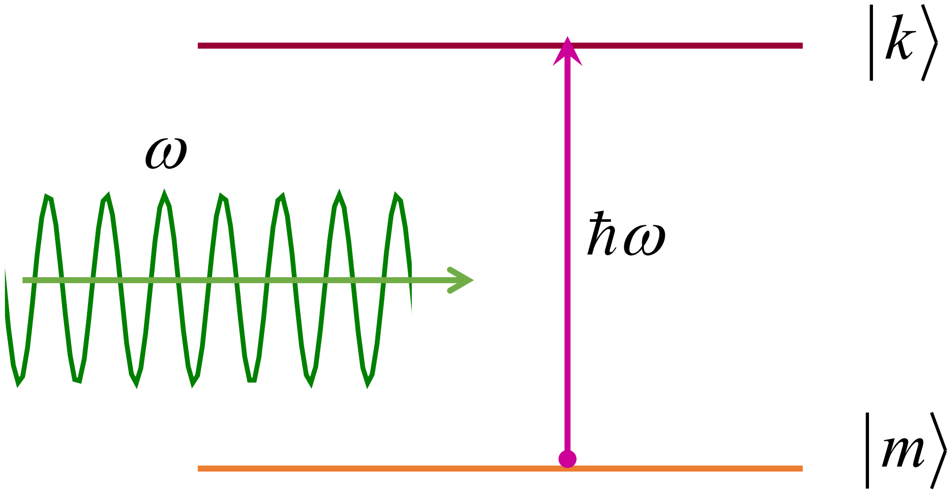
Weakly-coupled transition



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

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

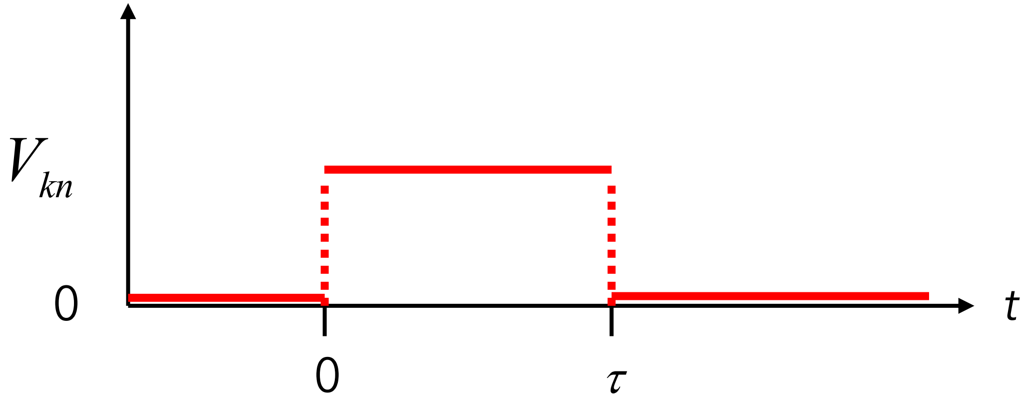
$$W_k = \frac{2\pi}{\hbar}\left|V_{mk}\right|^2 \delta\left(\Delta E_{mk} + \hbar\omega\right)$$



See derivation in the appendix to this presentation.

Fermi’s Golden Rule

Constant perturbation

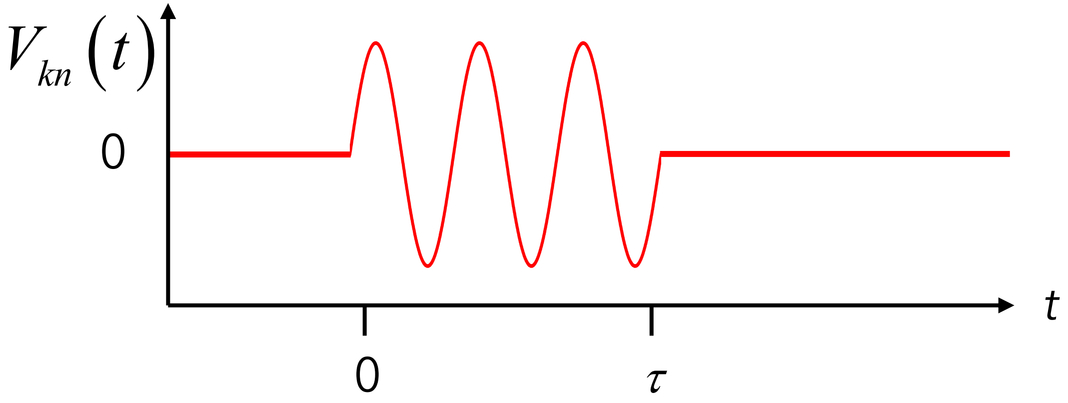


$$\boxed{W_{km} = \frac{2\pi}{\hbar}\left|V_{km}\right|^2 \delta\left(\varepsilon_k - \varepsilon_m\right)}$$

P_{mk} is max for $\varepsilon_k - \varepsilon_m = 0$

(Internal conversion, collision)

Harmonic perturbation



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

P_{mk} is max for $\hbar\omega = \mp\left(\varepsilon_m - \varepsilon_k\right)$

(photon absorption, emission)

The rate has units of time inverse

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

$$\left[W \right] = \frac{1}{\left[Et \right]} \left[E \right]^2 \times \frac{1}{\left[E \right]}$$

$$= \frac{1}{\left[t \right]}$$

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

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

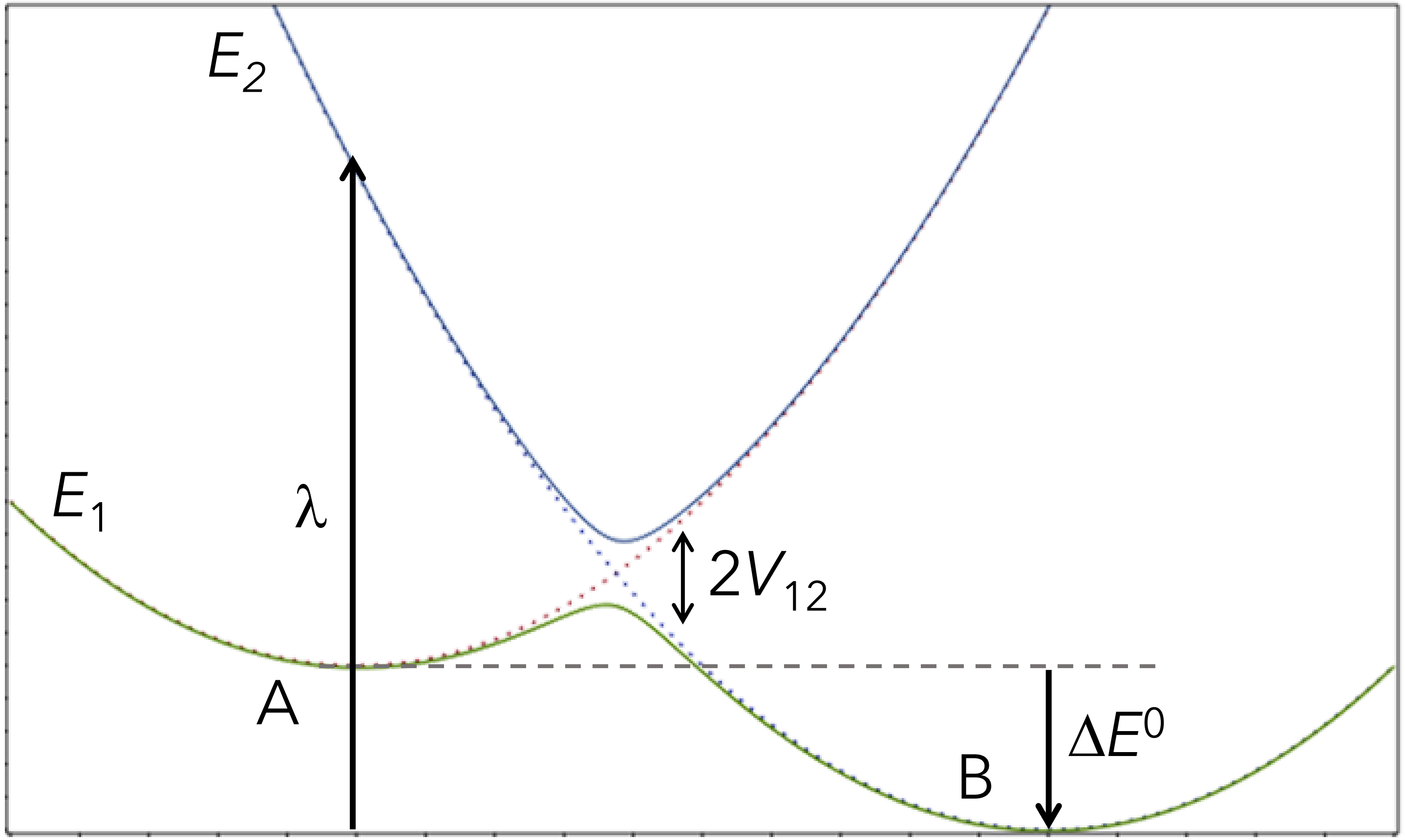
Fermi’s golden rule is used to derive:

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

Marcus rate

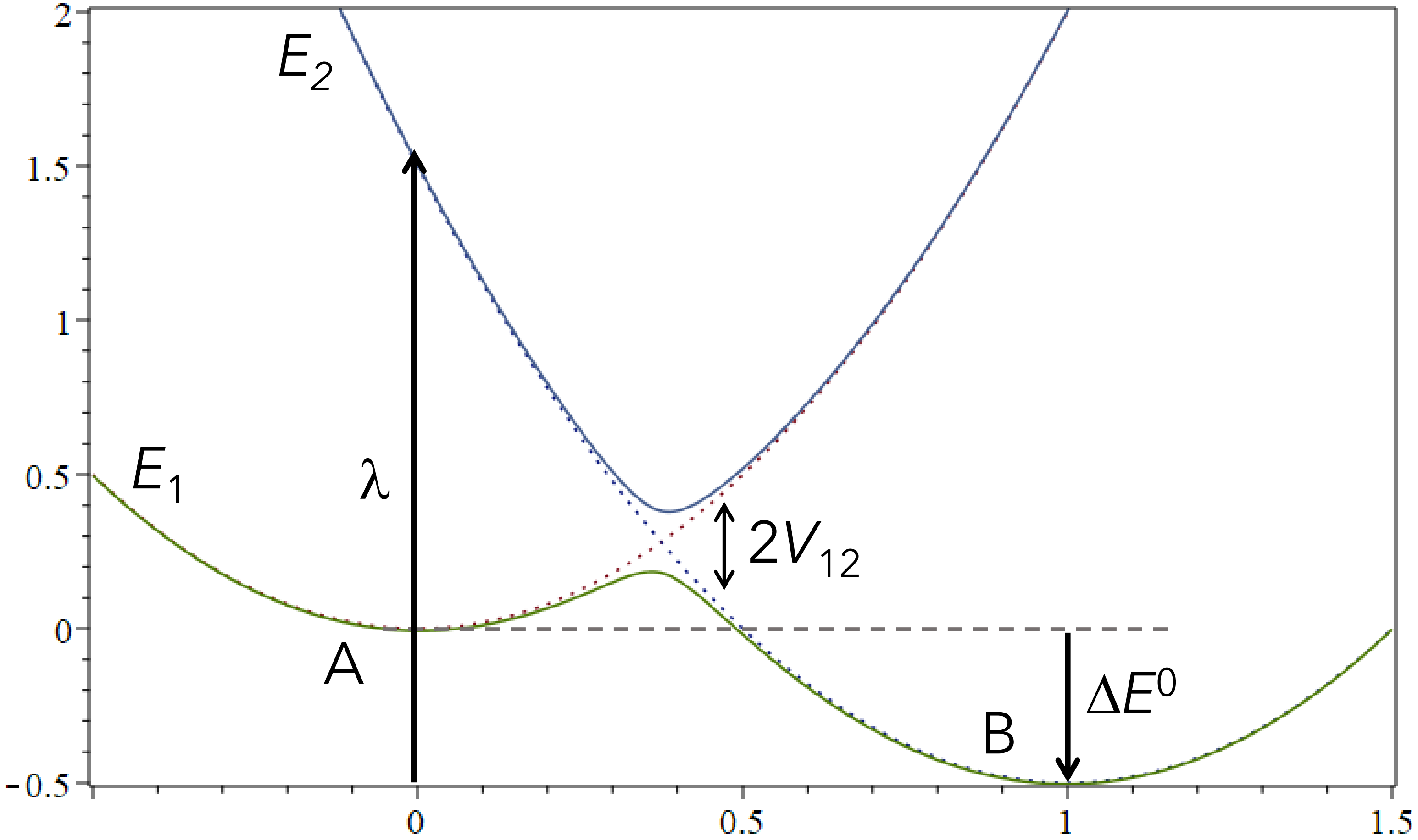
Marcus rate

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



Brédas *et al. Chem Rev* **2004**, *104*, 4971

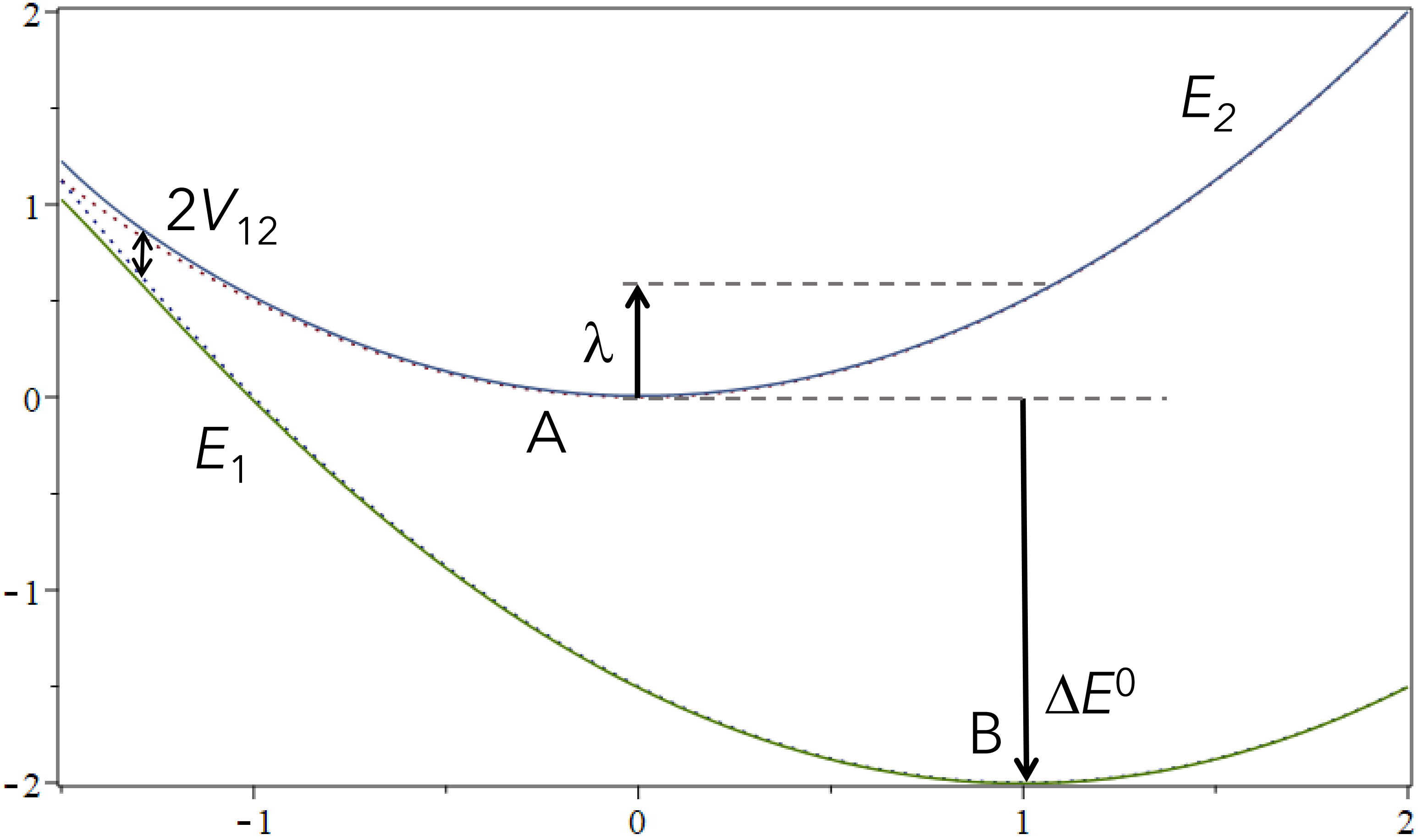
$$W_{AB}^{(Marcus)} = \frac{2\pi}{\hbar} |V_{12}|^2 \frac{1}{\sqrt{4\pi\lambda k_B T}} \exp\left(-\frac{(\lambda + \Delta E^0)^2}{4\lambda k_B T} \right)$$



- Adiabatic energy gap
  ΔE^0 = -0.5 eV
- Reorganization energy
  λ = 1.5 eV
- Diabatic coupling
  V_{12} = 0.1 eV

Rate (300 K):
$W_{A\rightarrow B} = 2.2\times10^{11}\ s^{-1}$
Lifetime:
$\tau_{A\rightarrow B}$ = 0.004 ns

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



- 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):

$W_{A\rightarrow B} = 3\times10^{-5}\ s^{-1}$

Lifetime:

$\tau_{A\rightarrow B} = 33{,}472$ s (9 h)

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 cm^{-1} at 300 K)

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

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

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

S_c is implemented in MOMAP

Bixon; Jortner, In *Election transfer-from isolated molecules to biomolecules*, ***1999***
Bozzi; Rocha. *J Chem Theory Comput* **2023**, *19*, 2316

Spontaneous emission rate

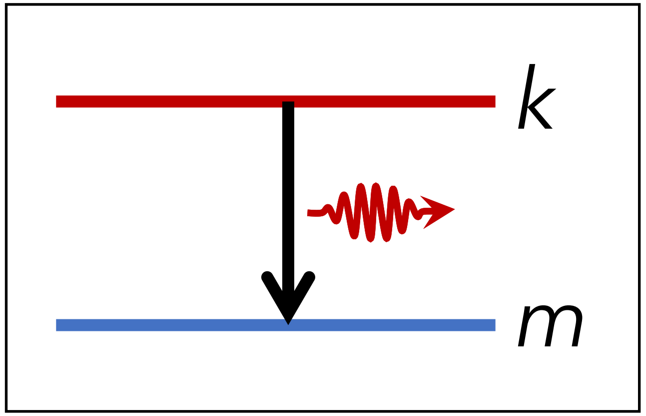
Spontaneous emission rate

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

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

$\alpha = \dfrac{e^2}{4\pi\varepsilon_0 \hbar c} \approx \dfrac{1}{137}$ is the fine structure constant

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

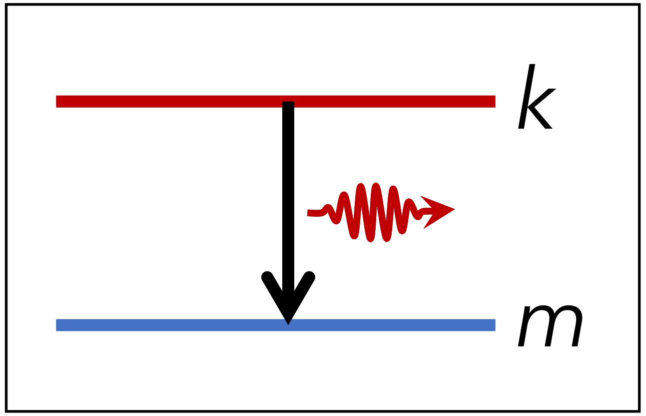


$$\boxed{W_{km}^{(Sp.Em.)} = \frac{1}{\tau_{km}^{(S.Em.)}} = \frac{2\alpha}{\hbar m_e c^2}\Delta E_{km}^2 f_{km}}$$

Spontaneous emission rate

$$\boxed{W_{km} = \frac{1}{\tau_{km}} = \frac{2\alpha}{\hbar m_e c^2} \Delta E_{km}^2 f_{km}}$$

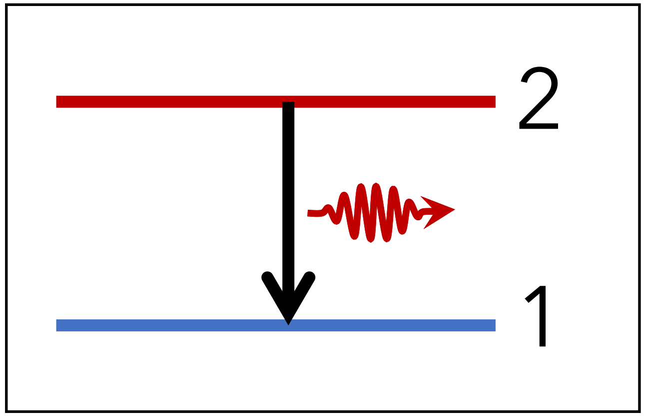
$\alpha = \frac{e^2}{4\pi\varepsilon_0 \hbar c} \approx \frac{1}{137}$ is the fine structure constant



Better to work in atomic units: $\begin{cases} m_e = \hbar = e = \frac{1}{4\pi\varepsilon_0} = 1 \\ \alpha = 1/c \end{cases}$

$$\boxed{W_{km}^{(au)} = \frac{1}{\tau_{km}^{(au)}} = \frac{2}{c^3} \Delta E_{km}^2 f_{km}}$$

$$W_{km}^{(au)} = \frac{1}{\tau_{km}^{(au)}} = \frac{2}{c^3} \Delta E_{km}^2 f_{km}$$



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

$$f_{21} \quad = 1.0$$

c = 137

Hartree-to-eV = 27.2114

au_time-to-seconds = 0.2419×10^{-16}

$$W_{21} = 1.7 \times 10^8 \text{ s}^{-1}$$

$$\tau_{21} \quad = 5.8 \times 10^{-9} \text{ s}$$

To know more:

Monte Carlo & Metropolis-Hastings
en.wikipedia.org/wiki/Monte_Carlo_method
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 (Ask me for the password)

Appendix
Derivation of the Fermi's Golden Rule

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

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

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

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

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

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

Left side

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

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

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

Right side

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

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

expand

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

replace

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

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

project

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

$$\sum_n c_n(t) \varepsilon_n \cancel{e}^{-i\varepsilon_n t/\hbar} \left\langle\phi_k | \phi_n\right\rangle + \sum_n c_n(t) e^{-i\varepsilon_n t/\hbar} \left\langle\phi_k \middle| V(t) \middle| \phi_n\right\rangle$$

$$\left\langle\phi_k | \phi_n\right\rangle = \delta_{kn} \qquad \left\langle\phi_k \middle| V(t) \middle| \phi_n\right\rangle = V_{kn}(t)$$

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

expand

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

replace

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

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

project

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

First-order perturbation solution

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

$$\frac{dc_k(t)}{dt} \approx -\frac{i}{\hbar}\sum_n c_n(0) e^{-i\omega_{nk}t} V_{kn}(t)$$

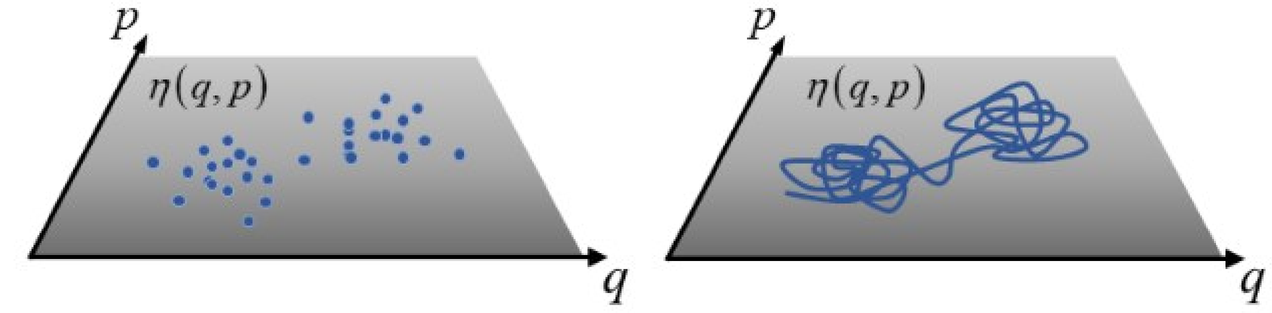
$$= -\frac{i}{\hbar}\sum_n \delta_{nm} e^{-i\omega_{nk}t} V_{kn}(t) = -\frac{i}{\hbar} e^{-i\omega_{mk}t} V_{km}(t)$$

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

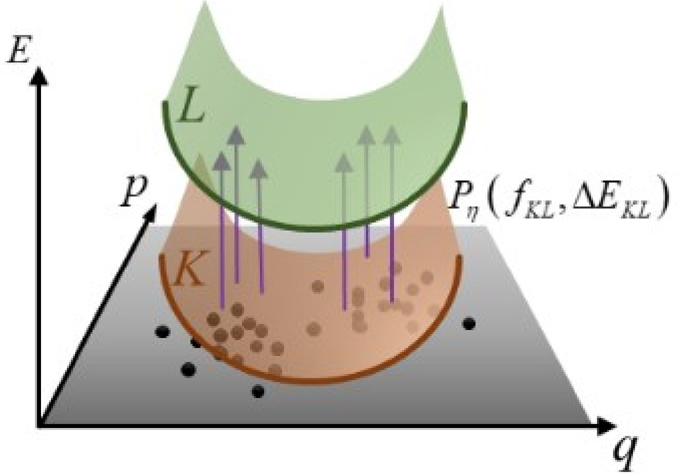
$$P_k(t) = \left|c_k(t)\right|^2 = \frac{1}{\hbar^2}\left|\int_0^t e^{-i\omega_{mk}t} V_{km}(t')dt'\right|^2$$

Appendix
Wigner x trajectory samplings

1. Sampling

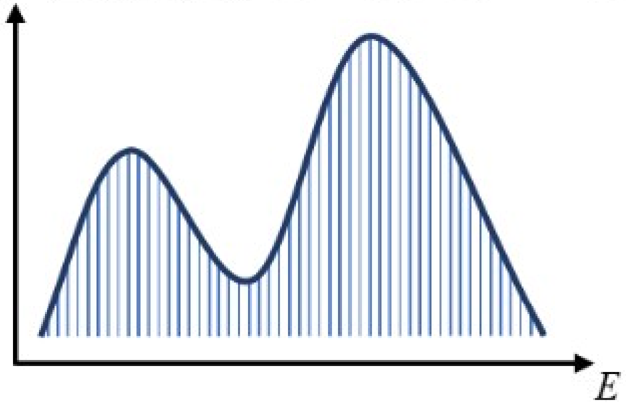


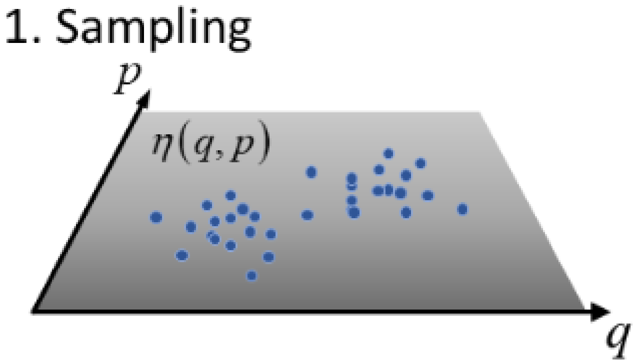
2. Transition probabilities



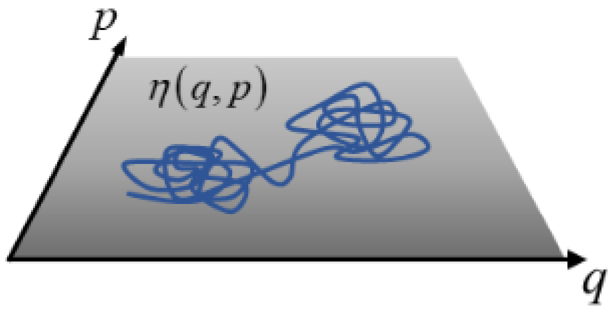
3. Spectrum

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





Wigner distribution
in the ground state

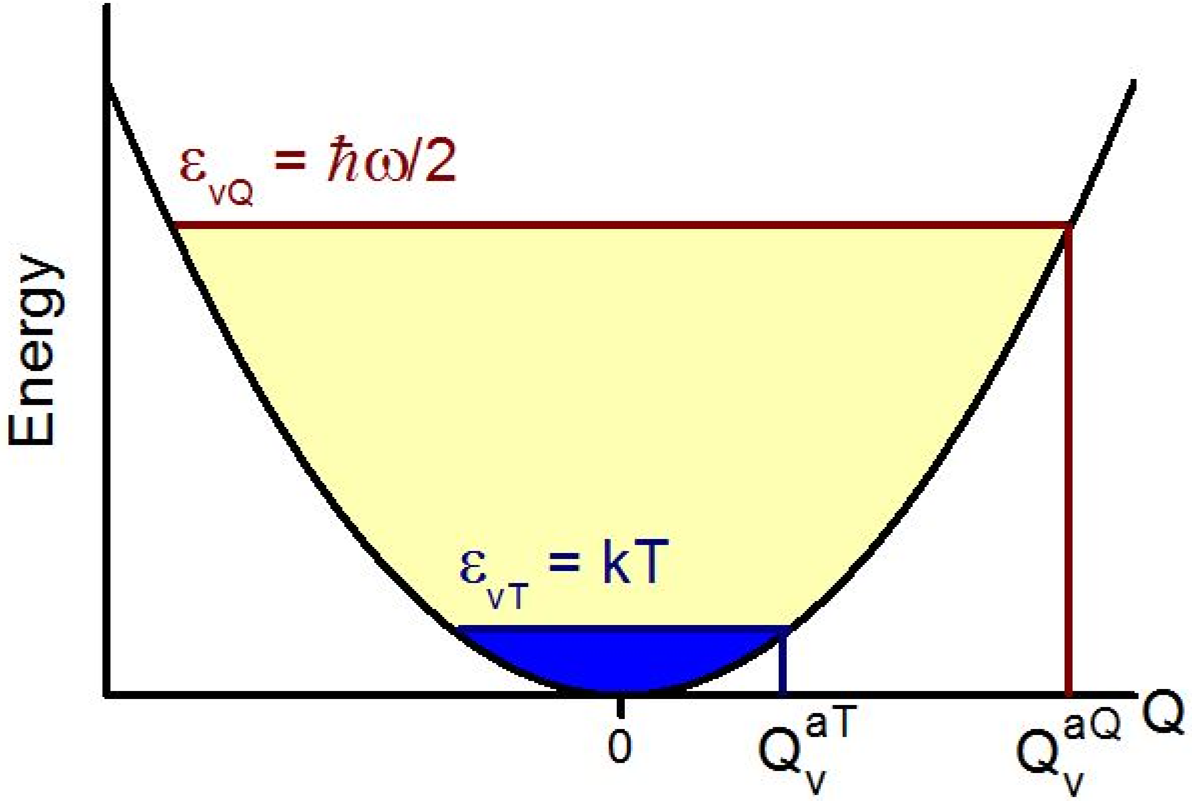


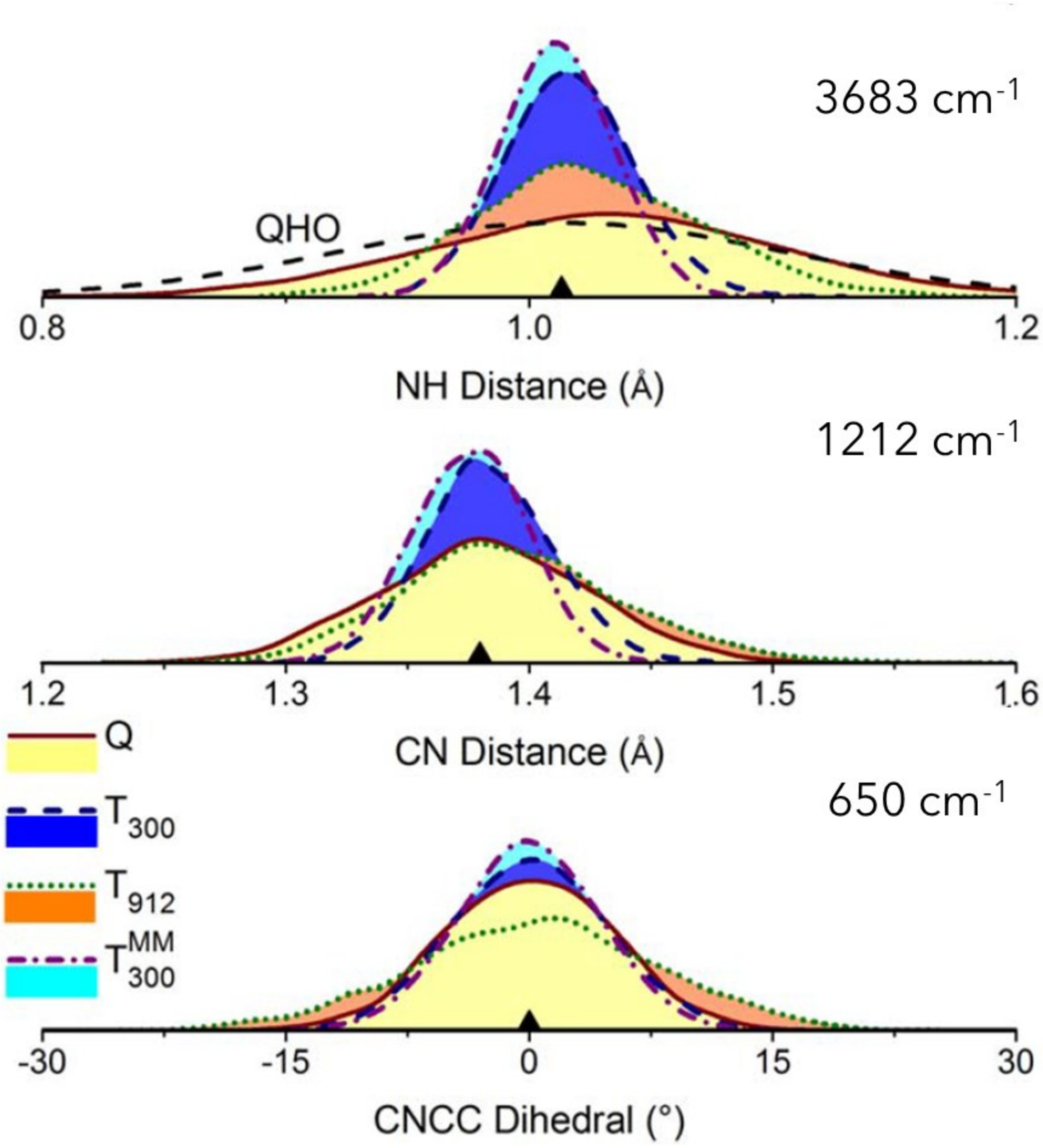
Long trajectory
in the ground state

They do NOT yield the same results

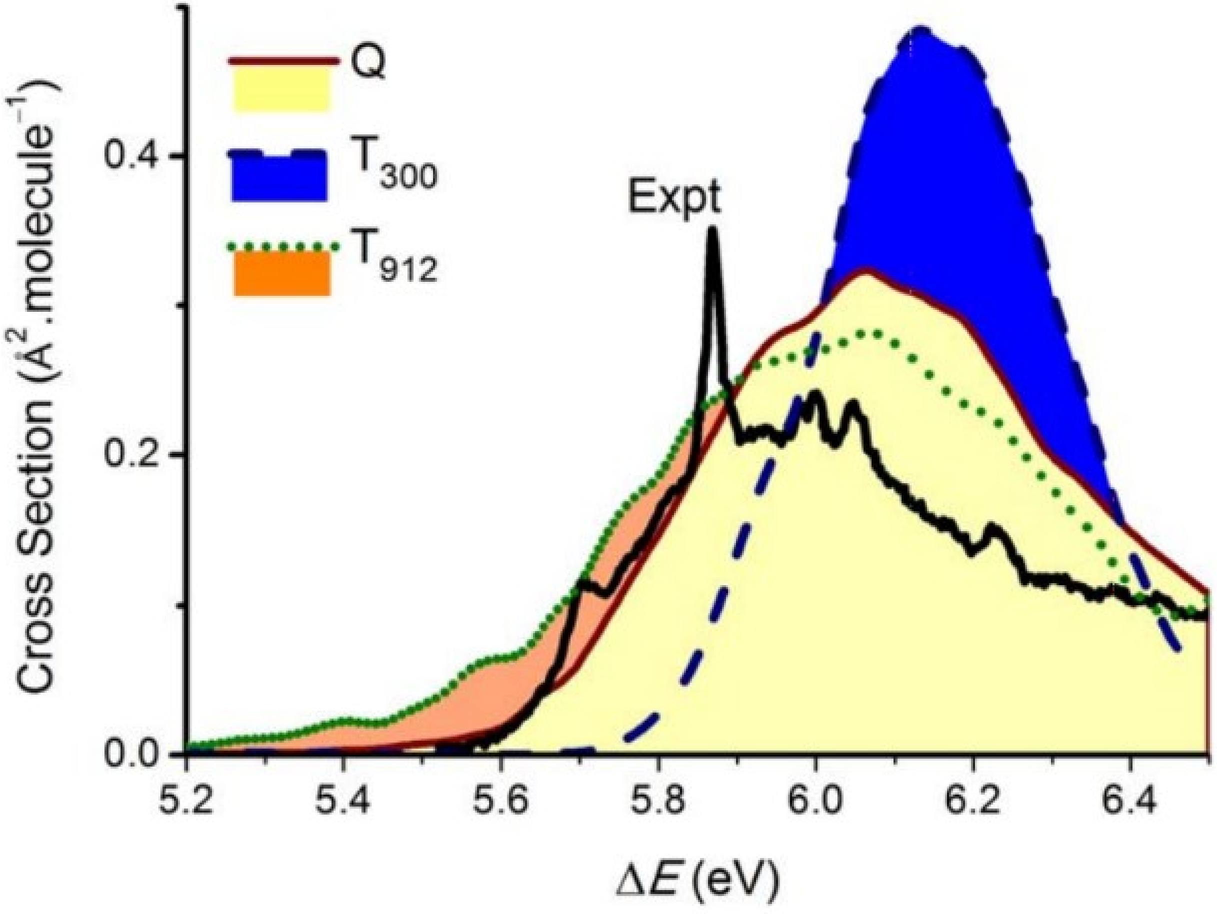
Barbatti; Sen. *Int J Quantum Chem* **2016,** *116*, *762*

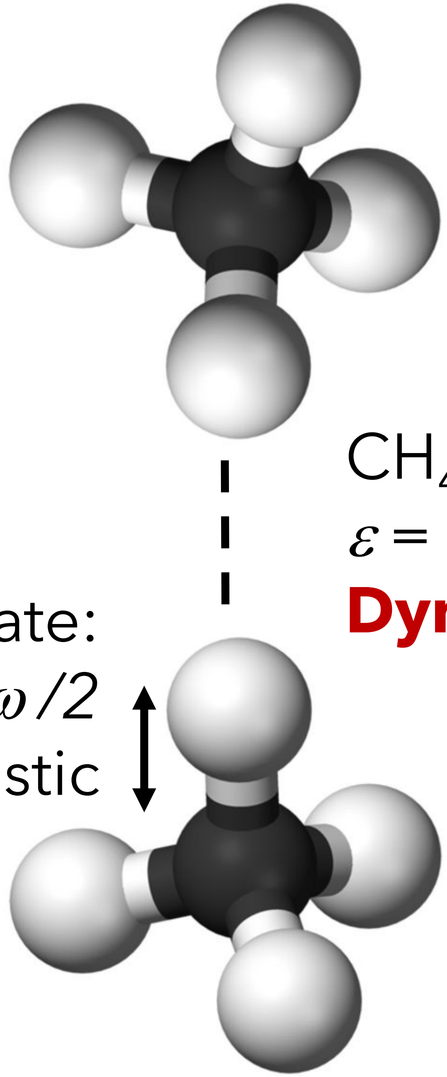
Energy per degree of freedom





The difference is bigger in low frequency modes


Q
T$_{300}$
T$_{912}$
Expt
Cross Section (Å^2.molecule^{-1})
0.0
0.2
0.4
5.2
5.4
5.6
5.8
6.0
6.2
6.4
ΔE (eV)



$CH_4 \ldots CH_4$ (intermolecular coordinate)
$\varepsilon = k_B T$
Dynamics is more realistic

CH (intramolecular) coordinate:
$\varepsilon = \hbar\omega/2$
Wigner sampling is more realistic

Hybrid sampling:

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