Introduction to Computational Physics

Jakub Rydzewski

NCU Institute of Physics

Updated: April 22, 2021

Course Information

- Every Thursday at 14.15, online only
- GitHub link: `https://github.com/jakryd/0800-fizobl/`
- Slack group: ncu-students/0800-fizobl-21
- Zoom meeting ID: 208 317 2371
- Exams take place in June 2021

Outline

Part 1. Statistical Mechanics

- Probability
- Phase space
- Thermodynamic equilibrium
- Statistical ensembles

Part 2. Monte Carlo Methods

- Sampling probability distributions
- Importance sampling

Part 3. Molecular Dynamics

- Verlet integrator
- Force and energy

Outline

Part 4. Enhanced Sampling

- Rare events
- Collective variables
- Free energy

Part 5. Nonequilibrium Statistical Physics

Part 6. Machine Learning

- Connection between machine learning and statistical physics
- Unsupervised learning

Literature

- M. E. Tuckerman, *Statistical Mechanics: Theory and Simulation*, Oxford University Press (2016).
- D. Chandler, *Introduction to Modern Statistical Mechanics*, Oxford University Press (1987).
- R. K. Patria, P. D. Beale, *Statistical Mechanics*, Elsevier (2011).
- M. Toda, R. Kubo, N. Saito, *Statistical Physics I: Equilibrium Statistical Mechanics*, Springer-Verlag (1983).
- R. Kubo, M. Toda, N. Hashitsume, *Statistical Physics II: Nonequilibrium Statistical Mechanics*, Springer-Verlag (1991).

Statistical Mechanics

Probability

- Probability is the language of statistical mechanics.
- Fundamental to the understanding of quantum mechanics.
- The large number of degrees of freedom of a macroscopic system make it necessary to use statistics.

Probability

Random Variable

A random variable $\boldsymbol{X}$ is completely defined by the range of values it can take, and its probability distribution $p_{\boldsymbol{X}}(x_1, \ldots, x_k)$. The value $p_{\boldsymbol{X}}$ is the probability that the random variable $\boldsymbol{X}$ takes the value $\mathbf{x} = (x_1, \ldots, x_k)$.

- Note that $p_{\boldsymbol{X}}(\mathbf{x}) \equiv \mathbb{P}[\boldsymbol{X} = \mathbf{x}]$.
- From now on we write $p(\mathbf{x})$ to denote $p_{\mathbf{X}}(\mathbf{x})$.
- $p_{\boldsymbol{X}}(\mathbf{x})$ is non-negative and satisfies the normalization condition:

$$\int \mathrm{d}\mathbf{x}\, p_{\boldsymbol{X}}(\mathbf{x}) = 1. \tag{1}$$

- The expectation value of $f(\boldsymbol{X})$ (or average) is denoted by:

$$\mathbb{E}[f] = \int \mathrm{d}\mathbf{x}\, p_{\boldsymbol{X}}(\mathbf{x}) f(\mathbf{x}). \tag{2}$$

Probability

Gaussian Random Variable $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

A continuous variable $\boldsymbol{X} \in \mathbb{R}^k$ has a Gaussian distribution of mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}^2$ if its probability density is:

$$p(\mathbf{x}) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^k \det(\boldsymbol{\Sigma})}}. \tag{3}$$

We have $\mathbb{E}[\boldsymbol{X}] = \boldsymbol{\mu}$ and $\mathrm{Var}[\boldsymbol{X}] = \Sigma$.

Probability

- The *entropy* of a random variable $\boldsymbol{X}$ with probability distribution $p(\mathbf{x})$ is defined as:

$$H_{\boldsymbol{X}} \equiv -\int \mathrm{d}\mathbf{x}\, p(\mathbf{x}) \log p(\mathbf{x}), \tag{4}$$

  where we define $0 \log 0 = 0$.[1]

- Entropy $H_{\boldsymbol{X}}$ is a measure of uncertainty of the random variable $\boldsymbol{X}$.

[1] Units for $\log_2$: bits and $\log_e$: nats

Probability

Examples

A fair coin has two values with equal probability. Its entropy is 1 bit.

Consider throwing M fair coins: the number of all possible outcomes is 2^M. The entropy equals M bits.

Bernoulli process. A Bernoulli random variable X can take two values 0, 1 with probabilities $p(0) = q$ and $p(1) = 1 - q$. Its entropy is:

$$H_X = -q \log q - (1 - q) \log(1 - q). \tag{5}$$

The entropy vanishes for $q = 0$ or $q = 1$ because the outcome is certain.

Probability

- The Kullback-Leibler (KL) divergence (or the relative entropy) measures a stiatistical distance between $p(x)$ and $q(x)$. It is defined as:

$$D_{\mathrm{KL}}(q\|p) \equiv \int \mathrm{d}x\; q(x) \log \frac{q(x)}{p(x)}. \tag{6}$$

- $D_{\mathrm{KL}}(q\|p)$ is convex in $q(x)$,
- $D_{\mathrm{KL}}(q\|p) \geq 0$,
- and $D_{\mathrm{KL}}(q\|p) = 0$ if $q(x) = p(x)$.
- Not symmteric: $D_{\mathrm{KL}}(q\|p) \neq D_{\mathrm{KL}}(p\|q)$.

Dirac δ function

- $\delta(x) = 0$ if $x \neq 0$ and $\delta(x) \to \infty$ (undefined) if $x = 0$
- $\int_{-\epsilon}^{\epsilon} \mathrm{d}x\ \delta(x) = 1$ for all $\epsilon > 0$
- Gaussian model:

$$\delta_\sigma(x) = \lim_{\sigma \to 0} \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-\frac{x^2}{2\sigma^2}} \tag{7}$$

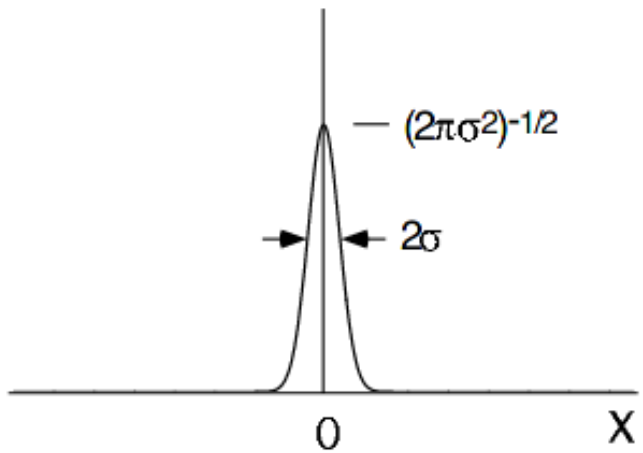


Figure 1: Gaussian model for the Dirac δ function.

Dirac δ function

- δ function times any arbitrary function $f(x)$ is:

$$\int_{\infty}^{\infty} \mathrm{d}x\ \delta(x) f(x) = f(0). \tag{8}$$

- Other models for δ function include Fourier integral:

$$\delta_\sigma(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathrm{d}k\ \mathrm{e}^{ikx - |\sigma| x}, \tag{9}$$

  and scaled sinc:

$$\delta_\sigma(x) = \frac{1}{\pi\sigma} \mathrm{sinc}(x/\sigma). \tag{10}$$

- Important for building histograms (notice the change of variables):

$$\int_{-\infty}^{\infty} \mathrm{d}x\ \delta(x - a) f(x) = f(a). \tag{11}$$

Definitions

- *Microscopic variable*: A variable pertaining to the individual atoms and molecules making up the system.
- *Macroscopic variable*: A measurable quantity used to describe the state of the system. It depends collectively on the behavior of all the atoms and molecules. These arealso referred to as *thermodynamic variables*.
- *Extensive variables*: The system under consideration is often defined as encompassing some specific N molecules. Then extensive variables are those whose magnitude is proportional to N.
- *Intensive variables*: Those macroscopic variables whose magnitude is independent of N.

Scales

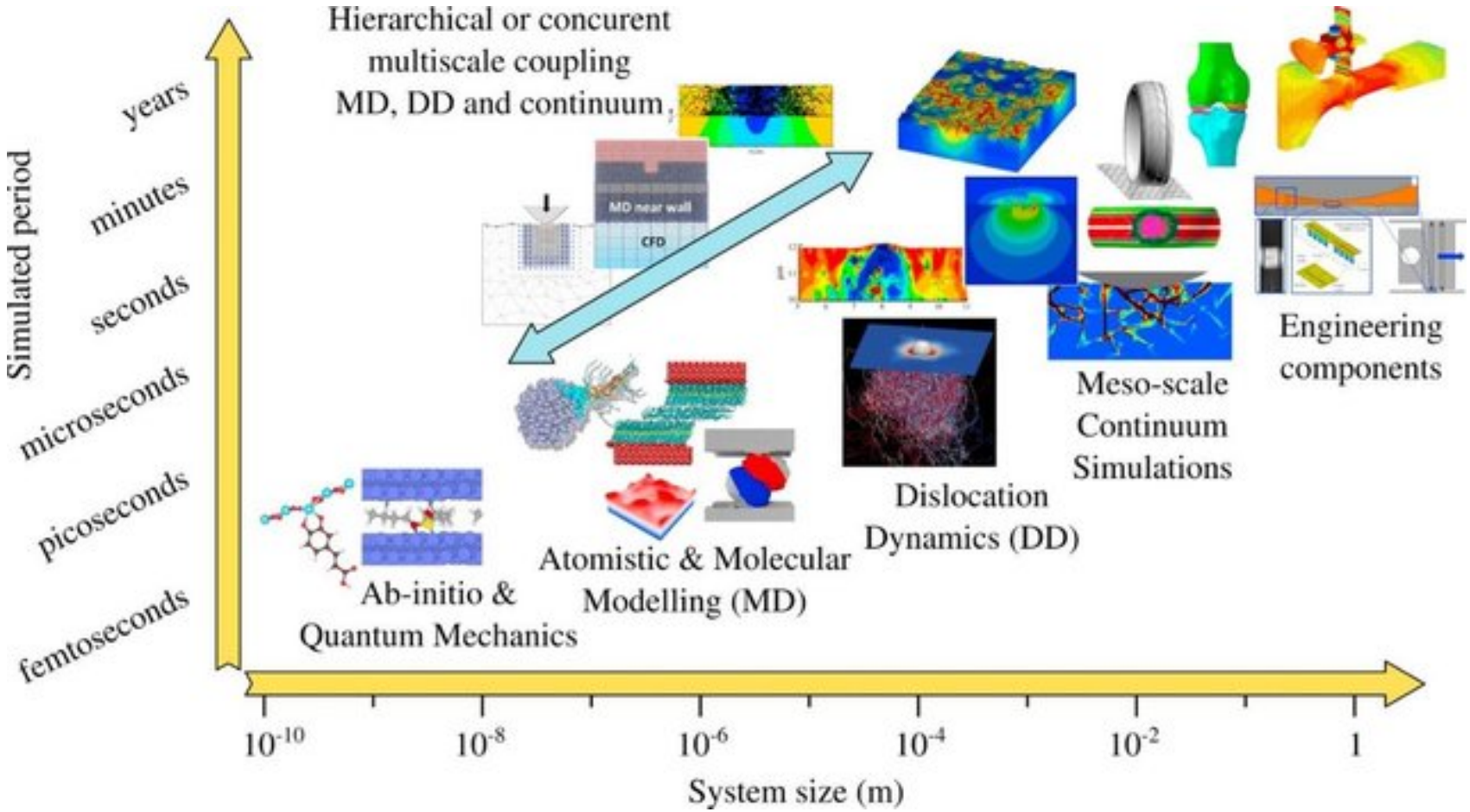


Figure 2: Spatial and temporal scales.

Phase Space

Figure 3: Ludwig Boltzmann (1844–1906)

Phase Space

- Consider a system with N particles.
- Microscopic coordinates: $\mathbf{r} \equiv (r_1, \ldots, r_{3N})$.
- Conjugate momenta: $\mathbf{p} \equiv (p_1, \ldots, p_{3N})$.
- We introduce the notion of generalized coordinates by stacking coordinates and momenta:

$$\mathbf{x} \equiv (r_1, \ldots, r_{3N}, p_1, \ldots, p_{3N}). \tag{12}$$

- $\mathbf{x}$ evolves in Γ which defines the $6N$-dimensional *phase space*.

Phase Space

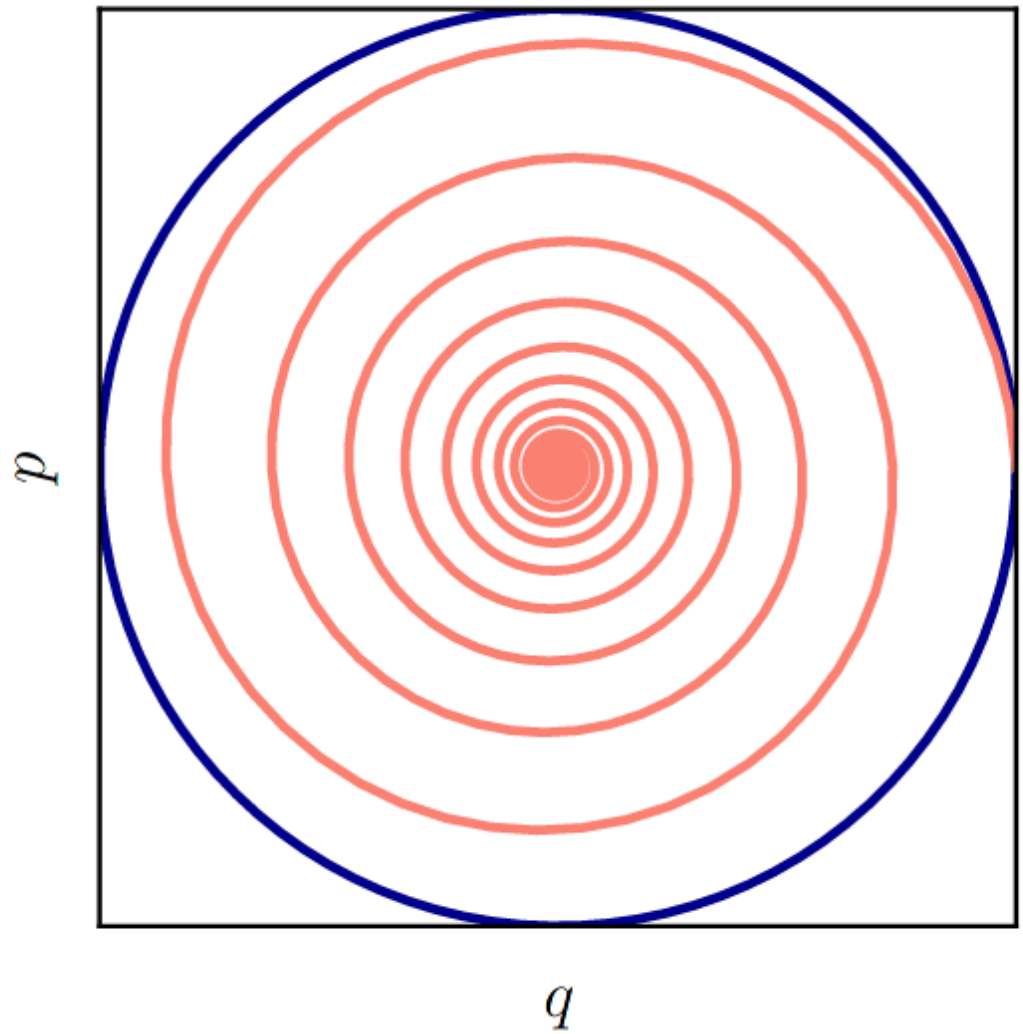


Figure 4: Phase spaces of an undamped (blue) and a damped (orange) harmonic oscillator.

Phase Space

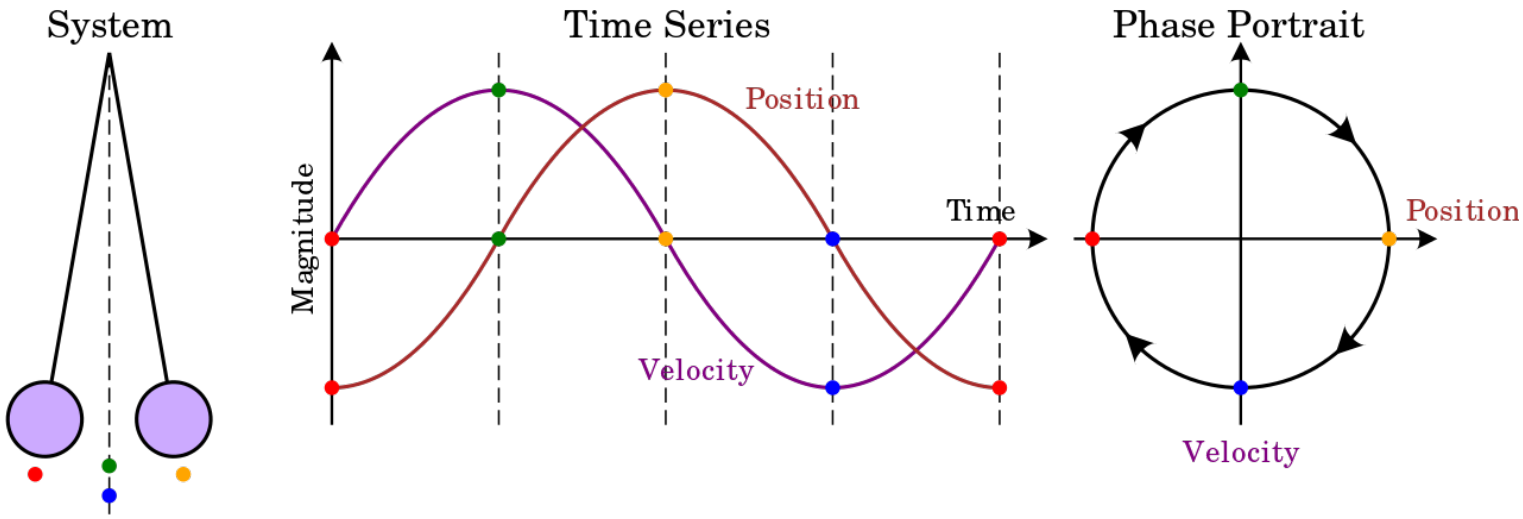


Figure 5: Phase spaces of the system.

Ensemble Average

- *Ergodicity hypothesis* – the assumption that all states in an ensemble are reached by the time evolution of the corresponding system.
- We define the ensemble average of a quantity $A(\mathbf{x})$ as:

$$\langle A \rangle = \frac{\int \mathrm{d}\mathbf{x}\ A(\mathbf{x})\rho(\mathbf{x})}{\int \mathrm{d}\mathbf{x}\ \rho(\mathbf{x})}, \tag{13}$$

  where $\rho(\mathbf{x})$ is the phase space probability density.

Hamiltonian

- The dynamics of the system under study is described by their *Hamiltonian* $H(\mathbf{x}) = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i} + U(\mathbf{r})$.
- The equations of motion are:

$$\dot{p}_i = -\frac{\partial H}{\partial r_i} \quad \text{and} \quad \dot{r}_i = \frac{\partial H}{\partial p_i} \quad (i = 1, \ldots, 3N). \tag{14}$$

- Can be rewritten as:

$$\dot{p}_i = -\frac{\partial U}{\partial r_i} \quad \text{and} \quad \dot{r}_i = \frac{p_i}{m_i}, \tag{15}$$

where $U(\mathbf{r})$ is the potential energy.

Liouville Theorem

- Temporal evolution of a phase space element of volume V and boundary ∂V is given by:

$$\frac{\partial}{\partial t}\int_V \mathrm{d}V\ \rho + \int_{\partial V} \mathrm{d}B\ \rho\mathbf{v} = 0, \tag{16}$$

  where $\mathbf{v}$ is a generalized velocity vector.

- In Eq. 16, ρ satisfies the continuity equation:

$$\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho\mathbf{v}) = 0, \tag{17}$$

  which simplifies to ($\nabla\cdot\mathbf{v} = 0$):

$$\frac{\partial \rho}{\partial t} = \{H, \rho\} \tag{18}$$

Liouville Theorem

- The Liouville Theorem describes the time evolution of the phase space density $\dot{\rho} = \{H, \rho\}$.
- Poisson bracket: $\{u, v\} = \sum_i \left(\frac{\partial u}{\partial r_i} \frac{\partial v}{\partial p_i} - \frac{\partial u}{\partial p_i} \frac{\partial v}{\partial r_i} \right)$.

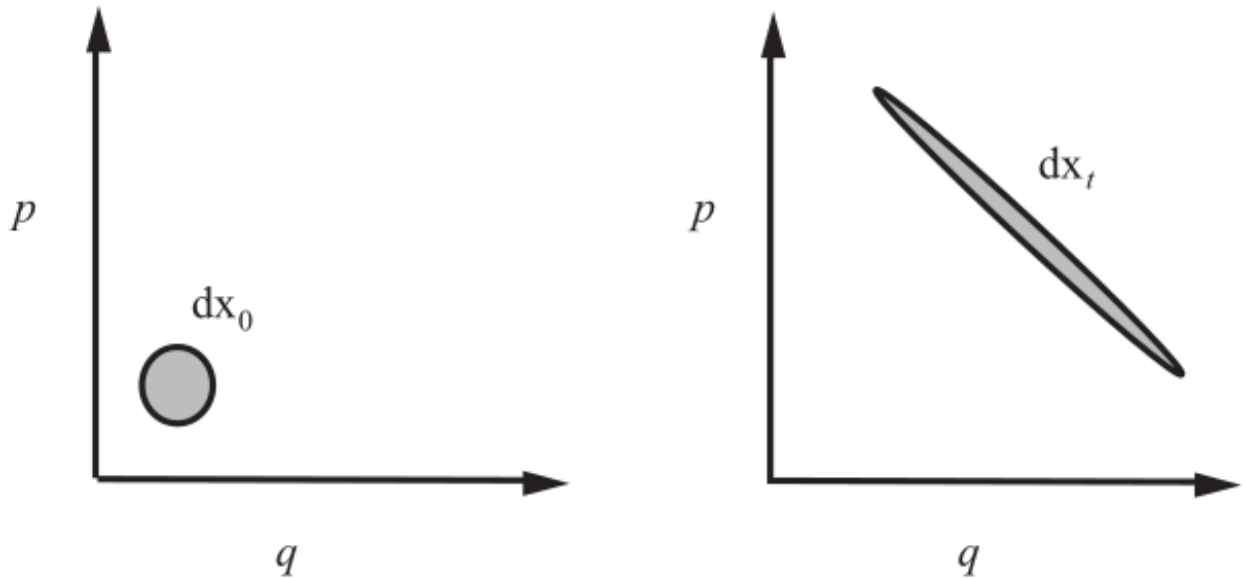


Figure 6: Illustration of phase space volume conservation prescribed by Liouville's theorem.

Source: M. E. Tuckerman, *Statistical Mechanics: Theory and Simulation*, Oxford University Press (2016).

Thermodynamic Equilibrium

- In general, if $\rho(\mathbf{x}, t)$ has an explicit time dependence, then so will an observable A.
- However, a system in thermodynamic equilibrium do not change in time, $\partial\rho/\partial t$ must be equal to 0.
- In such a case, no external forces act on the system.
- The Liouville equation reduces to:

$$\{\rho, H\} = 0. \tag{19}$$

Thermodynamic Equilibrium

- The general solution to Eq. 19 is any function of H:

$$\rho(\mathbf{x}) \propto F\left[H(\mathbf{x})\right]. \tag{20}$$

- As ρ needs to be properly normalized, we write the solutions as:

$$\rho(\mathbf{x}) = \frac{1}{Z} F[H(\mathbf{x})], \tag{21}$$

  where $Z = \int \mathrm{d}\mathbf{x}\ F[H(\mathbf{x})]$ is referred to as the *partition function*.

- The partition function is a measure of the number of microscopic states in the phase space accessible within a given ensemble.

- Each ensemble has a particular partition function that depends on the macroscopic observables used to define the ensemble.

Ensembles

- N – the number of particles, V – volume, P – pressure, T – temperature, E – internal energy.
- Microcanonical ensemble: constant NVE.
- Canonical ensemble: constant NVT.
- Canonical pressure ensemble: constant NPT.

Microcanonical Ensemble

- NVE are fixed, so also the probability density is constant:

$$\rho(\mathbf{x}) = \frac{1}{Z}\delta[H(\mathbf{x}) - E], \tag{22}$$

where $\delta[\cdot]$ is the Dirac delta function and the microcanonical partition function is given by:

$$Z = \int \mathrm{d}\mathbf{x}\, \delta[H(\mathbf{x}) - E]. \tag{23}$$

- We can see that the function $F[H(\mathbf{x})]$ from Eq. 20 is:

$$F[H(\mathbf{x})] \propto \delta[H(\mathbf{x}) - E]. \tag{24}$$

Microcanonical Ensemble

- State function is the entropy – a quantity that can be related to the number of the microscopic states of the system.
- Control variables are:

$$\frac{1}{T} = \left(\frac{\partial S}{\partial E}\right)_{N,V}, \frac{P}{T} = \left(\frac{\partial S}{\partial V}\right)_{N,E}, \frac{\mu}{T} = -\left(\frac{\partial S}{\partial N}\right)_{V,E}. \tag{25}$$

- Let Ω be the number of microscopic states; the relation of S to Ω is:

$$S(N, V, E) = k_{\mathrm{B}} \log \Omega(N, V, E), \tag{26}$$

where k_{B} is Boltzmann's constant:

$$k_{\mathrm{B}} = 1.3806505(24) \times 10^{-23}\ \mathrm{J/K}. \tag{27}$$

Canonical Ensemble

- NVT are fixed.

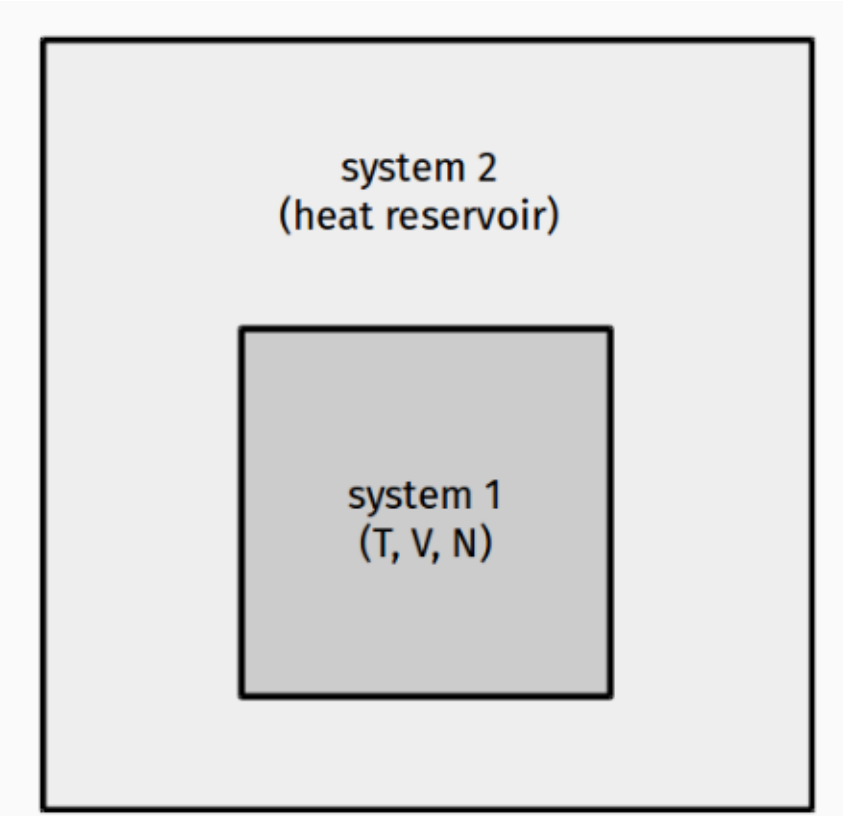


Figure 7: In a canonical ensemble setup, the system we study (system 1) is coupled to a heat reservoir (system 2) that guarantees a constant temperature.

Source: M. E. Tuckerman, *Statistical Mechanics: Theory and Simulation*, Oxford University Press (2016).

Canonical Ensemble

- At a given temperature T, the probability for a system to be in a certain configuration $\mathbf{x}$ with energy $E(\mathbf{x})$ is:

$$\rho(\mathbf{x}) = \frac{1}{Z} \mathrm{e}^{-\beta E(\mathbf{x})}, \tag{28}$$

where $\beta = \frac{1}{k_{\mathrm{B}} T}$ is the *inverse temperature*[2], and the canonical partition function is:

$$Z = \int \mathrm{d}\mathbf{x}\ \mathrm{e}^{-\beta E(\mathbf{x})}. \tag{29}$$

- The ensemble average of a quantity A is given by:

$$\langle A \rangle = \frac{1}{Z} \int \mathrm{d}\mathbf{x}\ A(\mathbf{x}) \mathrm{e}^{-\beta E(\mathbf{x})}. \tag{30}$$

[2] $\beta \approx 0.4$ kJ/mol at 300 K.

Canonical Ensemble

- We can derive Eq. 28 by using the maximum entropy princliple. We know that:

$$\int \mathrm{d}\mathbf{x}\ \rho(\mathbf{x})E(\mathbf{x}) = U, \tag{31}$$

and that ρ must be normalized.

- We introduce the augumented function J:

$$J \equiv -k_{\mathrm{B}} \int \mathrm{d}\mathbf{x}\ \rho(\mathbf{x}) \log \rho(\mathbf{x}) \tag{32}$$

$$+ \lambda_1 \left(\int \mathrm{d}\mathbf{x}\ \rho(\mathbf{x})E(\mathbf{x}) - U \right) \tag{33}$$

$$+ \lambda_0 \left(\int \mathrm{d}\mathbf{x}\ \rho(\mathbf{x}) - 1 \right), \tag{34}$$

where λ_0 and λ_1 are the Lagrange multipliers.

Canonical Ensemble

- Taking the functional derivative w.r.t. to ρ and setting it to zero, we have:

$$\frac{\delta J}{\delta \rho} = -k_{\mathrm{B}} \log \rho(\mathbf{x}) - k_{\mathrm{B}} + \lambda_1 E + \lambda_0 = 0, \tag{35}$$

- which implies that the maximum entropy distribution is:

$$\rho(\mathbf{x}) = \lambda_1 \mathrm{e}^{\frac{-k_{\mathrm{B}} + \lambda_0 + \lambda_1 E(\mathbf{x})}{k_{\mathrm{B}}}}, \tag{36}$$

which translates to:

$$\rho(\mathbf{x}) = Z \mathrm{e}^{\frac{-k_{\mathrm{B}} + \lambda_0}{k_{\mathrm{B}}}} \tag{37}$$

where Z is the canonical partition function:

$$Z = \int \mathrm{d}\mathbf{x}\ \mathrm{e}^{\frac{\lambda_1}{k_{\mathrm{B}}} E}. \tag{38}$$

Canonical Ensemble

- We get that the density must have a form:

$$\rho(\mathbf{x}) = \frac{1}{Z} \mathrm{e}^{\frac{\lambda_1}{k_\mathrm{B}} E(\mathbf{x})} \tag{39}$$

and the entropy is:

$$S(\mathbf{x}) = -\lambda_1 U + k_\mathrm{B} \log Z. \tag{40}$$

- We differentiate S w.r.t. U and apply $\mathrm{d}U = T\mathrm{d}S - P\mathrm{d}V$:

$$\frac{\mathrm{d}S}{\mathrm{d}U} = -\lambda_1 \equiv \frac{1}{T} \tag{41}$$

from which we get that:

$$\frac{\lambda_1}{k_\mathrm{B}} = -\frac{1}{k_\mathrm{B} T} = -\beta. \tag{42}$$

Monte Carlo Methods

Monte Carlo Methods

The main steps of the Monte Carlo sampling are:

- Choose randomly a new configuration in phase space based on a Markov chain.
- Accept or reject the new configuration, depending on the strategy used.
- Compute the physical quantity and add it to the averaging procedure.
- Repeat the previous steps until convergence.

Monte Carlo Methods

$$I = \int_0^1 \mathrm{d}x \int_0^{\sqrt{1-x^2}} \mathrm{d}y = \frac{\pi}{4} \tag{43}$$

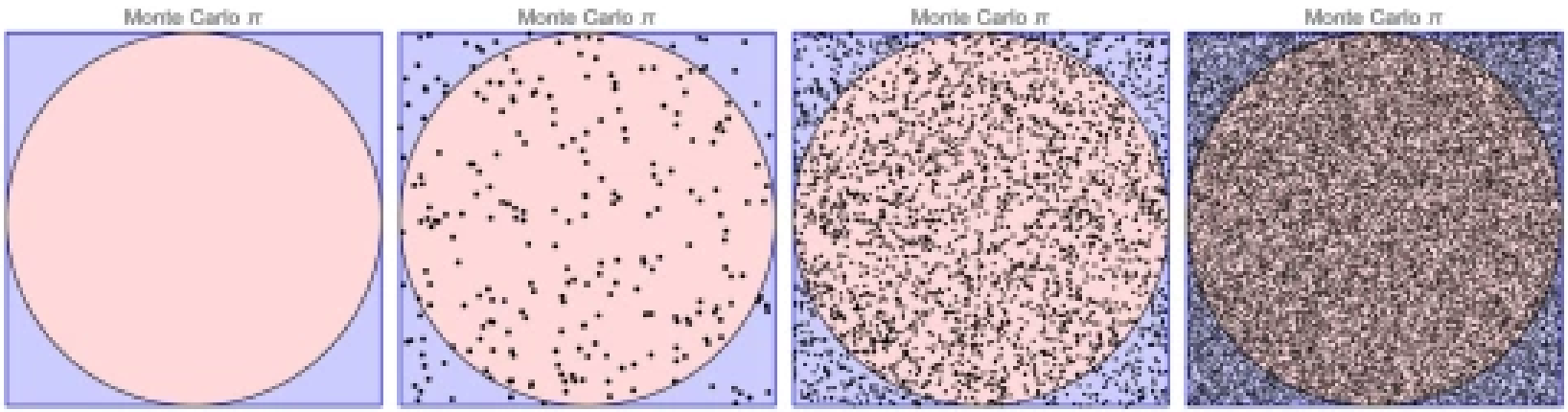


Figure 8: Example of a Monte Carlo method to compute π.

Source: https://thatsmaths.com/2020/05/28/the-monte-carlo-method/.

Monte Carlo Methods

- The integrals that must be evaluated in equilibrium statistical mechanics are generally of the form:

$$I = \int \mathrm{d}\mathbf{x}\, \phi(\mathbf{x}) p(\mathbf{x}), \tag{44}$$

where $\mathbf{x}$ is an n-dimensional vector, $\phi(\mathbf{x})$ is an arbitrary function, and $p(\mathbf{x})$ is a function satisfying the properties of a probability distribution:

$$p(\mathbf{x}) \geq 0 \quad \text{and} \quad \int \mathrm{d}\mathbf{x}\, p(\mathbf{x}) = 1. \tag{45}$$

- Eq. 44 represents the ensemble average of a physical observable in equilibrium statistial mechanics.

Monte Carlo Methods

- Let $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M$ be a set of M n-dimensional vectors that are sampled from $p(\mathbf{x})$.
- The problem of sampling from $p(\mathbf{x})$ is a nontrivial one.
- For now, let us assume that such an algorithm exists. Then, an estimator:

$$\hat{I}_M = \frac{1}{M} \sum_{i=1}^{M} \phi(\mathbf{x}_i) \tag{46}$$

  is such that:

$$\lim_{M \to \infty} \hat{I}_M = I. \tag{47}$$

- Eq. 47 is guaranteed by the *central limit theorem*.[3]

[3] For the derivation, see M. E. Tuckerman, *Statistical Mechanics: Theory and Simulation*, Oxford University Press (2016), p. 281.

Monte Carlo Methods

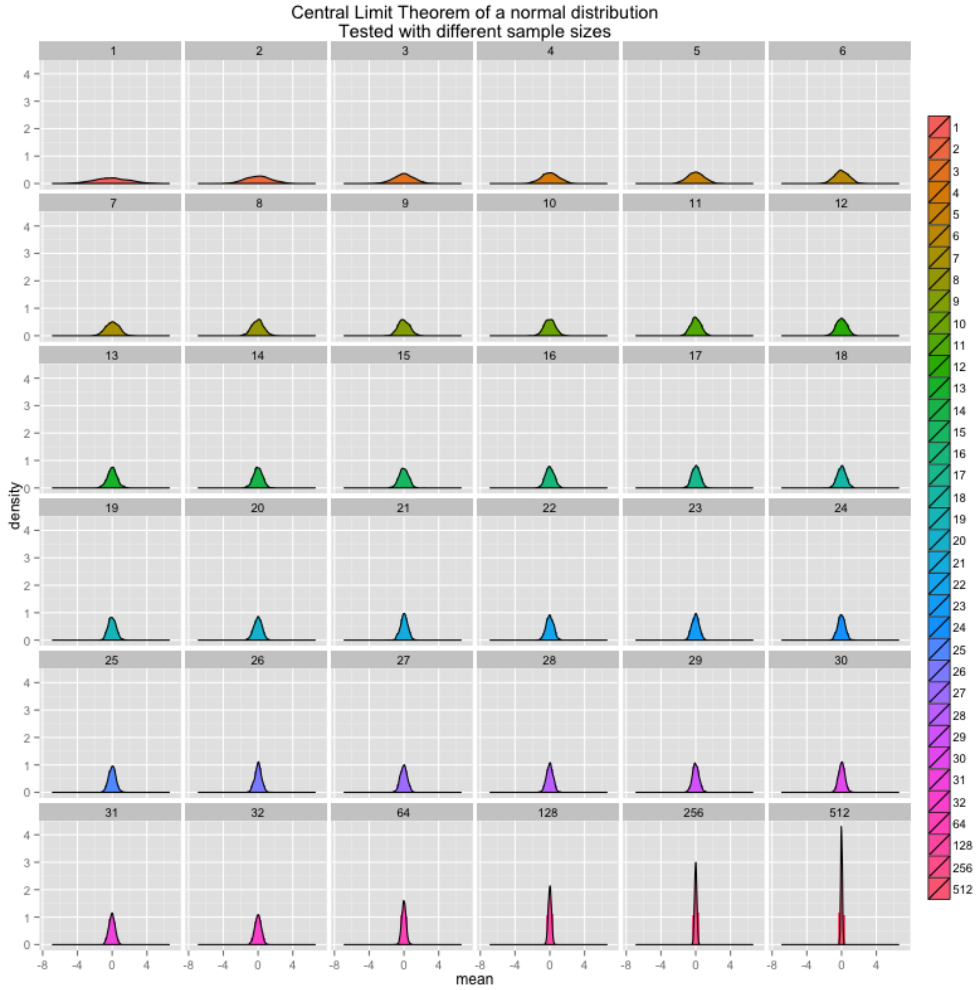


Figure 9: Central limit theorem for a normal distribution.

Source: https://www.reddit.com/r/math/comments/2wrduu/central_limit_theorem/.

Monte Carlo Methods

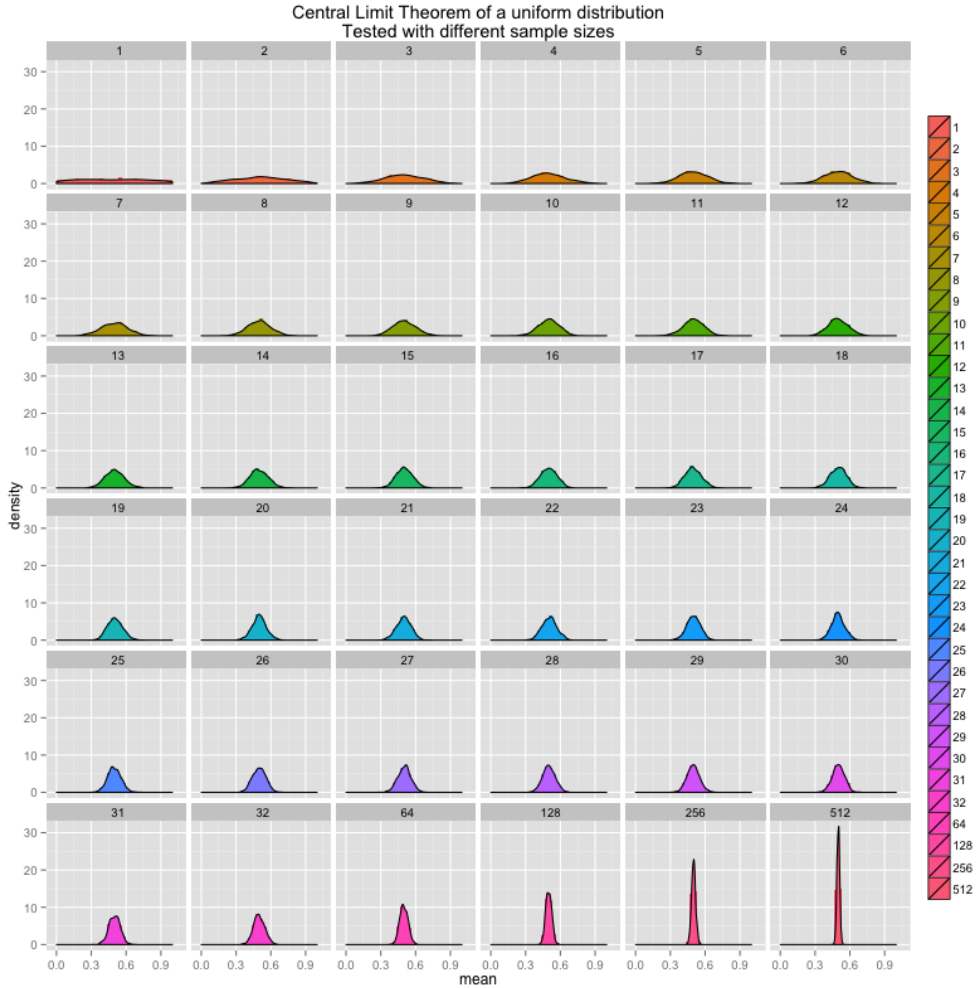


Figure 10: Central limit theorem for a uniform distribution.

Source: https://www.reddit.com/r/math/comments/2wrduu/central_limit_theorem/.

Monte Carlo Methods

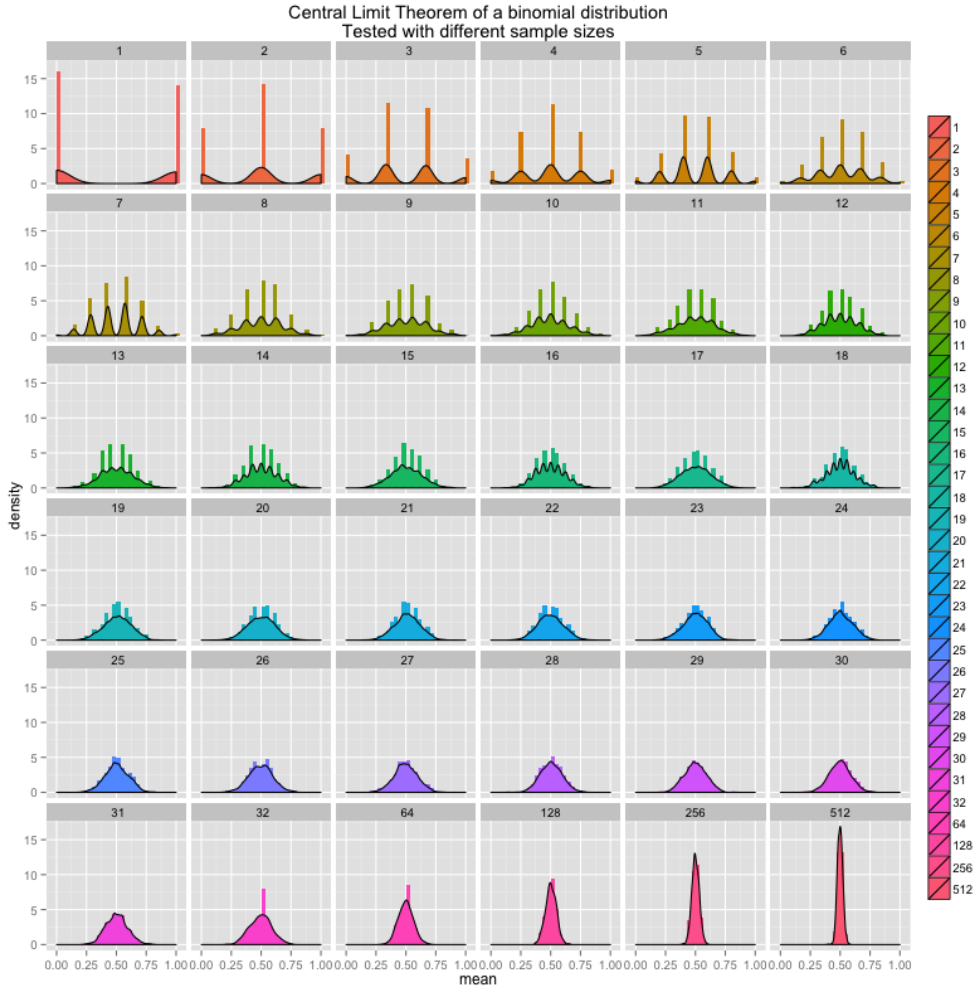


Figure 11: Central limit theorem for a binomial distribution.

Source: https://www.reddit.com/r/math/comments/2wrduu/central_limit_theorem/.

Central Limit Theorem

If $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ are n random variables drawn from a population with overall mean μ and finite variance σ^2, and if $\bar{\mathbf{x}}_n$ is the sample mean, the limiting form of the distribution, $\lim_{n\to\infty} \sqrt{n}\left(\frac{\bar{\mathbf{x}}_n - \mu}{\sigma}\right)$, is the standard normal distribution.

Monte Carlo Methods

- Example of sampling a simple distribution given by:

$$p(x) = c\mathrm{e}^{-cx}, \tag{48}$$

on the interval $x \in [0, \infty)$.

- To sample from $p(x)$, we need $P(X)$ such that:

$$P(X) = \int_0^X \mathrm{d}x \; c\mathrm{e}^{-cx} = 1 - \mathrm{e}^{-cX}, \tag{49}$$

and equate Eq. 49 to a random number $\xi \in [0, 1]$, then we can solve for X, which gives:

$$X = -\frac{1}{c} \log(1 - \xi). \tag{50}$$

- In general, we do not have such simple distribution to sample from in statistical mechanics.

Monte Carlo Methods: Examples

Estimator no 1

```
def pi_estimator(n_samples):
  est = list()
  n_inside = 0; n_all = 0
  for i in range(n_samples):
    x = np.random.uniform(-1.0, 1.0)
    y = np.random.uniform(-1.0, 1.0)
    dist = np.sqrt(np.power(x, 2) +
                   np.power(y, 2))
    if dist <= 1.0:
      n_inside += 1
    n_all += 1
    est.append(4.0 * np.float(n_inside) /
               np.float(n_all))
  return np.array(est)
```

Monte Carlo Methods: Examples

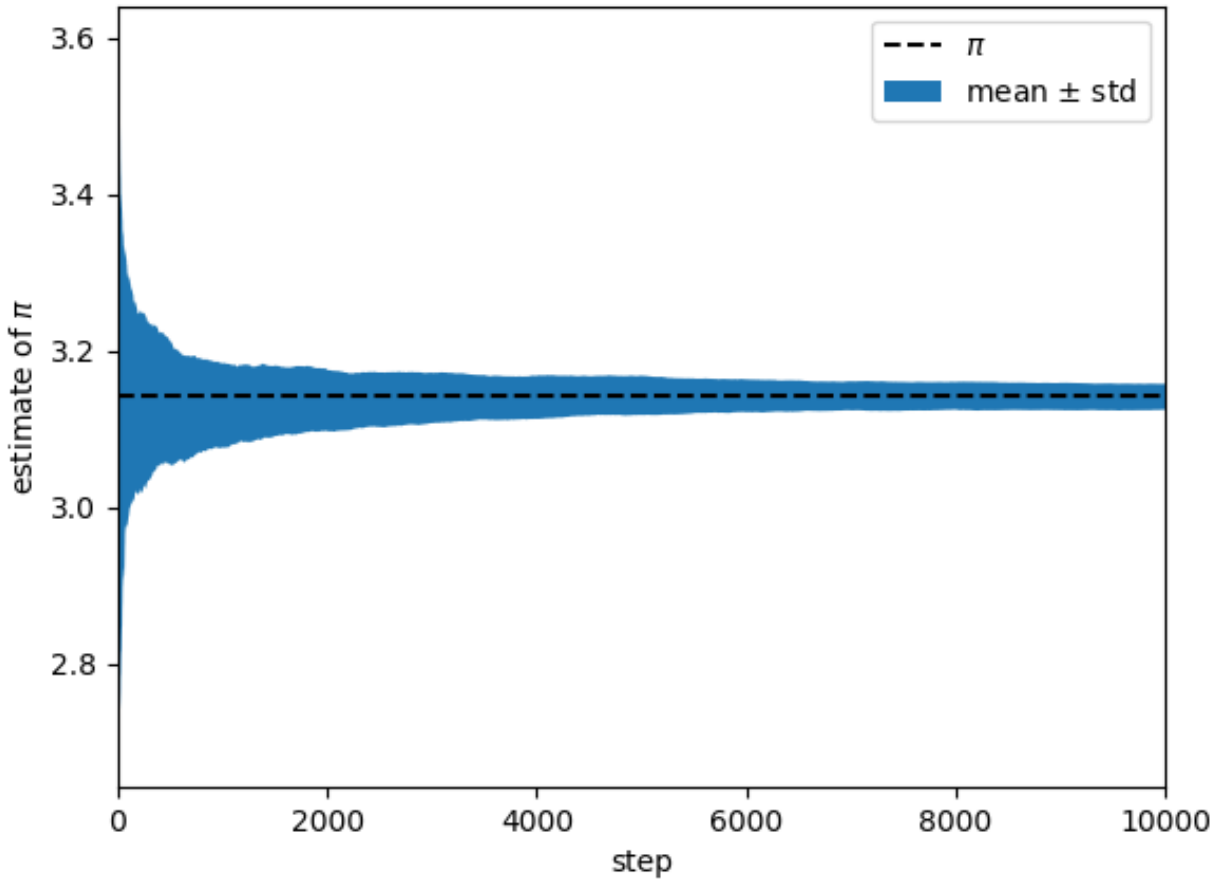


Figure 12: Estimator no 1.

Monte Carlo Methods: Examples

Estimator no 2

```
def pi_estimator_integral(n_samples):
  est = list()
  e = 0.0
  for i in range(n_samples):
    x = np.random.uniform(0.0, 1.0)
    y = 4.0 * np.sqrt(1.0 - np.power(x, 2))
    e += (y - e) / (np.float(i) + 1.0)
    est.append(e)
  return np.array(est)
```

Monte Carlo Methods: Examples

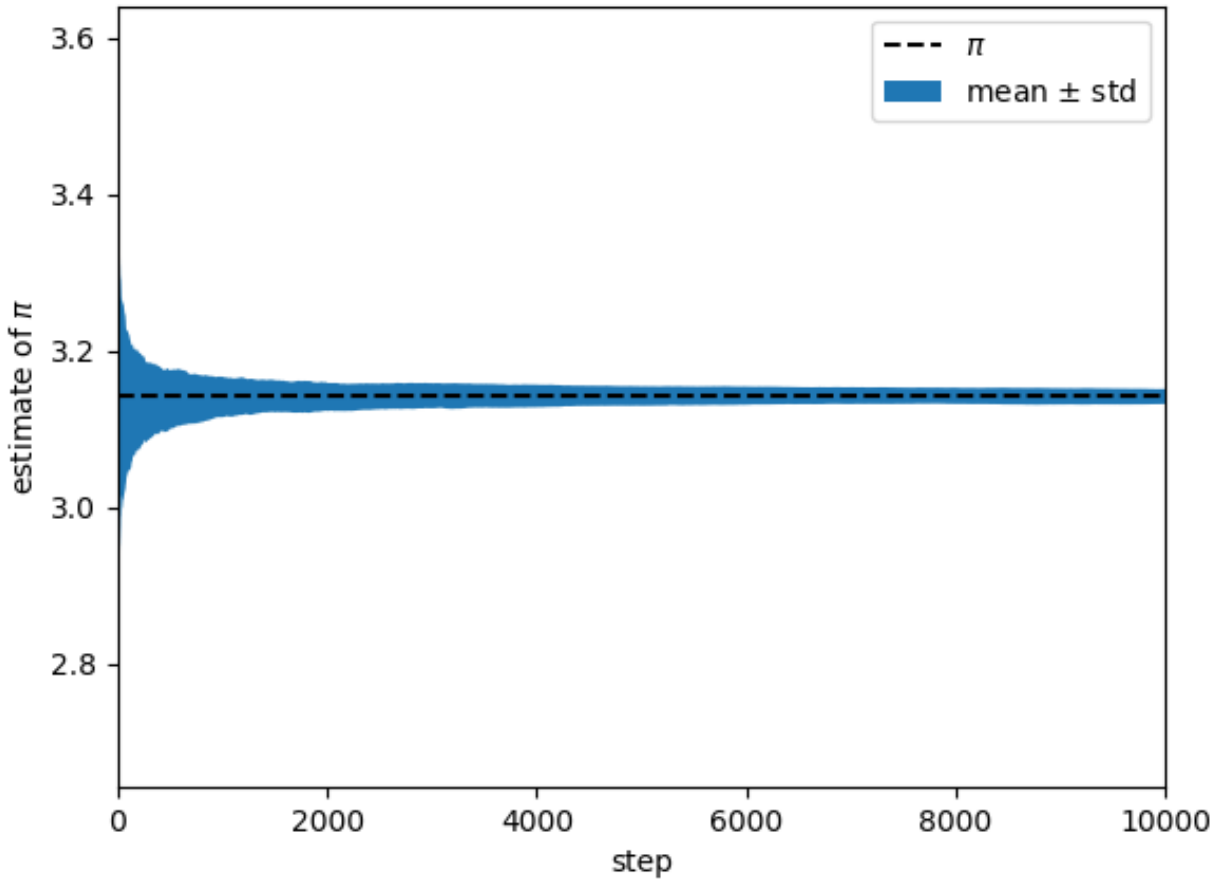


Figure 13: Estimator no 2.

Monte Carlo Methods: Examples

Estimator no 3

```
def pi_estimator_markov_chain(n_samples, step=0.1):
  est = list()
  n_inside = 0; n_all = 0;
  x = np.random.uniform(-1.0, 1.0)
  y = np.random.uniform(-1.0, 1.0)
  for i in range(n_samples):
    dx = np.random.uniform(-step, step)
    dy = np.random.uniform(-step, step)
    xn = x + dx; yn = y + dy
    if abs(xn) < 1.0 and abs(yn) < 1.0:
      x = xn; y = yn
    dist = np.sqrt(np.power(x, 2) + np.power(y, 2))
    if dist <= 1.0: n_inside += 1
    n_all += 1
    est.append(4.0 * np.float(n_inside) /
               np.float(n_all))
  return np.array(est)
```

Monte Carlo Methods: Examples

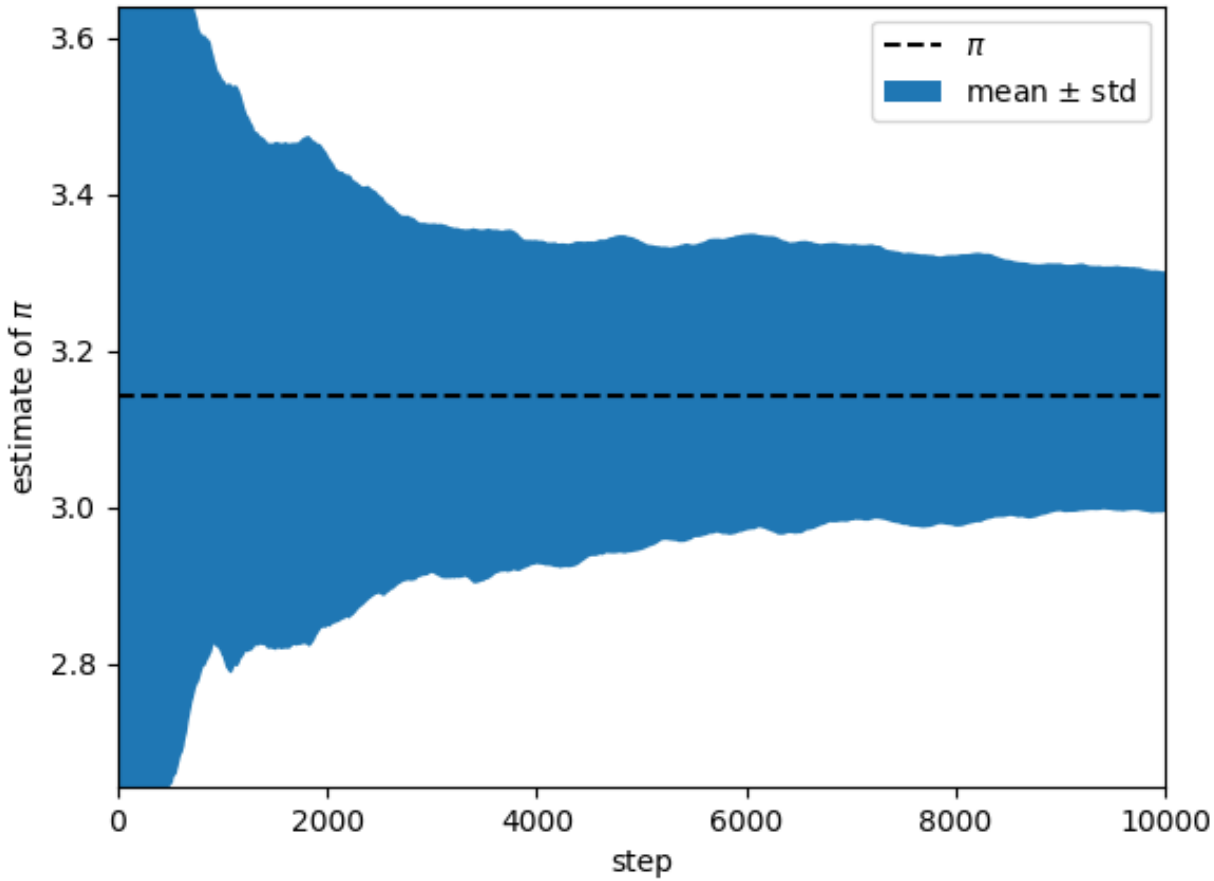


Figure 14: Estimator no 3.

Importance Sampling

- However, instead of sampling $p(\mathbf{x})$ to estimate $\int \mathrm{d}\mathbf{x}\ \phi(\mathbf{x})p(\mathbf{x})$, we could sample from a different distribution $b(\mathbf{x})$ by rewriting the integral as:

$$I = \int \mathrm{d}\mathbf{x}\ \left[\frac{\phi(\mathbf{x})p(\mathbf{x})}{b(\mathbf{x})}\right] b(\mathbf{x}). \tag{51}$$

- We set $\psi(\mathbf{x}) = \phi(\mathbf{x})p(\mathbf{x})/b(\mathbf{x})$ which gives us:

$$I = \int \mathrm{d}\mathbf{x}\ \psi(\mathbf{x})b(\mathbf{x}) \tag{52}$$

$$\approx \frac{1}{M}\sum_{i=1}^{M} \psi(\mathbf{x}_i), \tag{53}$$

  where the vectors $\mathbf{x}_i$ are sampled from $b(\mathbf{x})$.

- This trick is the basis of *importance sampling*.

Importance Sampling

- Using $b(\mathbf{x})$ as an importance function may lead to easier sampling.
- But how do we select $b(\mathbf{x})$?
- Is there an optimal choice of $b(\mathbf{x})$? The best one leads to the smallest possible variance:

$$\text{Var}[b(\mathbf{x})] = \int \mathrm{d}\mathbf{x}\ \psi^2(\mathbf{x})b(\mathbf{x}) - \left(\int \mathrm{d}\mathbf{x}\ \psi(\mathbf{x})b(\mathbf{x})\right)^2 \quad (54)$$

that gives us:

$$\int \mathrm{d}\mathbf{x}\ \frac{\phi^2(\mathbf{x})p^2(\mathbf{x})}{b^2(\mathbf{x})}b(\mathbf{x}) - \left(\int \mathrm{d}\mathbf{x}\ \phi(\mathbf{x})p(\mathbf{x})\right)^2 \quad (55)$$

- Minimize Var w.r.t. $b(\mathbf{x})$ subject to the constraint $\int \mathrm{d}\mathbf{x}\ b(\mathbf{x}) = 1$

Importance Sampling

- From Eq. 55, we have:

$$F[b(\mathbf{x})] = \mathrm{Var}[b(\mathbf{x})] - \lambda \int \mathrm{d}\mathbf{x}\, b(\mathbf{x}), \tag{56}$$

  where λ is a Lagrange multiplier.

- Computing the functional derivative $\delta F / \delta b(\mathbf{x})$, we obtain:

$$\frac{\phi^2(\mathbf{x}) p^2(\mathbf{x})}{b^2(\mathbf{x})} + \lambda = 0 \tag{57}$$

  or

$$b(\mathbf{x}) = \frac{1}{\sqrt{-\lambda}} \phi(\mathbf{x}) p(\mathbf{x}). \tag{58}$$

Importance Sampling

- Normalizing $b(\mathbf{x})$, we have:

$$\int \mathrm{d}\mathbf{x}\; b(\mathbf{x}) = \frac{1}{\sqrt{-\lambda}} \int \mathrm{d}\mathbf{x}\; \phi(\mathbf{x}) p(\mathbf{x}) = 1. \tag{59}$$

- Thus, the optimal choice for $b(\mathbf{x})$ is:

$$\boxed{b(\mathbf{x}) = \frac{\phi(\mathbf{x}) p(\mathbf{x})}{I}.} \tag{60}$$

- But if we knew the integral value I, we would not need to perform the calculation...

Importance Sampling: Examples

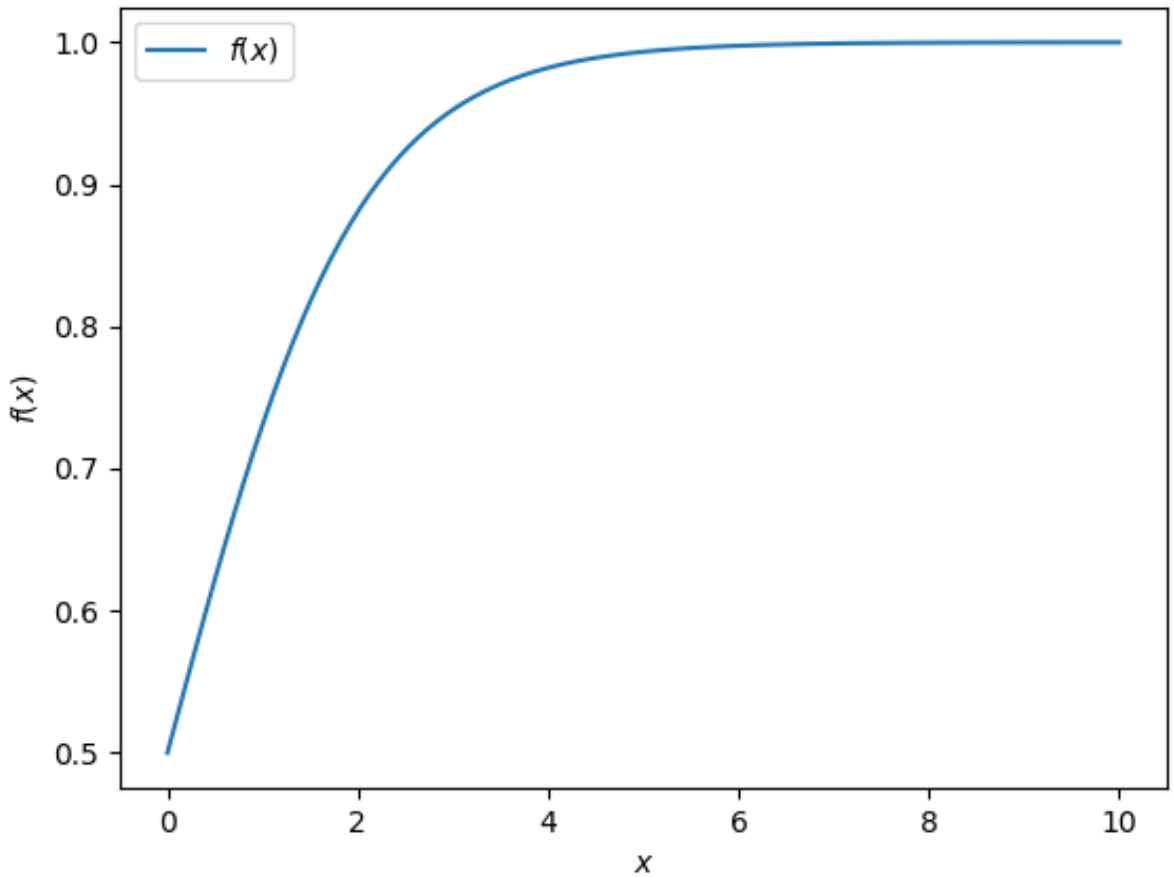


Figure 15: Function $f(x) = 1/(1 + \exp(-x))$.

Importance Sampling: Examples

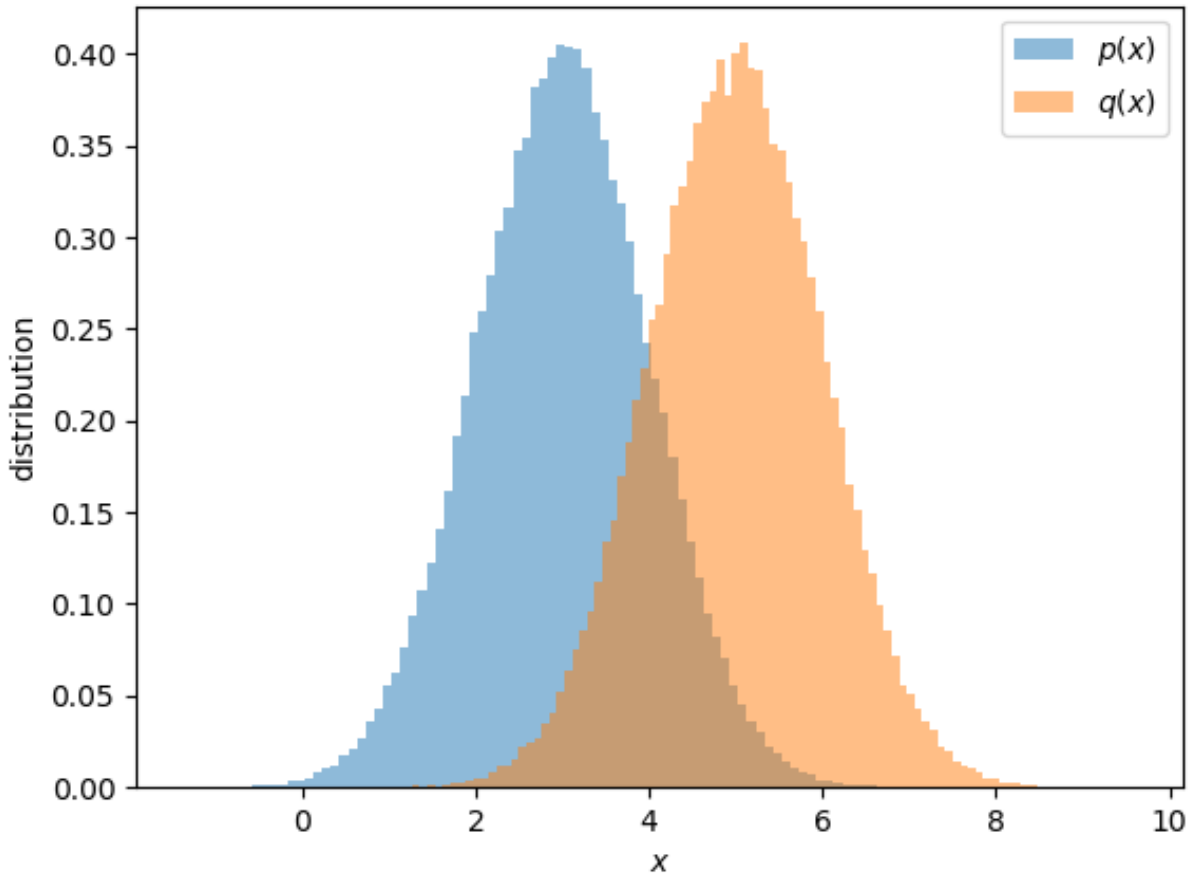


Figure 16: Distributions to sample from.

Importance Sampling: Examples

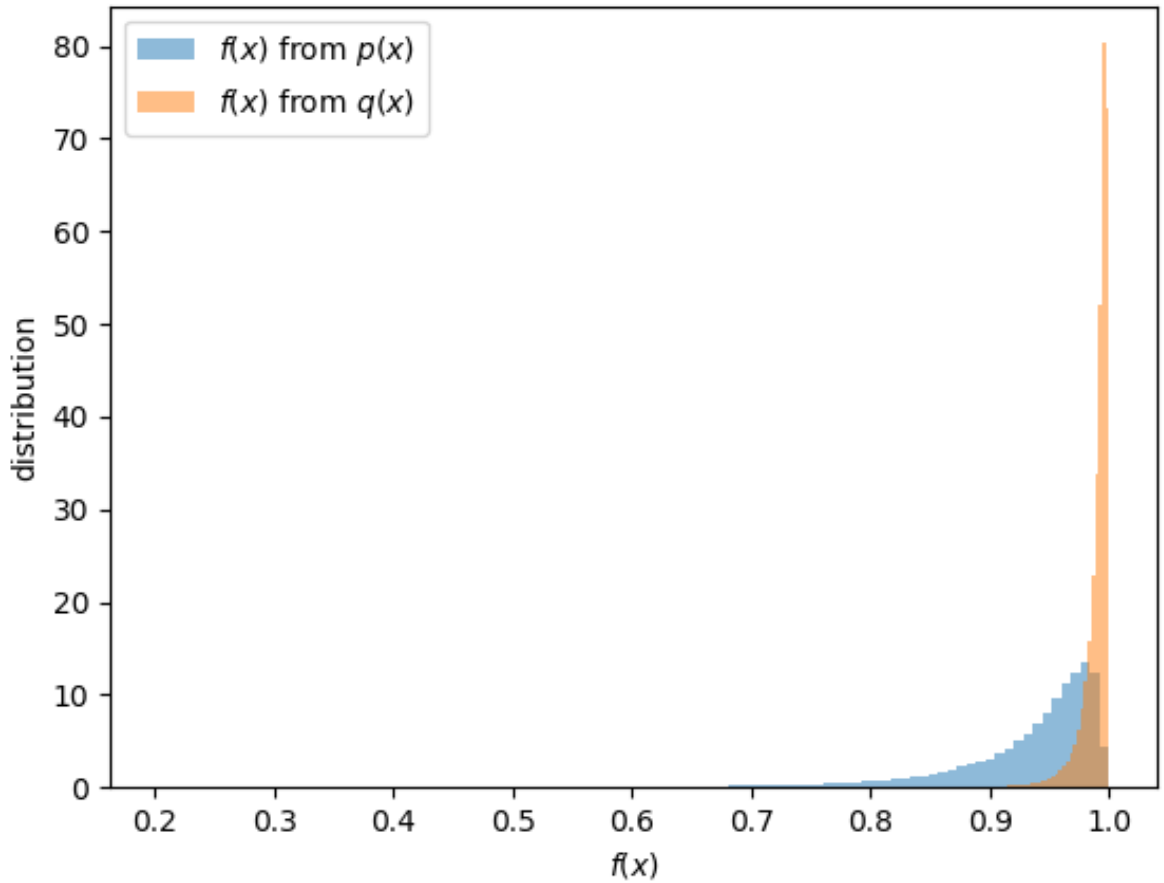


Figure 17: Distributions of evaluated samples.

Importance Sampling: Examples

Results

```
python importance-sampling.py --n_samples 100000

f(x) from p(x): mean=0.9306712389 std=0.0686291035
f(x) from q(x): mean=0.9891990210 std=0.0132275039
f(x) from IS:   mean=0.9261305966 std=4.5258393804
```

Markov Chain

If the vectors $\mathbf{x}_1, \ldots, \mathbf{x}_M$ are generated sequentially, and $\mathbf{x}_{i+1}$ is generated only based on the knowledge of $\mathbf{x}_i$. the sequence is called a Makov chain.

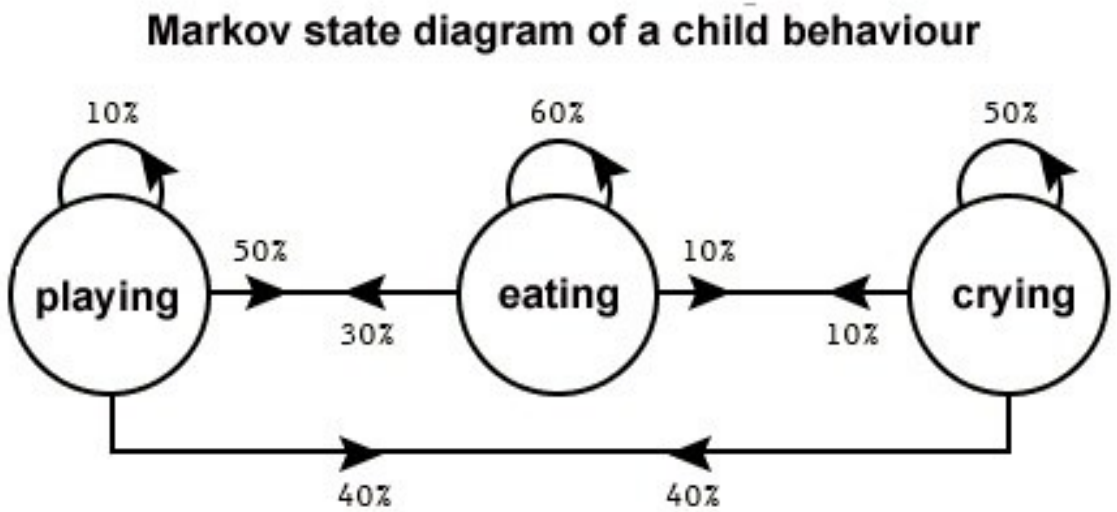


Figure 18: Example of a simple Markov chain.

M(RT)^2

- For physical systems, a Markov chain must satisfy the condition of *detailed balance*, which ensures that the Markov process is microscopically reversible.
- M(RT)^2 is a rejection method.
- With the acceptance probability given by:

$$A(\mathbf{x}_i|\mathbf{x}_j) = \min\left[1, f(\mathbf{x}_i)/f(\mathbf{x}_j)\right], \tag{61}$$

  where $f(\mathbf{x})$ is the probability of the system being at $\mathbf{x}$.
- Sampling in the canonical ensemble:[4]

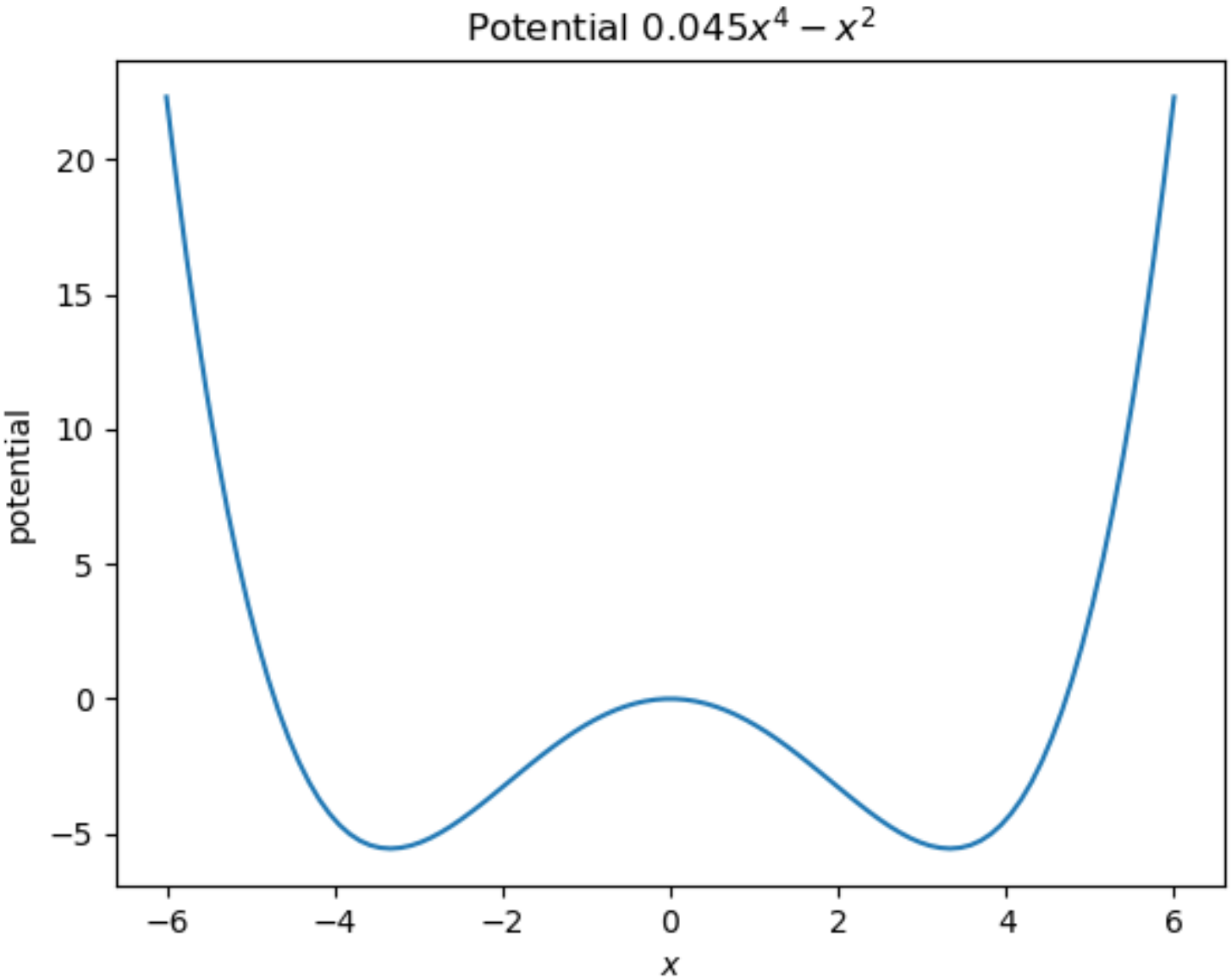
$$A(\mathbf{r}_i|\mathbf{r}_j) = \min\left[1, \mathrm{e}^{-\beta\left[U(\mathbf{r}_i)-U(\mathbf{r}_j)\right]}\right], \tag{62}$$

  as $f(\mathbf{r}) \propto \mathrm{e}^{-\beta U(\mathbf{r})}$.

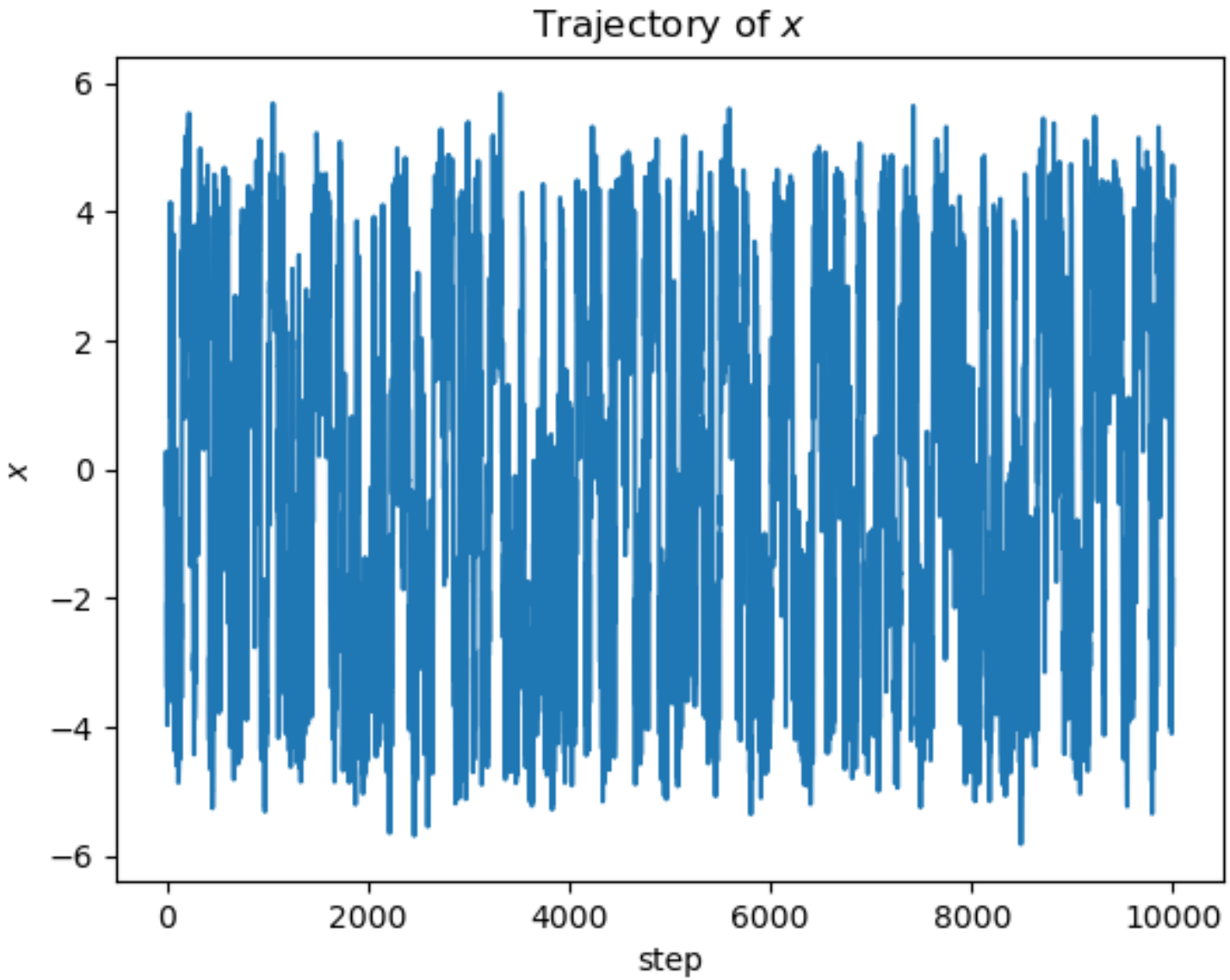
[4] Reminder: $\mathbf{r}$ are microscopic coordinates.

Particle on Double-Well Potential

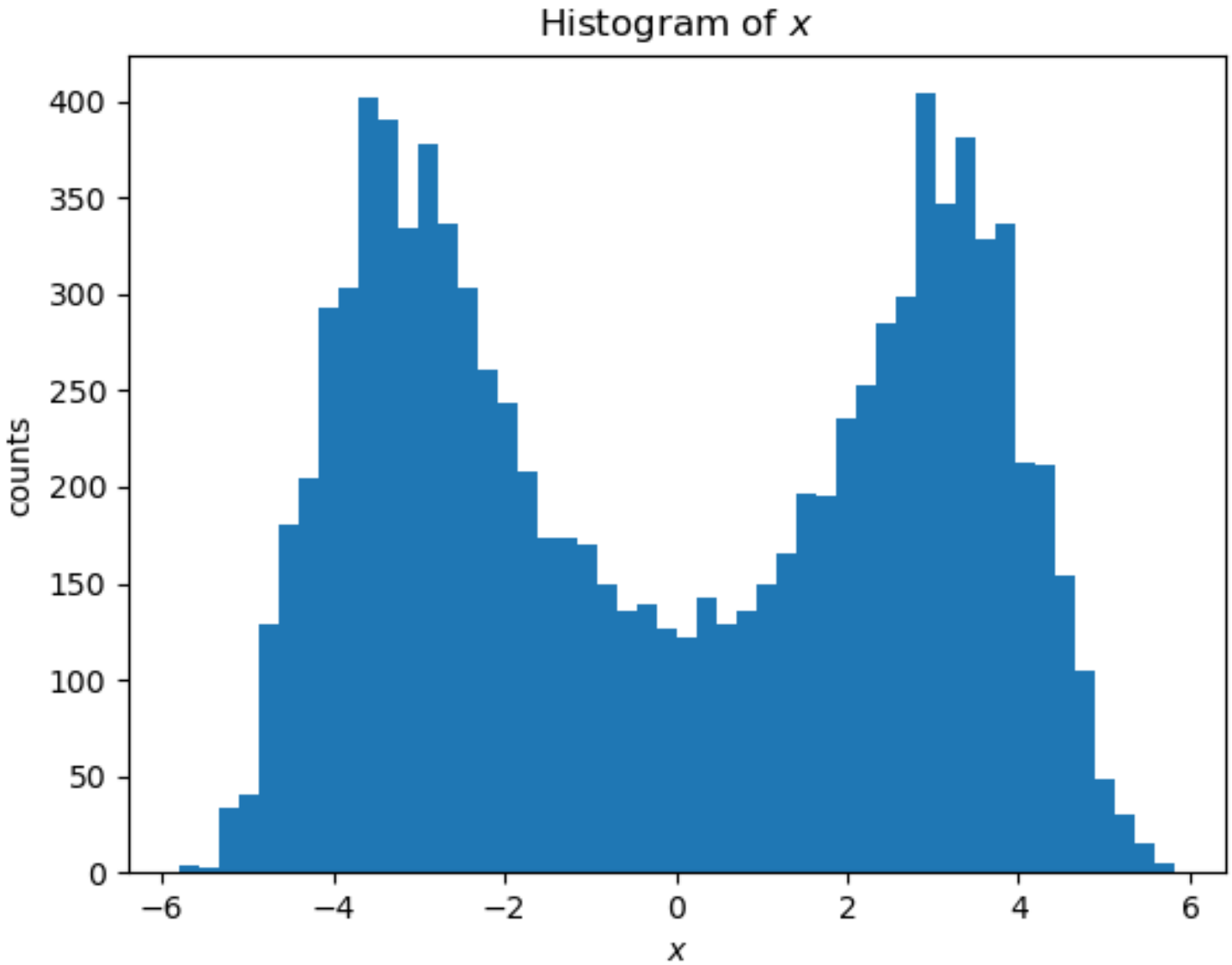
```
python mc-double-well.py --n_samples 10000 --temp 5
```



Particle on Double-Well Potential

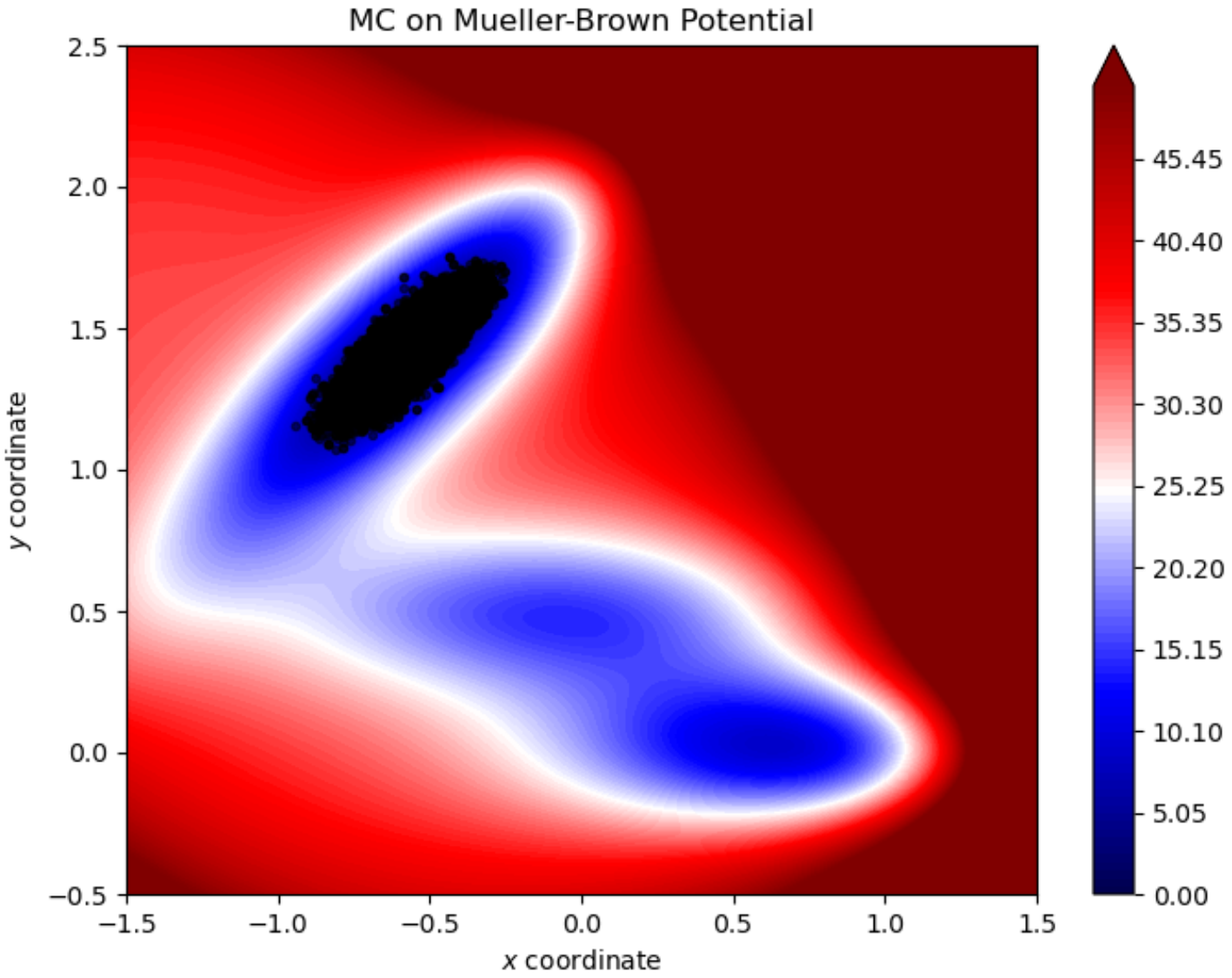


Particle on Double-Well Potential



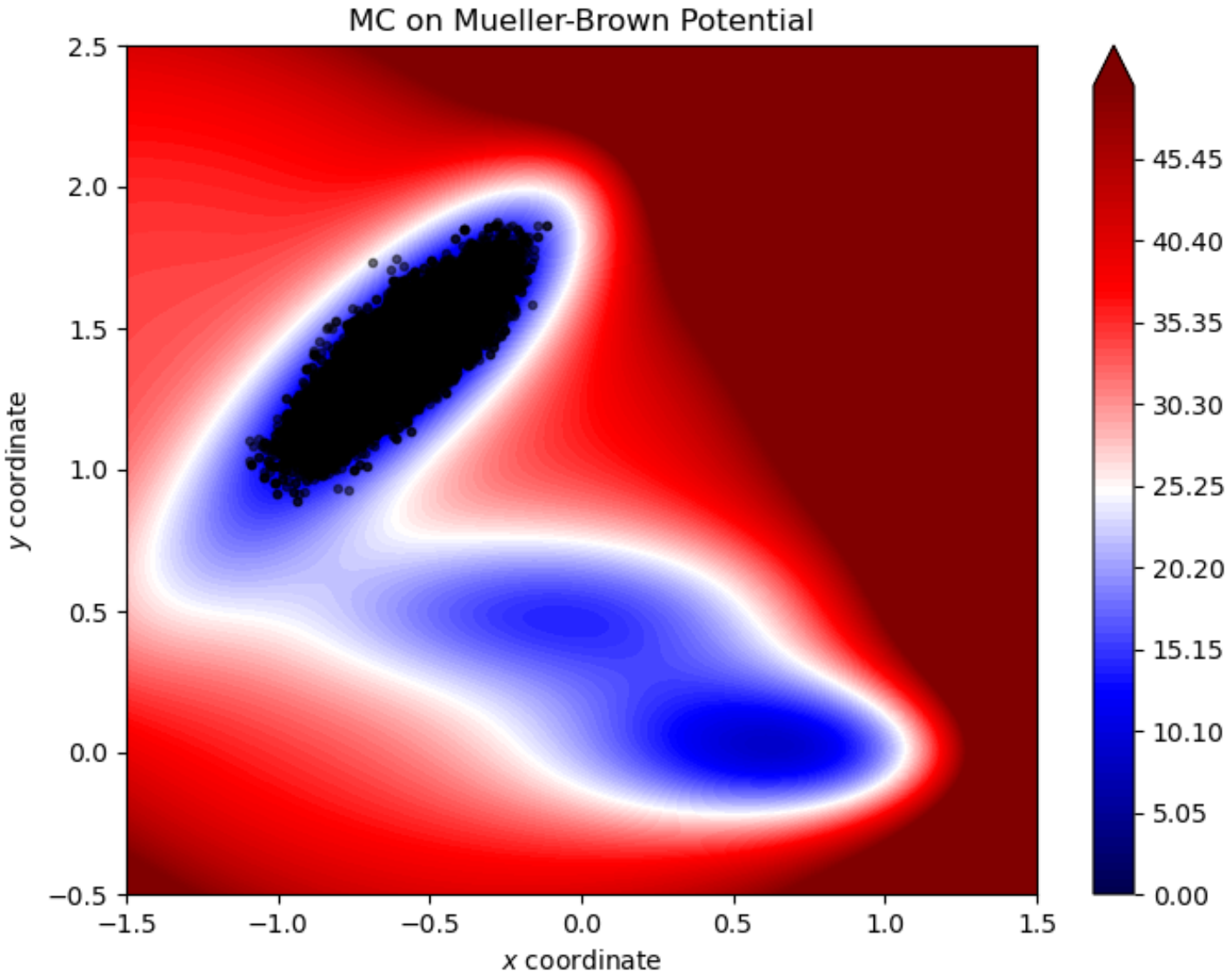
Particle on the Müller-Brown Potential

```
python mc-potential.py --n_samples 10000 --temp 1.0
```



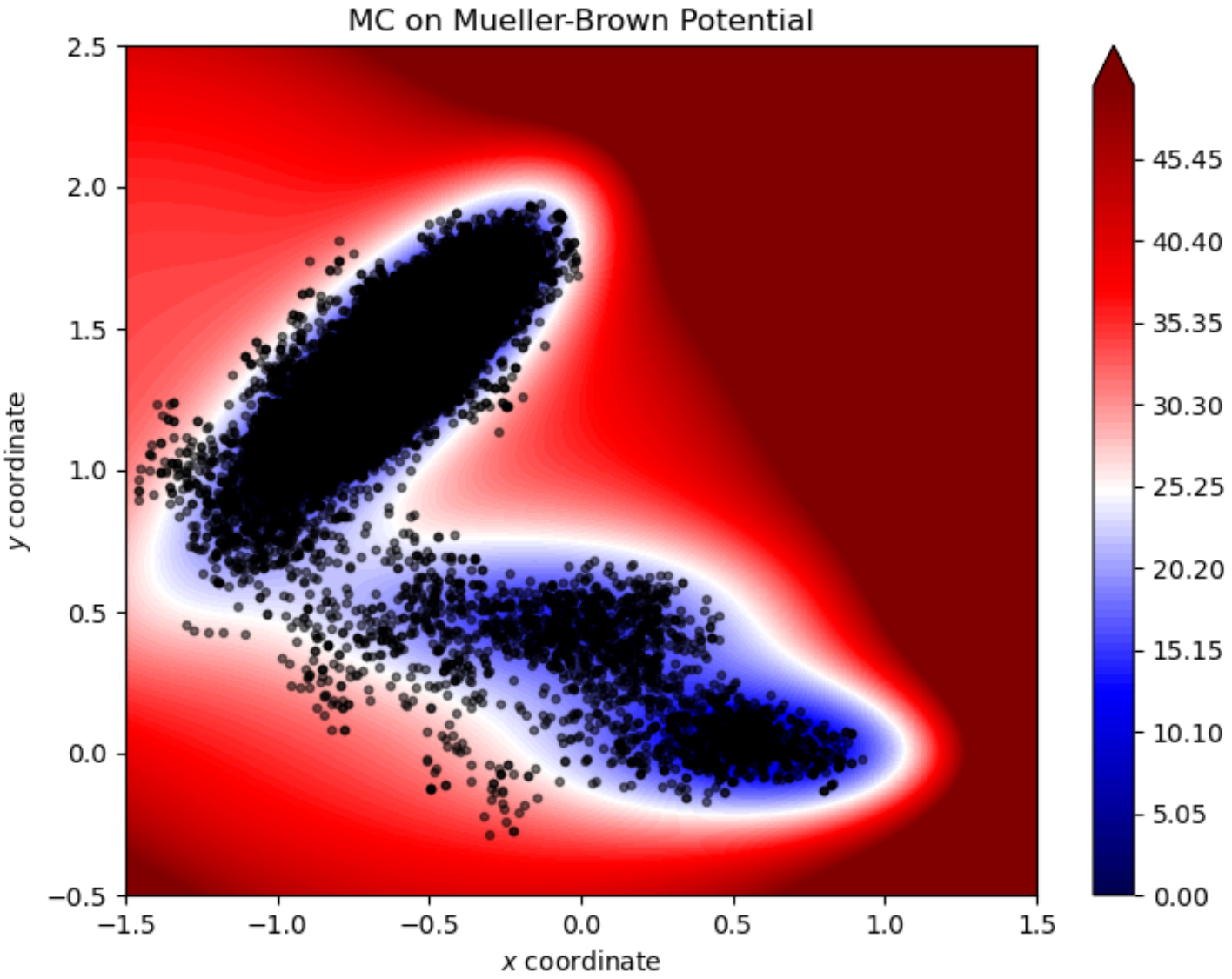
Particle on the Müller-Brown Potential

```
python mc-potential.py --n_samples 10000 --temp 2.0
```



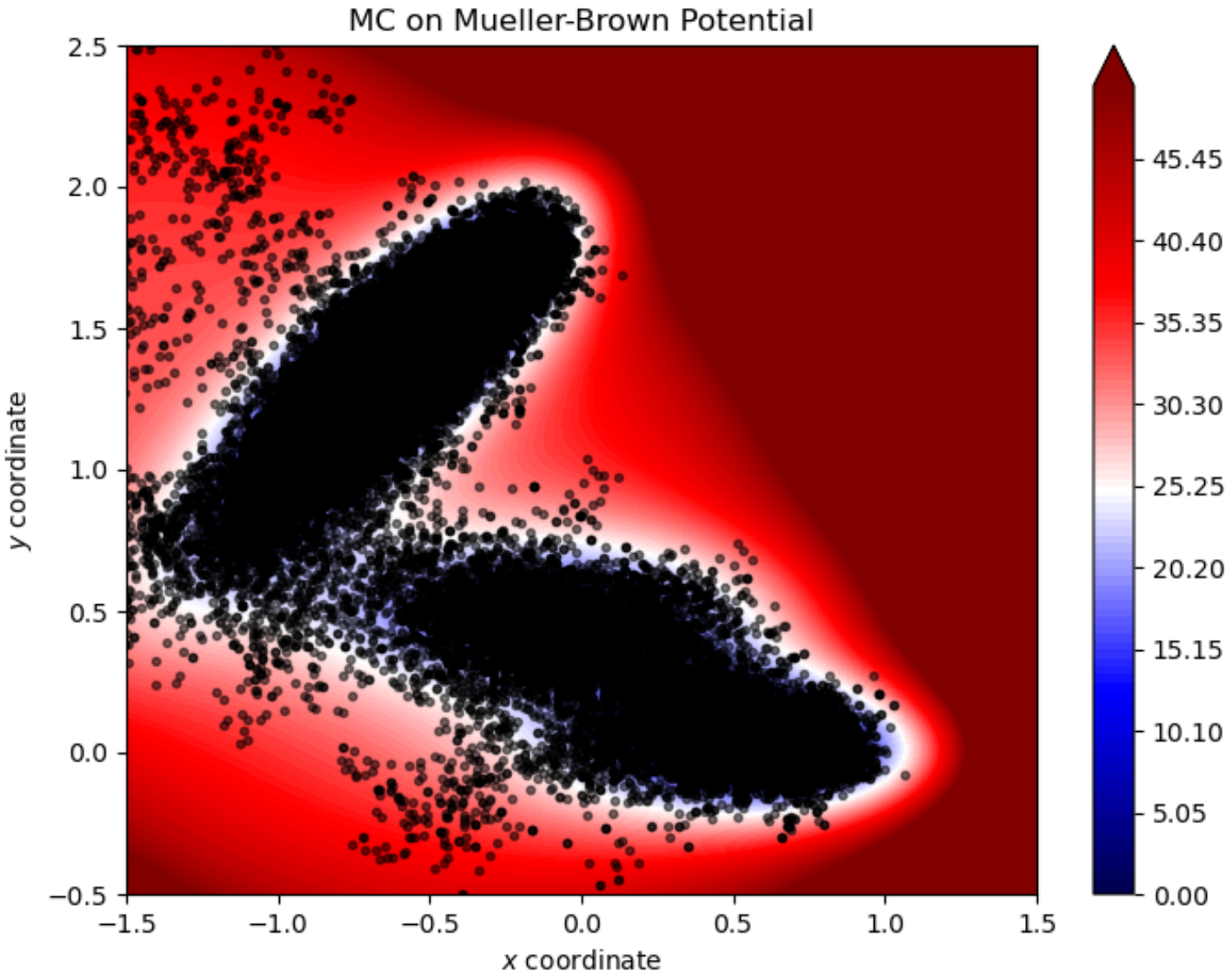
Particle on the Müller-Brown Potential

```
python mc-potential.py --n_samples 10000 --temp 3.0
```



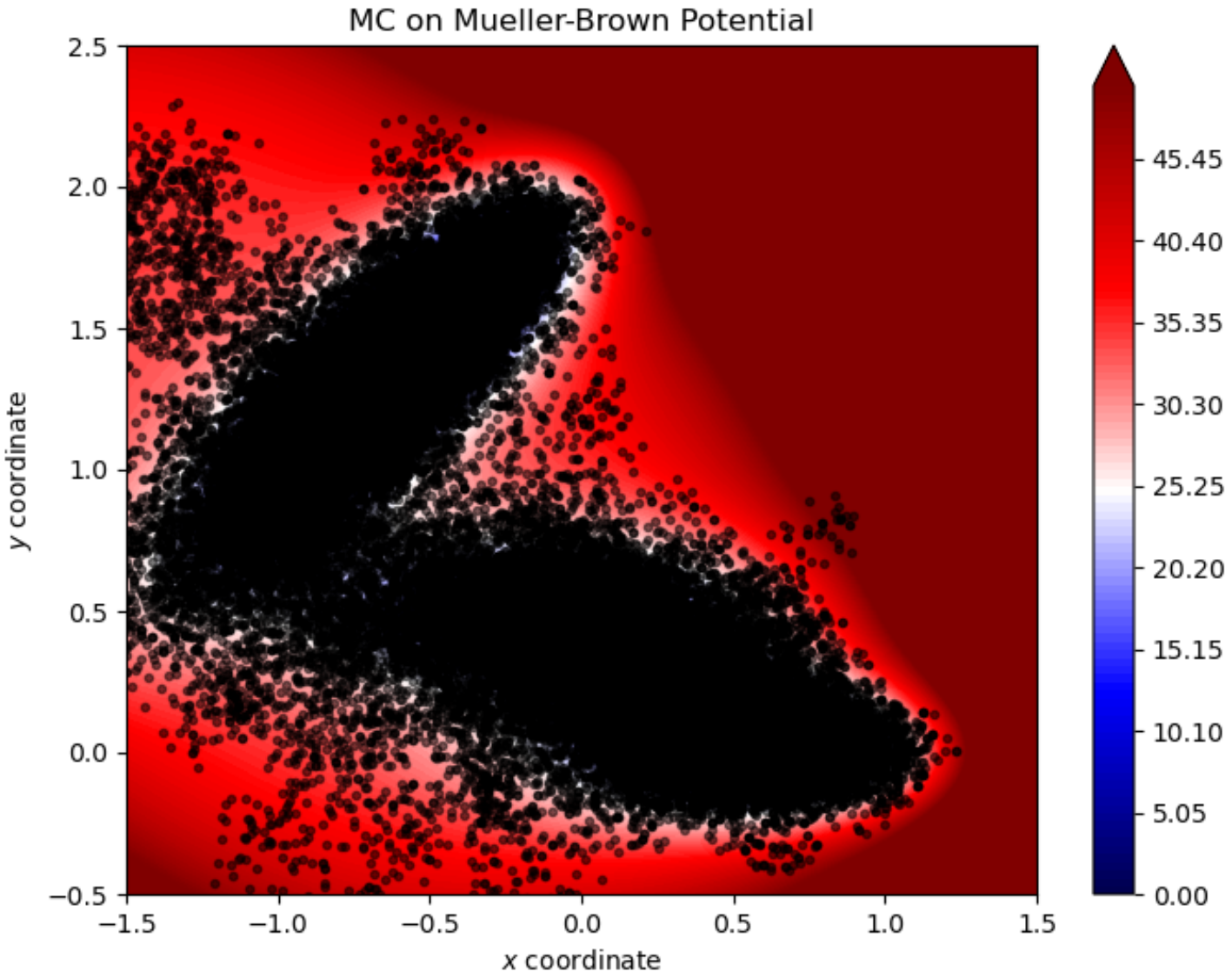
Particle on the Müller-Brown Potential

```
python mc-potential.py --n_samples 10000 --temp 4.0
```



Particle on the Müller-Brown Potential

```
python mc-potential.py --n_samples 10000 --temp 5.0
```



Replica Exchange Monte Carlo

- Barrier crossing are frequently rarely encountered during a simulation.
- For a barrier height of 15 kJ/mol at 300 K, the Boltzmann factor is approximately 3×10^{-3}; for 30 kJ/mol is 6×10^{-6}.

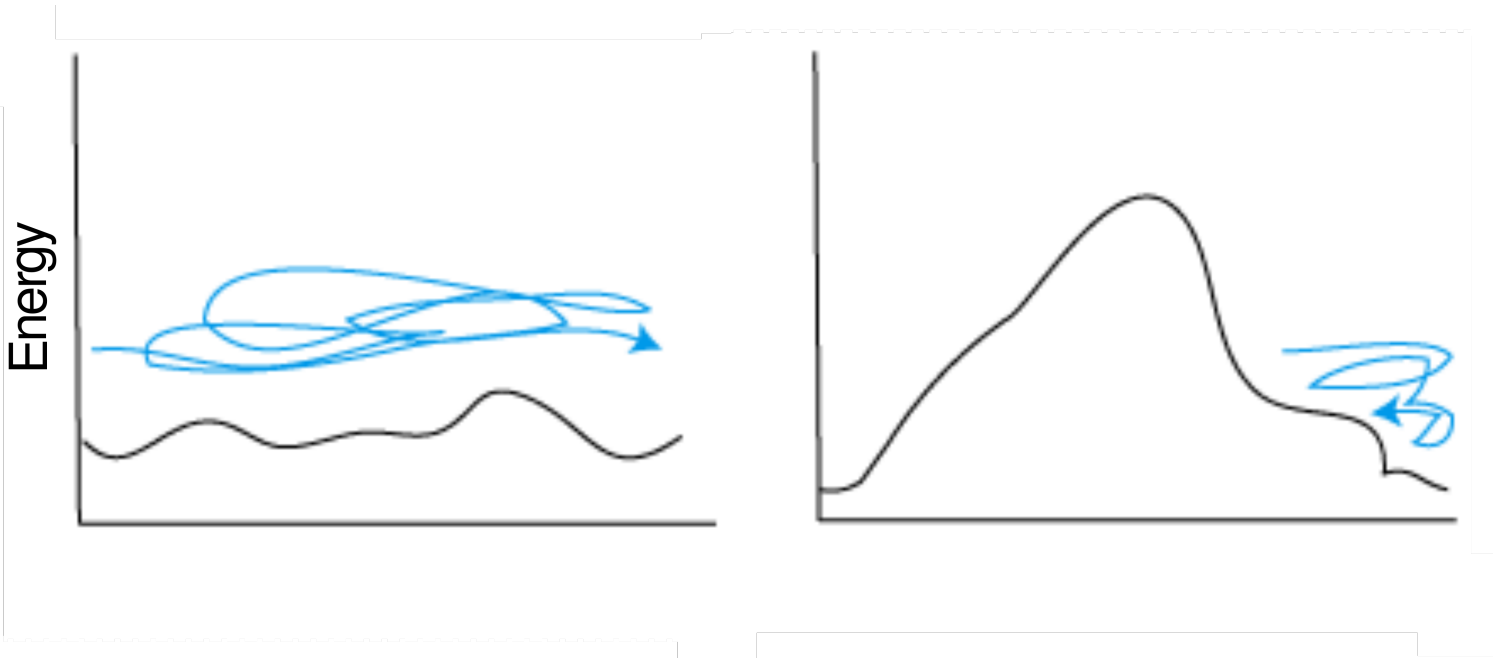


Figure 19: Barrier crossing showing a trajectory in the case of a low and a high energy barrier, respectively.

Parallel Tempering[5]

- Temperature is used as the control variable, and different temperatures are assigned to the *replicas*.
- In the parallel tempering scheme, a set of temperatures $T_1, \ldots, T_M$ such that:

$$T_1 < T_2 < \cdots < T_M \tag{63}$$

  are assigned to the M replicas, where T_1 is the temperature T of the canonical distribution.
- The high-temperature replicas easily cross potential energy barriers.
- We attempt exchanges between the neighboring replicas.

[5] Marinari and Parisi. *Simulated tempering: A new Monte Carlo scheme* Europhys. Lett. 19 (1992)

Parallel Tempering

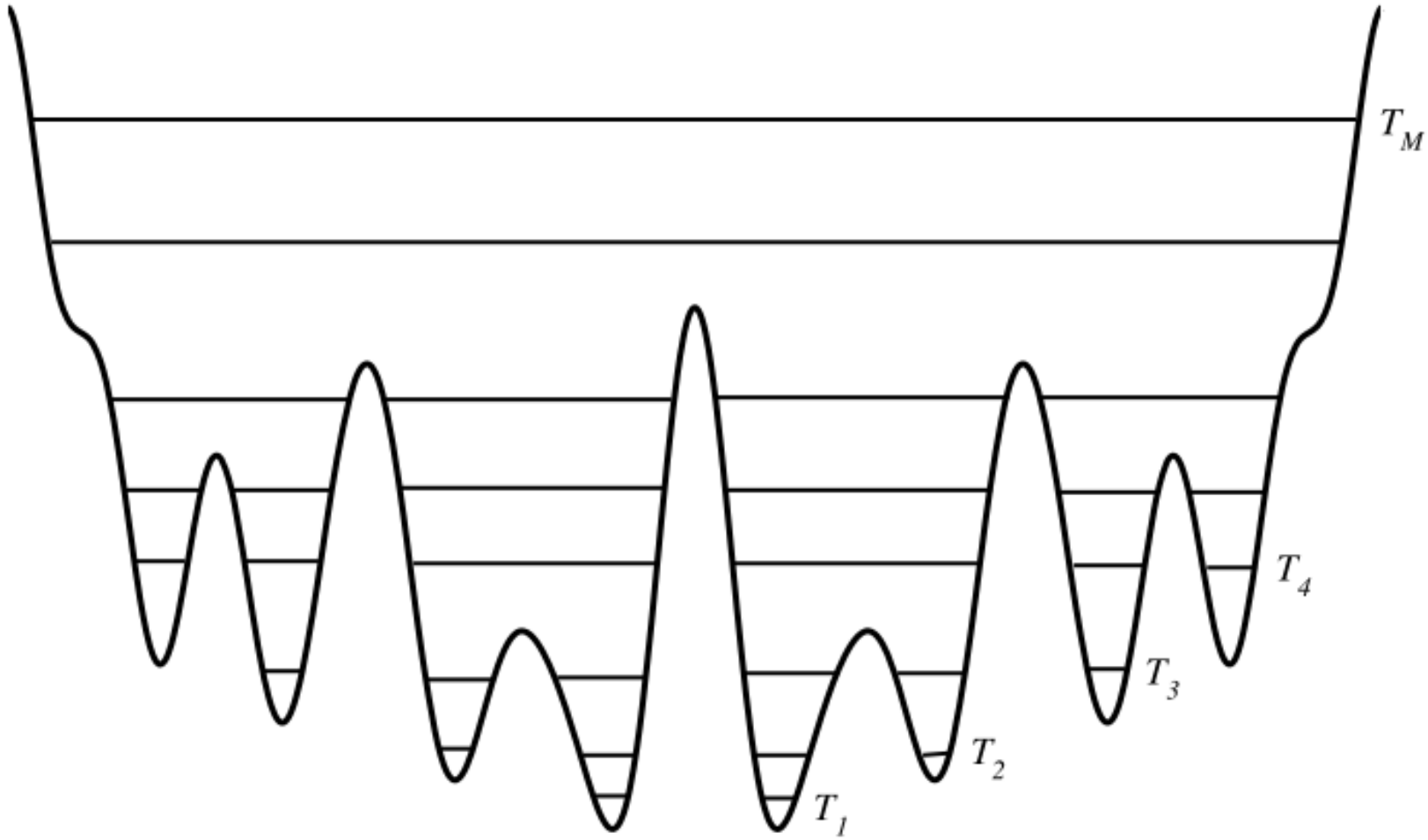


Figure 20: Schematic of the parallel-tempering replica exchange Monte Carlo.[6]

[6] Source: M. E. Tuckerman, *Statistical Mechanics: Theory and Simulation*, Oxford University Press (2016)

Parallel Tempering

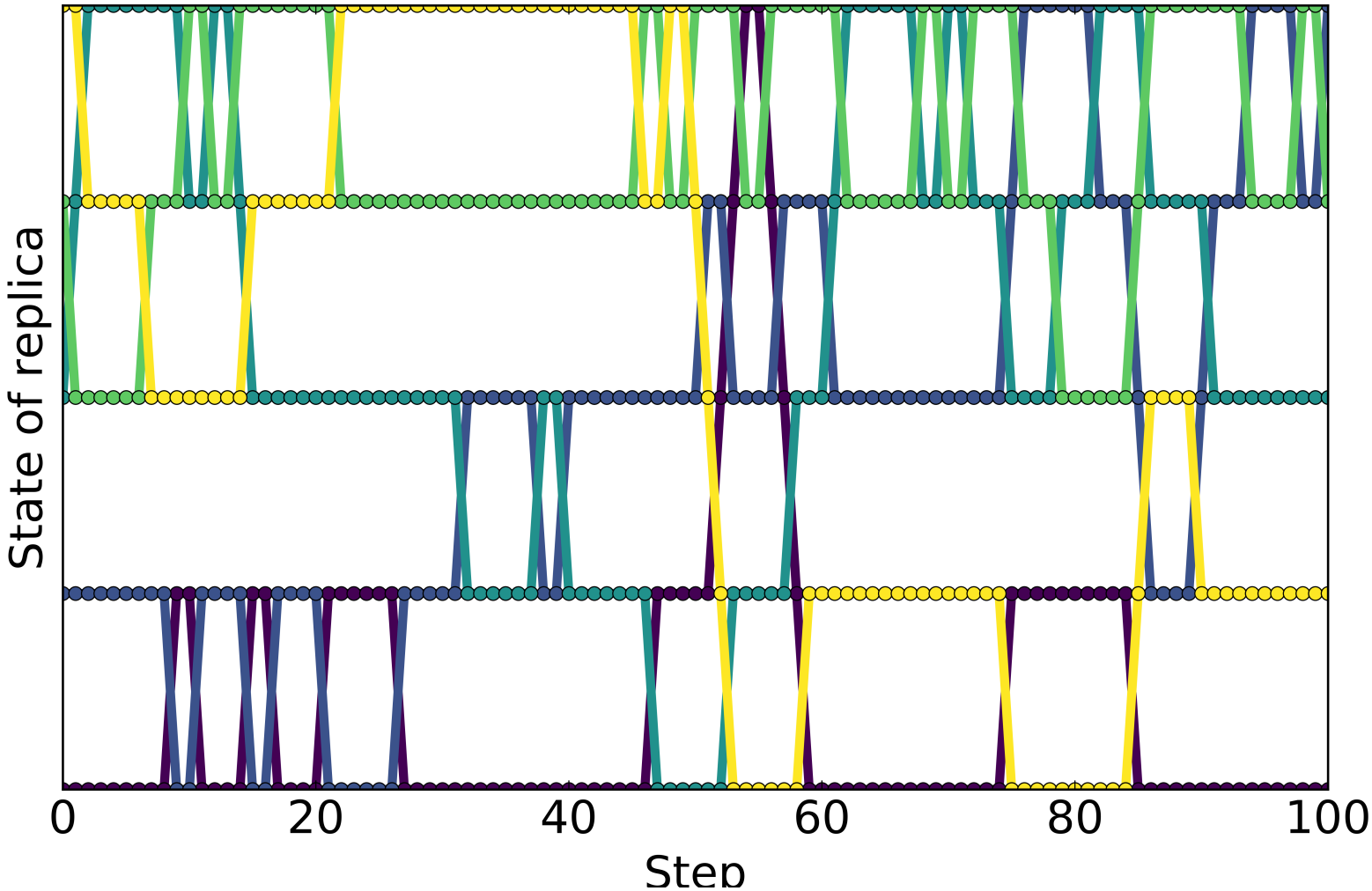


Figure 21: An example of a parallel tempering trajectory.[7]

[7] Source: https://coulomb.umontpellier.fr/perso/daniele.coslovich/pt/

Paralell Tempering

- Let $\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(M)}$ be the complete configurations of the M replicas, i.e., $\mathbf{r}^{(K)} \equiv \left(\mathbf{r}_1^{(K)}, \ldots, \mathbf{r}_N^{(K)}\right)$.
- The replicas are independent, so the total probability distribution $F\left(\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(M)}\right)$ is:

$$F\left(\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(M)}\right) = \prod_{K=1}^{M} f_K\left(\mathbf{r}^{(K)}\right), \tag{64}$$

where:

$$f_K\left(\mathbf{r}^{(K)}\right) = \frac{1}{Z} \mathrm{e}^{-\beta_K U\left(\mathbf{r}^{(K)}\right)}, \tag{65}$$

in which $\beta_K = \left(k_{\mathrm{B}} T_K\right)^{-1}$.

Parallel Tempering

- Periodically, a neighboring pair of replicas K and $K+1$ is selected, and an attempted switch is made with probability:

$$A_s = A\left[(\mathbf{r}^{(K+1)}, \mathbf{r}^{(K)})|(\mathbf{r}^{(K)}, \mathbf{r}^{(K+1)})\right] \tag{66}$$

$$= \min\left[1, \frac{f_K(\mathbf{r}^{(K+1)})}{f_K(\mathbf{r}^{(K)})} \cdot \frac{f_{K+1}(\mathbf{r}^{(K)})}{f_{K+1}(\mathbf{r}^{(K+1)})}\right] \tag{67}$$

$$= \min\left[1, \mathrm{e}^{-\Delta_{K,K+1}}\right], \tag{68}$$

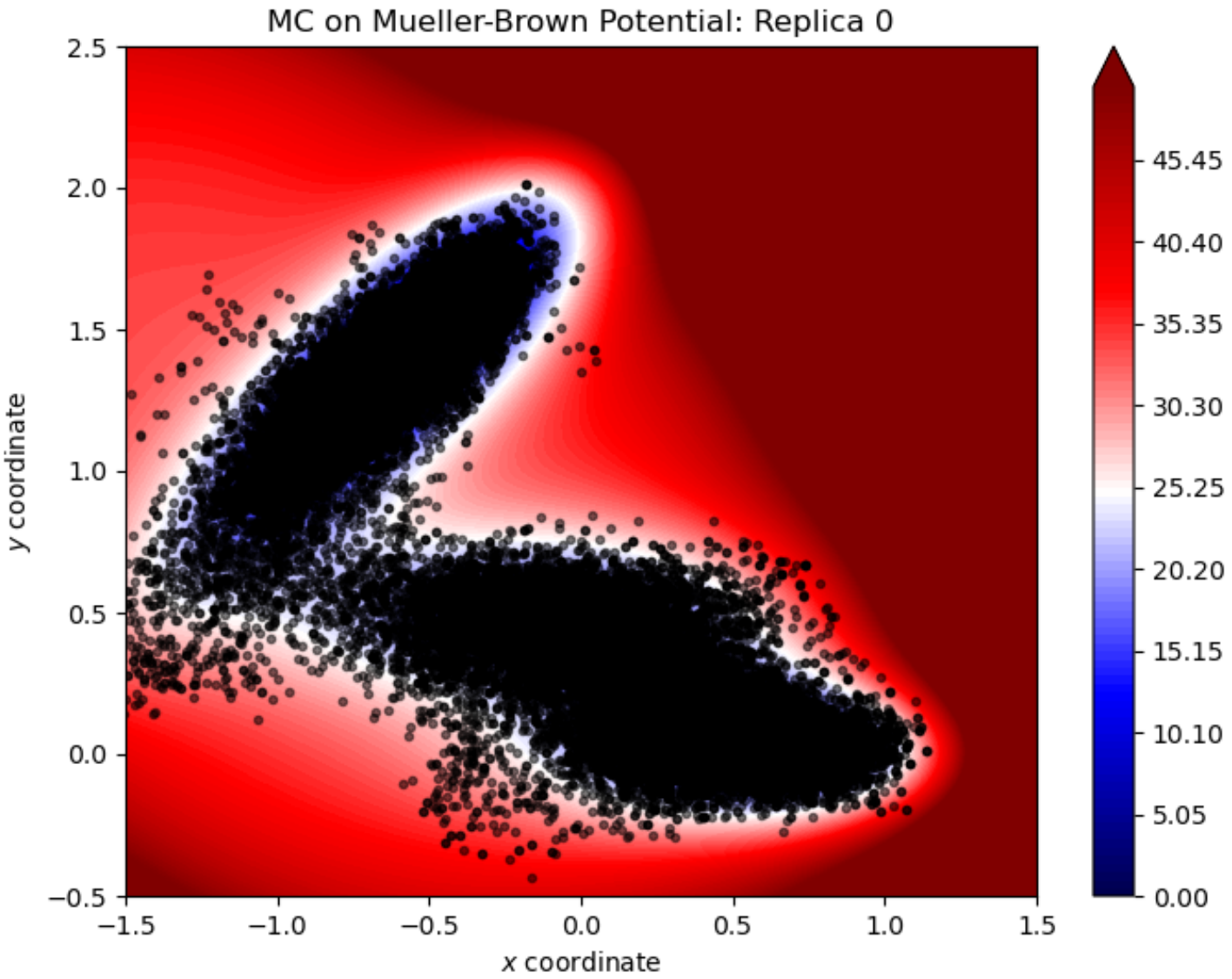
where:

$$\Delta_{K,K+1} = (\beta_K - \beta_{K+1})\left[U(\mathbf{r}^{(K)}) - U(\mathbf{r}^{(K+1)})\right] \tag{69}$$

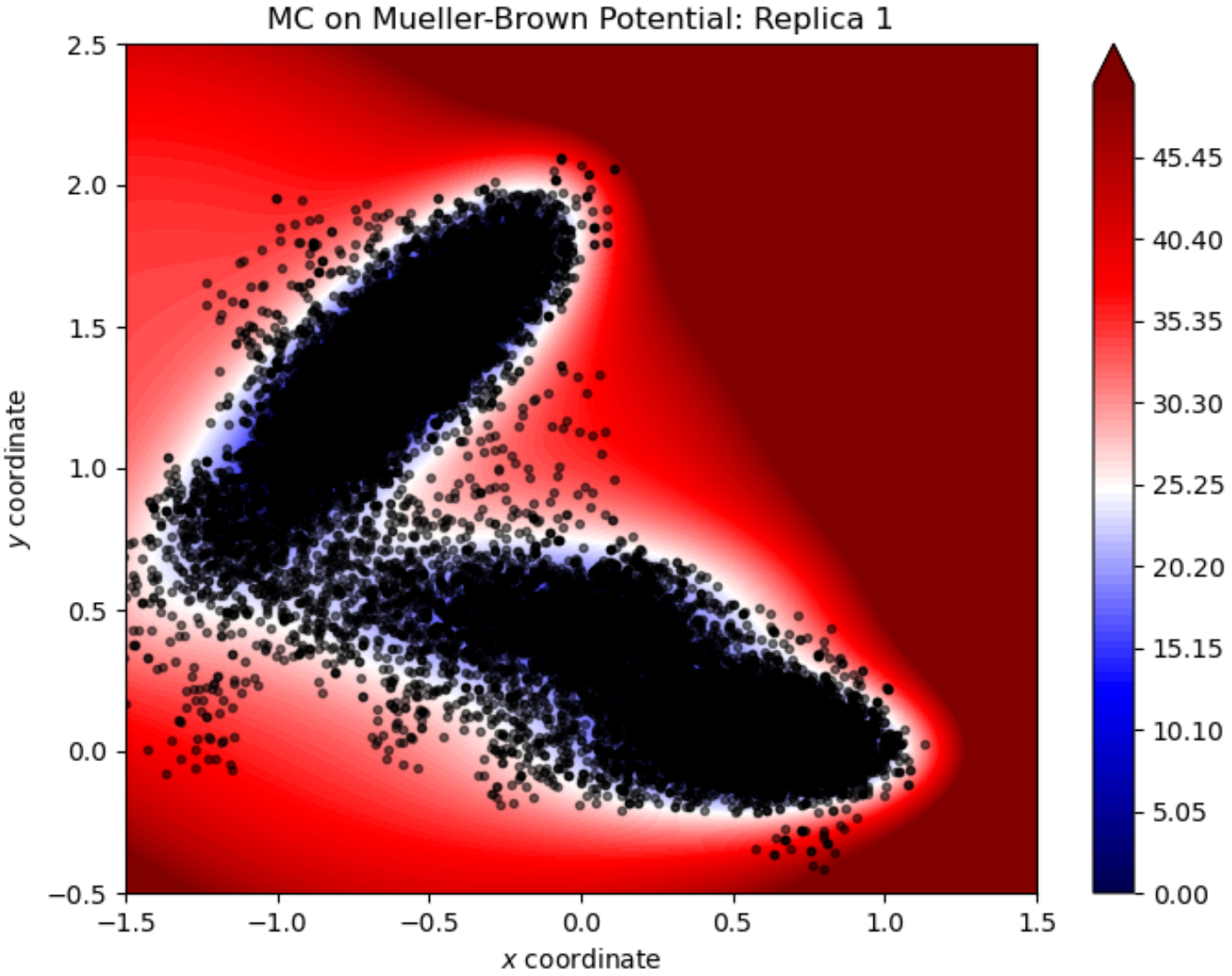
Particle on the Müller-Brown Potential

```
python mc-pt-potential.py --n_samples 10000
                          --temp 1,2,3,4,5
                          --n_replicas 5
                          --exchange 1000
```

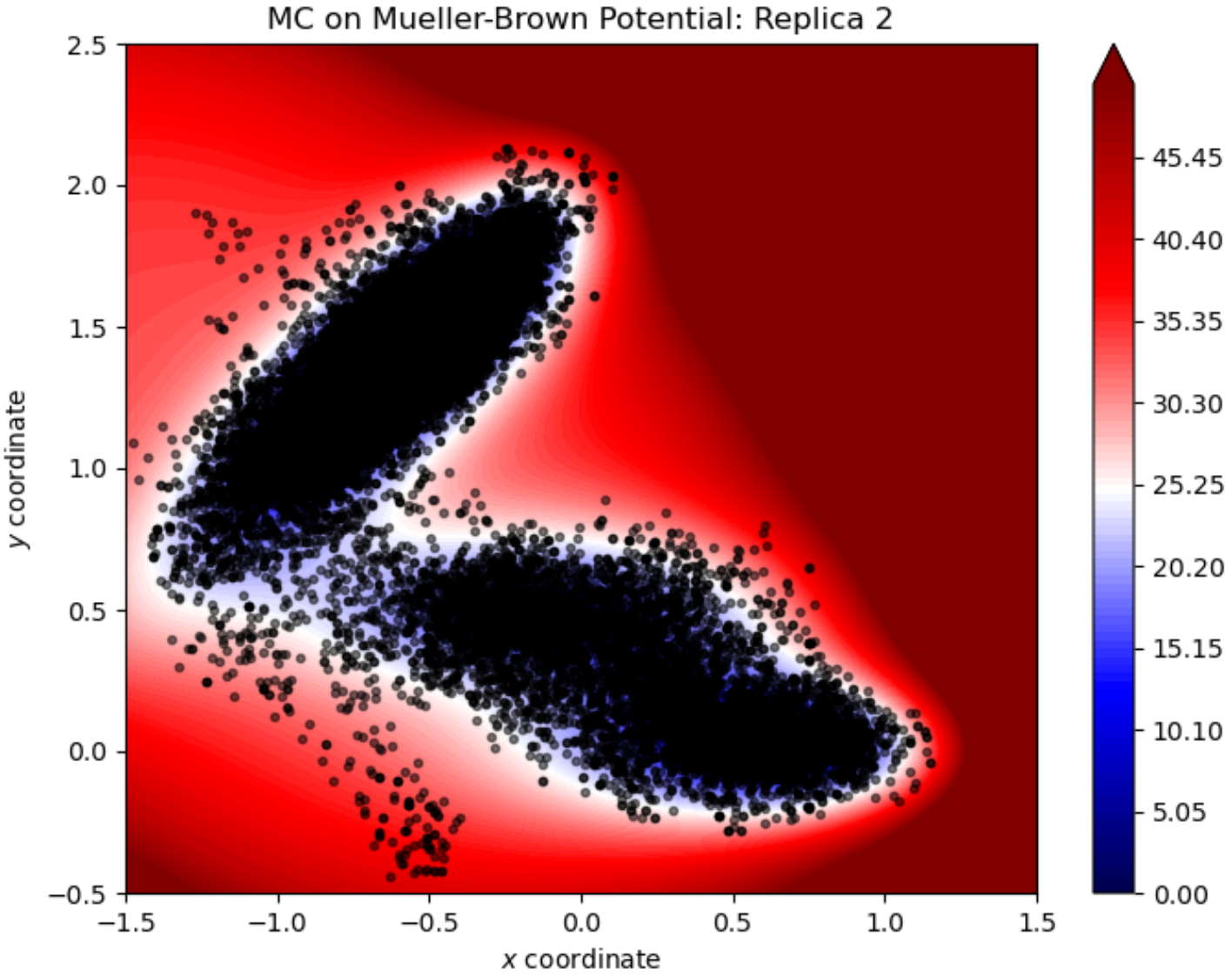
Particle on the Müller-Brown Potential



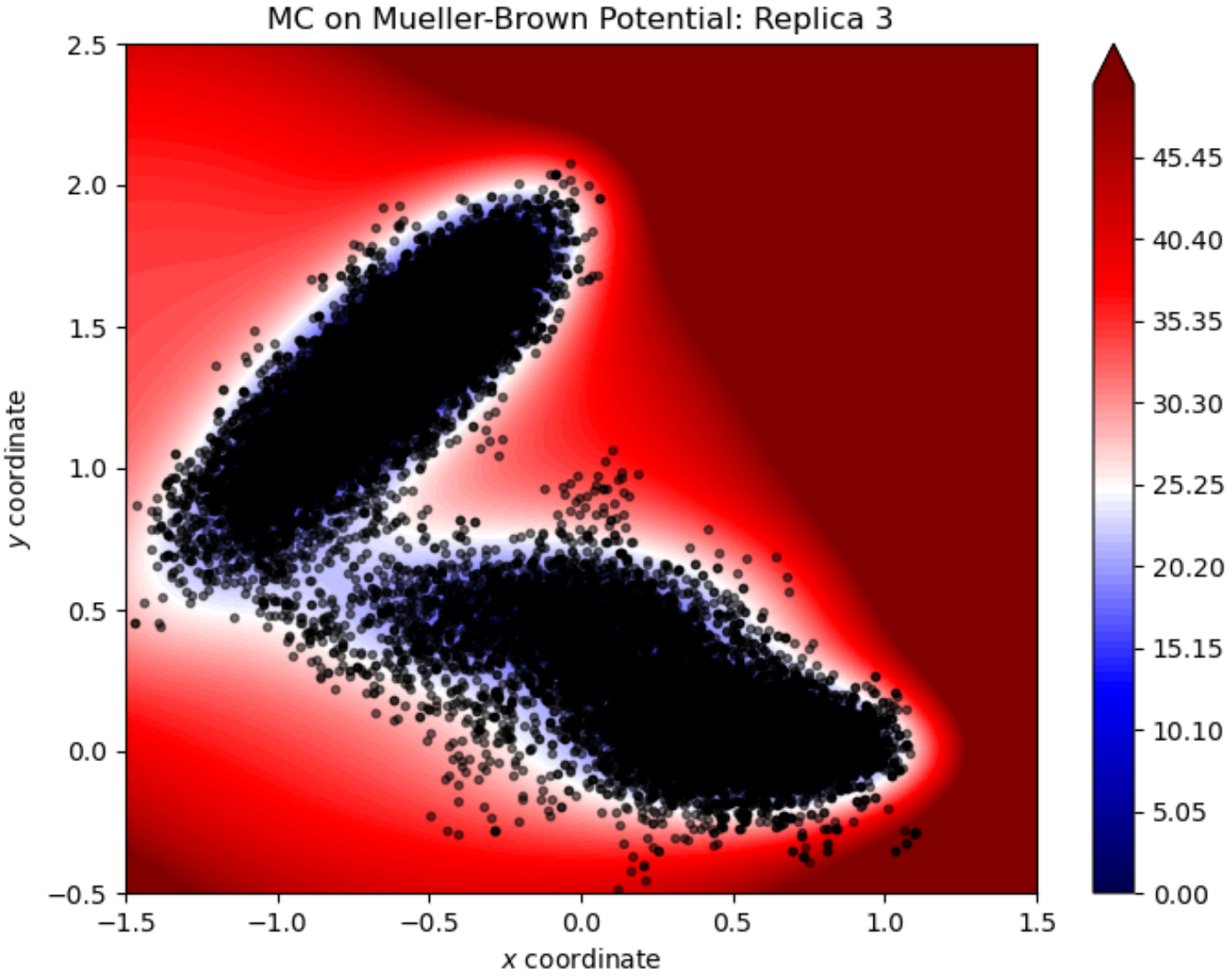
Particle on the Müller-Brown Potential



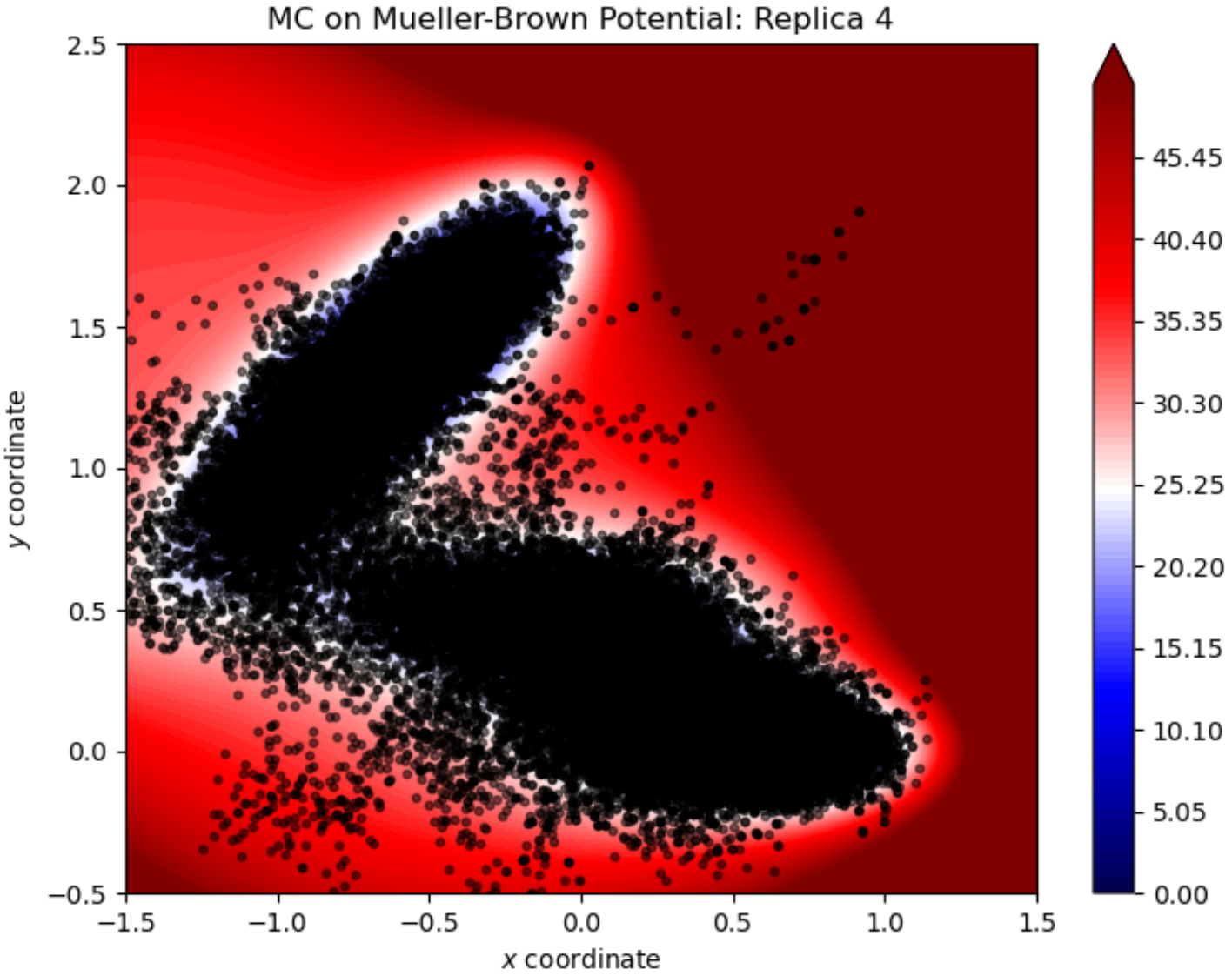
Particle on the Müller-Brown Potential



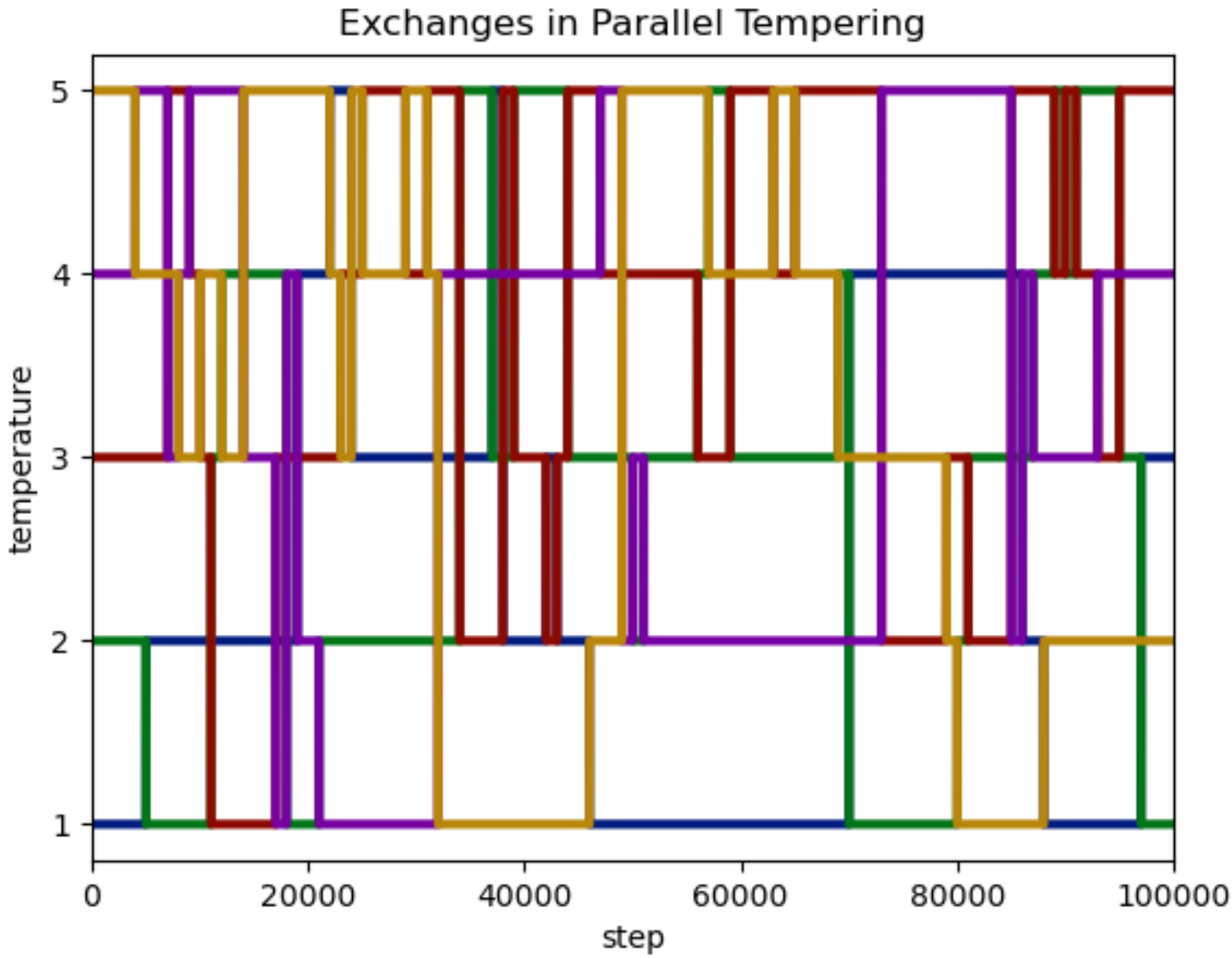
Particle on the Müller-Brown Potential



Particle on the Müller-Brown Potential



Particle on the Müller-Brown Potential



Wang-Landau Sampling

- Recall that the canonical partition function $Z(N, V, T)$ can be expressed in terms of the microcanonical partition function $\Omega(N, V, E)$:

$$Z(N, V, T) = \frac{1}{E_0} \int_0^\infty \mathrm{d}E \, \mathrm{e}^{-\beta E} \Omega(N, V, E), \tag{70}$$

  where E_0 is an arbitrary reference energy.

- Setting $E_0 = 1$ and dropping fixed V and N, we have:

$$Z(\beta) = \int_0^\infty \mathrm{d}E \, \mathrm{e}^{-\beta E} \Omega(E). \tag{71}$$

- Eq. 71 suggests a procedure to compute the canonical partition function, and thus, all thermodynamic quantities.

Wang-Landau Sampling

Wang-Landau Algorithm[8]

1. Set $\Omega(E)$ for all values of E.
2. Attempt a trial move from E to E' with the acceptance probability given by:

$$A(E'|E) = \min\left[1, \frac{\Omega(E)}{\Omega(E')}\right]. \tag{72}$$

3. Modify the bin such that $\Omega(E) \to \Omega(E)f$, where $f > 1$.
4. Accumulate the histogram of energy, $h(E) \to h(E) + 1$.
5. If $h(E)$ is flat enought, then $f \to \sqrt{f}$.
6. If not converged, move to 2; else return $\Omega(E)$.

[8] Numerical implementations require that $\Omega(E)$ be discretized into a number of energy bins.

Wang-Landau Sampling

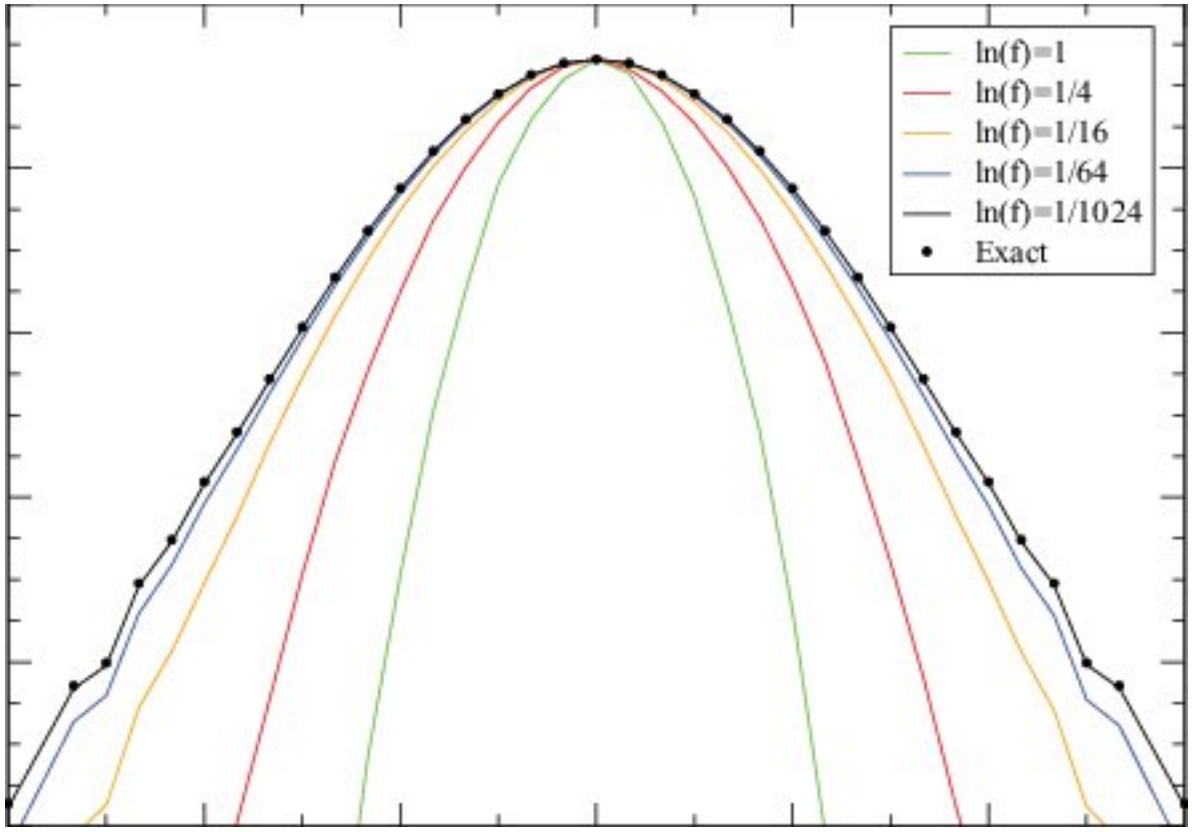


Figure 22: Convergence of the Wang-Landau algorithm.[9]

[9] Source: Brown, Odbadrakh, Nicholson, and Eisenbach, *Convergence for the Wang-Landau density of states*, Phys. Rev. E 84, 065702(R) (2011)

Molecular Dynamics

Molecular Dynamics

There are three main ingredients in molecular dynamics (MD):

- The algorithm to integrate the equations of motion,
- The model describing the interparticle interactions,
- The calculation of forces and energies from the model.

Verlet Algorithm

- The simplest approach to obtain a numerical integration scheme is to use a Taylor series:

$$\mathbf{r}_i(t+\Delta t) \approx \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t) + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t), \tag{73}$$

and

$$\mathbf{r}_i(t-\Delta t) \approx \mathbf{r}_i(t) - \Delta t \mathbf{v}_i(t) + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t). \tag{74}$$

- Adding Eq. 73 to Eq. 74, we get a velocity-independent Verlet algorithm:

$$\mathbf{r}_i(t+\Delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t-\Delta t) + \frac{\Delta t^2}{m_i}\mathbf{F}_i(t). \tag{75}$$

Velocity Verlet Algorithm

- Consider a shift in time in comparision to Eq. 73:

$$\mathbf{r}_i(t) \approx \mathbf{r}_i(t+\Delta t) - \Delta t \mathbf{v}_i(t+\Delta t) + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t+\Delta t), \quad (76)$$

  which after some rearrangements takes the following form:

$$\mathbf{v}_i(t+\Delta t) = \mathbf{v}_i(t) + \frac{\Delta t}{2m_i}\left[\mathbf{F}_i(t) + \mathbf{F}_i(t+\Delta t)\right]. \quad (77)$$

- This scheme allows to use both evolution of the positions and velocities.

Initial Conditions

- Initial coordinates.
- Initial velocities are sampled from a Maxwell-Boltzmann distribution, taking care to ensure that the sampled velocities are consistent with any constraints imposed on the system:

$$f(v) = \frac{m}{2\pi k_{\mathrm{B}} T}^{1/2} \mathrm{e}^{-mv^2/2k_{\mathrm{B}}T}, \tag{78}$$

  where v is the velocity of a particle of mass m at temperature T.
- An example of a Gaussian probability distribution.

Potential Energy Model

- Commonly used model of potential energy in MD for biological macromolecules is the following:

$$U(\mathbf{r}) = \sum_{\text{bond}} \frac{1}{2} K_{\text{bond}} (r - r_0)^2 \tag{79}$$

$$+ \sum_{\text{angle}} \frac{1}{2} K_{\text{bends}} (\theta - \theta_0)^2 \tag{80}$$

$$+ \sum_{\text{dihedral}} \sum_{n=0}^{6} A_n [1 + \cos(C_n \phi + \delta_n)] \tag{81}$$

$$+ \sum_{i,j \in \text{non-bonded}} \left(4\epsilon_{ij} \left[\frac{\sigma_{ij}}{r_{ij}}^{12} - \frac{\sigma_{ij}}{r_{ij}}^{6} \right] + \frac{q_i q_j}{r_{ij}} \right). \tag{82}$$

- Force is given by $\mathbf{F} = -\partial U / \partial \mathbf{r}$.
- Called *force field* in MD.

Potential Energy Model

- Force field provides constraints into the systems.

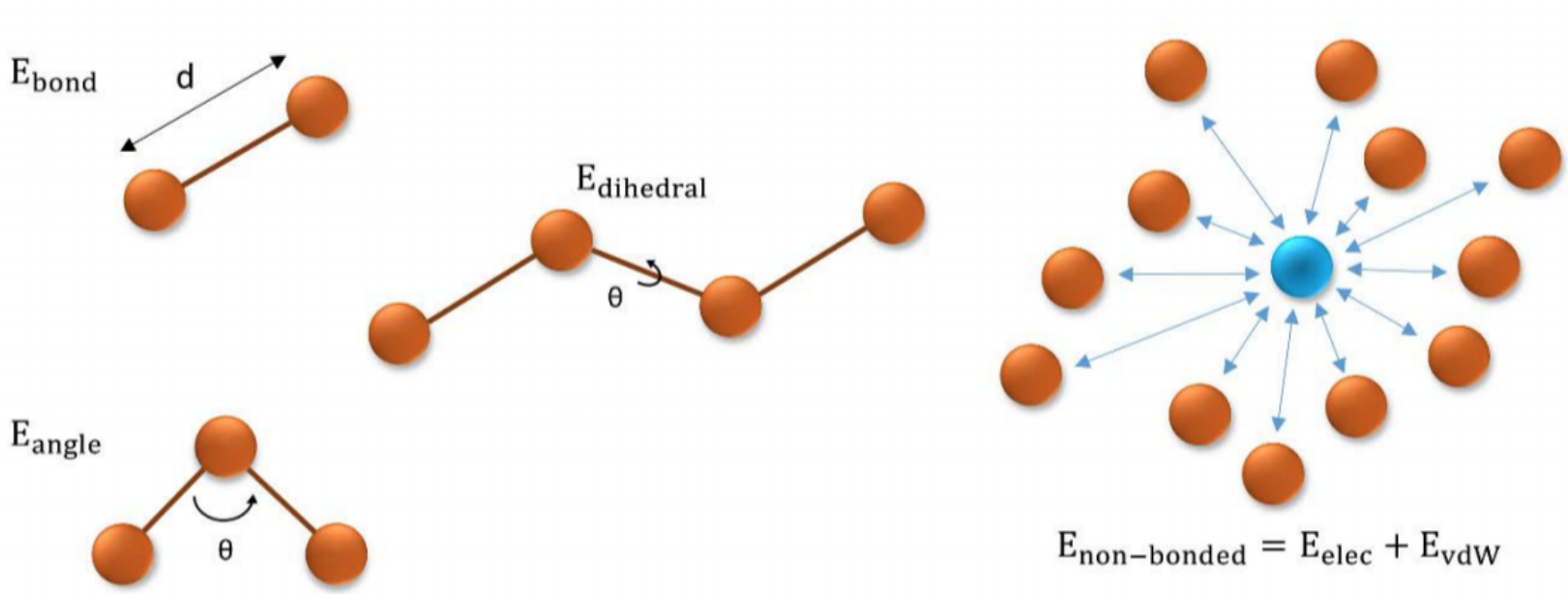
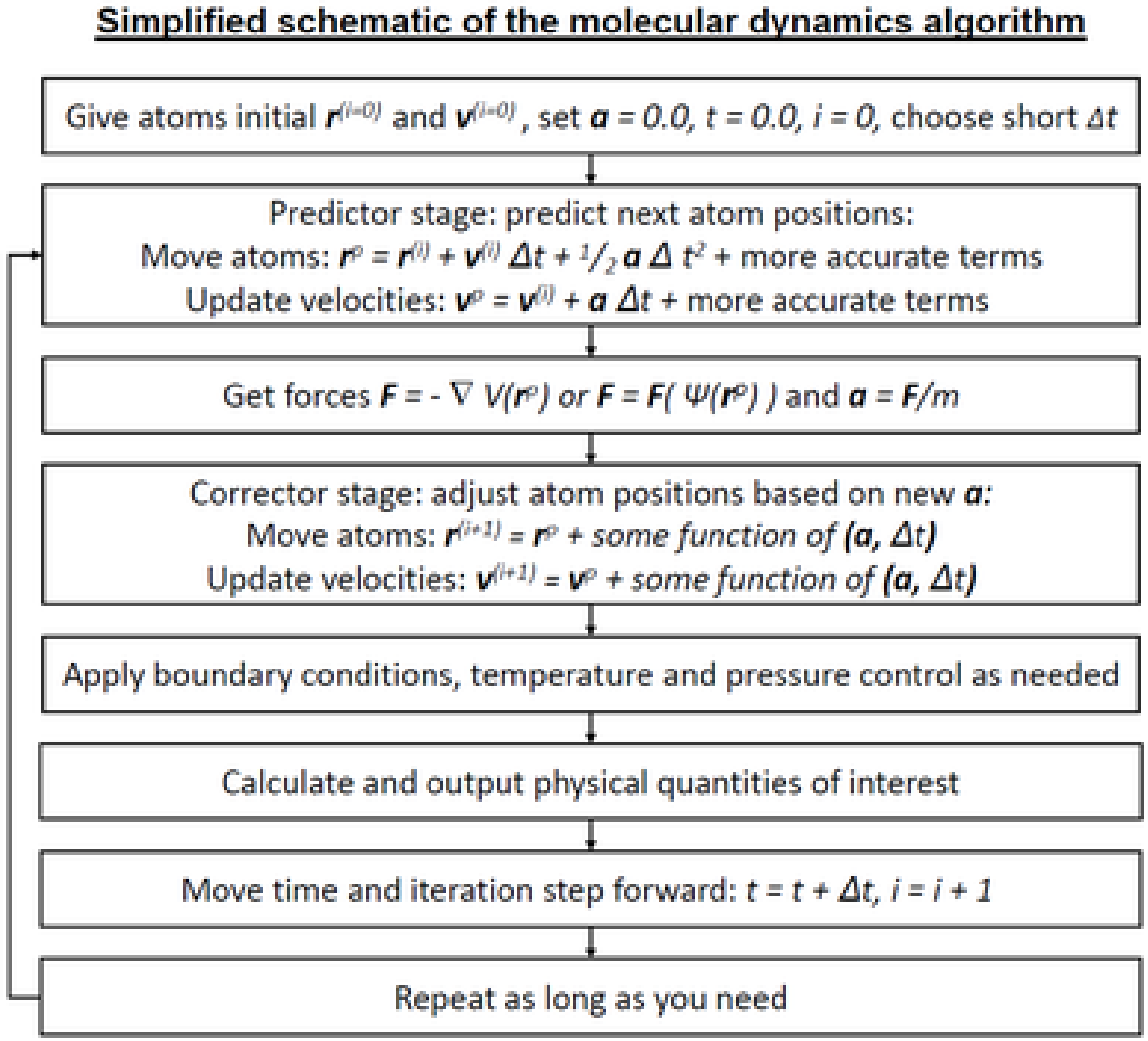


Figure 23: Force field contributions.

- Many force fields are available: Amber, CHARMM, Gromos.

General Scheme in MD

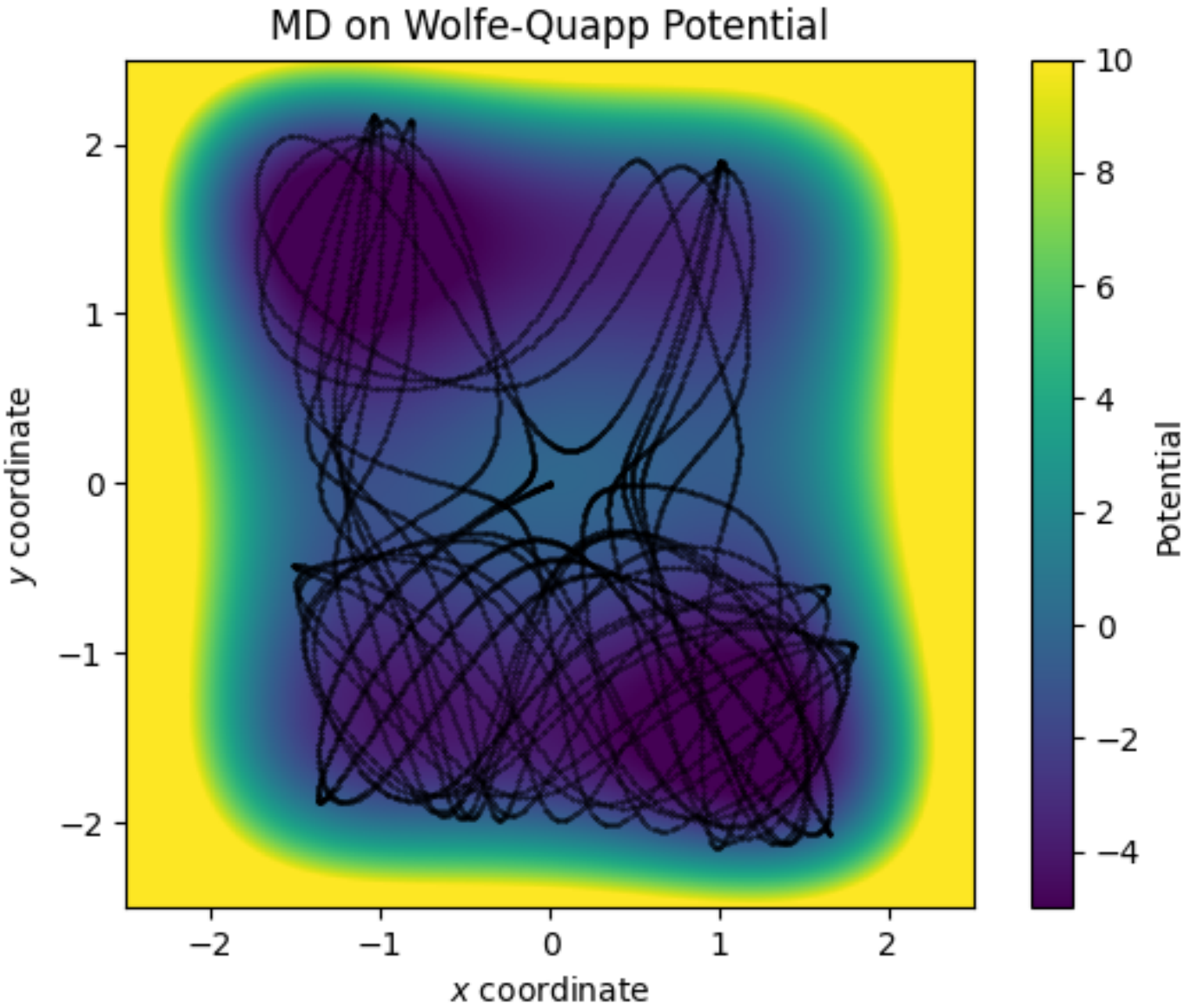


Particle on the Wolfe-Quapp Potential

```
for i in range(0, args.n_samples-1):
  position[i+1] = position[i]
                + velocity[i]*dt
                + 0.5*(force/m)*dt**2
  p, f = U.eval(position[i+1])
  velocity[i+1] = velocity[i]
                + 0.5/m*dt*(force+f)
  p_energy[i+1] = p
  force = f
```

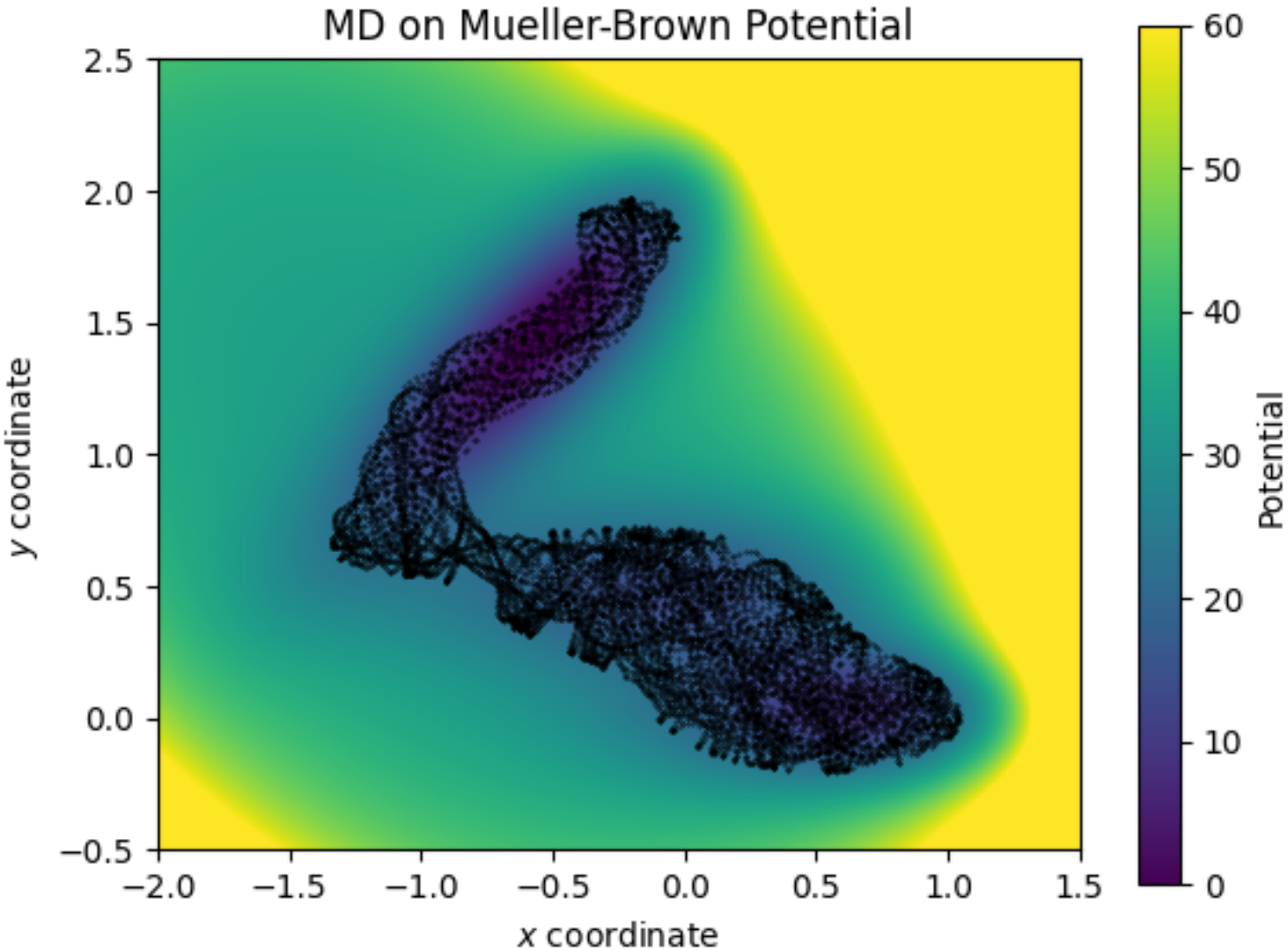
Particle on the Wolfe-Quapp Potential

```
python MD-WolfeQuapp.py --n_samples 10000 --dt 0.01 --mass 1
```

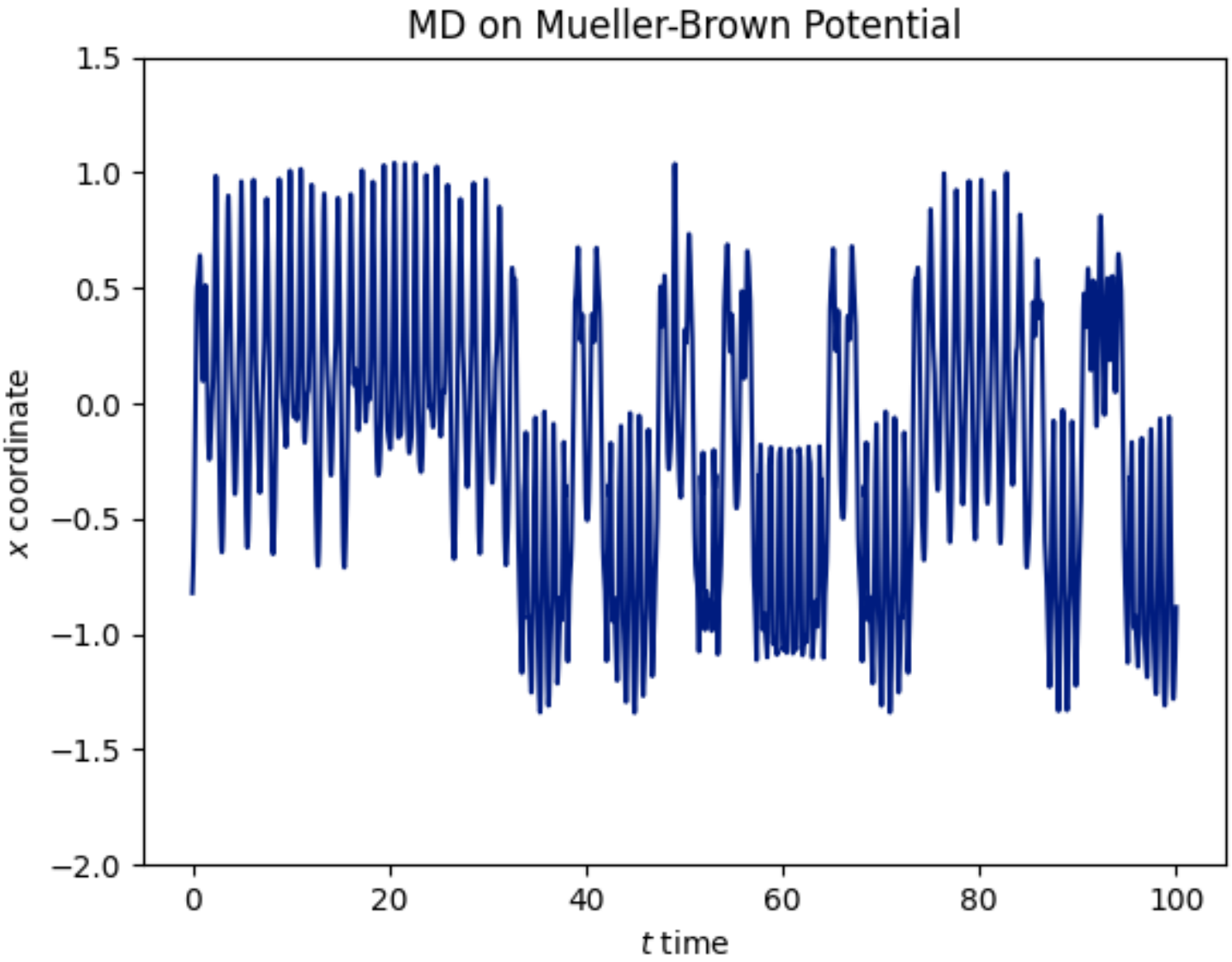


Particle on the Müller-Brown Potential

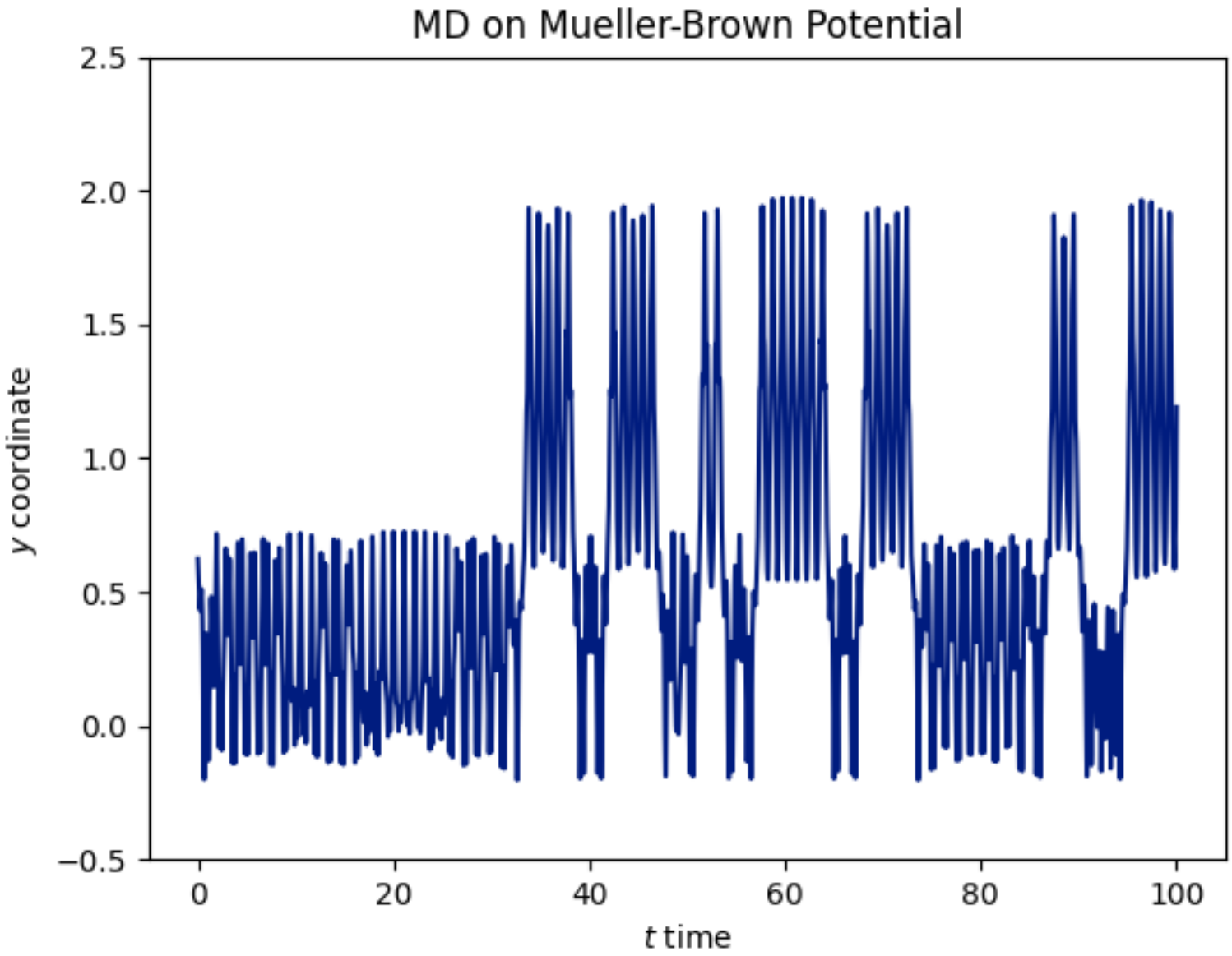
```
python MD-MuellerBrown.py --n_samples 10000 --dt 0.01 --mass 1
```



Particle on the Müller-Brown Potential



Particle on the Müller-Brown Potential



Alanine Dipeptide

- Benchmark system for MD methods.

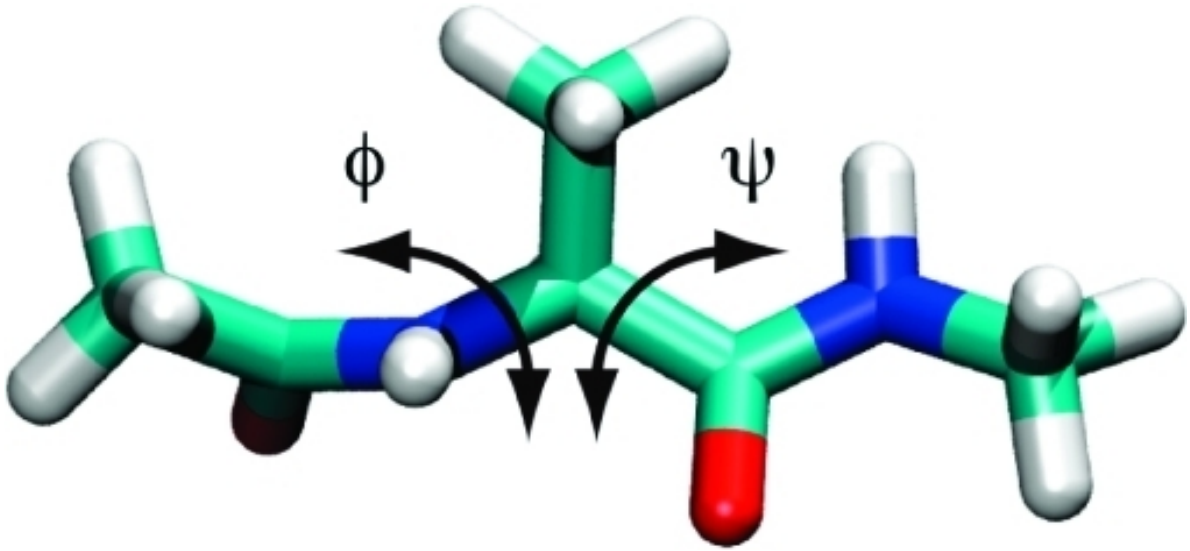


Figure 24: Alanine dipeptide with the dihedral angles.

- Scatter plot of the dihedral angles in called the Ramachandran plot.

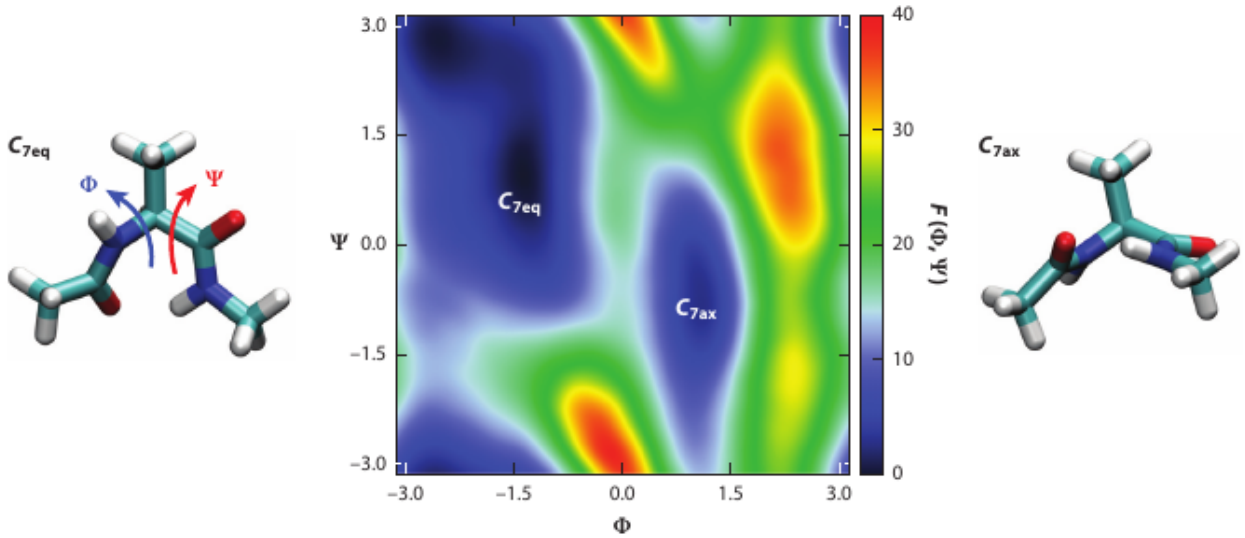
Alanine Dipeptide

- An example of the `.gro` file with alanine dipeptide.

```
 22
1ACE    HH31    1   1.075   1.111   1.585
1ACE     CH3    2   1.129   1.065   1.502
1ACE    HH32    3   1.070   0.991   1.448
1ACE    HH33    4   1.215   1.011   1.540
1ACE       C    5   1.162   1.167   1.395
1ACE       O    6   1.140   1.286   1.416
2ALA       N    7   1.230   1.118   1.289
2ALA       H    8   1.248   1.019   1.293
2ALA      CA    9   1.260   1.194   1.165
2ALA      HA   10   1.304   1.125   1.093
2ALA      CB   11   1.126   1.224   1.093
2ALA     HB1   12   1.078   1.312   1.135
2ALA     HB2   13   1.147   1.247   0.989
2ALA     HB3   14   1.073   1.129   1.094
2ALA       C   15   1.357   1.317   1.171
2ALA       O   16   1.392   1.372   1.068
3NME       N   17   1.404   1.353   1.294
3NME       H   18   1.350   1.322   1.373
3NME     CH3   19   1.517   1.445   1.315
3NME    HH31   20   1.495   1.544   1.278
3NME    HH32   21   1.546   1.457   1.420
3NME    HH33   22   1.606   1.405   1.267
```

Alanine Dipeptide

- Evolution of alanine dipeptide in vacuum.



- Free energy can be calculated if the trajectory diplays ergodicity as:

$$F(\Phi, \Psi) = -\beta^{-1} \log P(\Phi, \Psi), \tag{83}$$

where β is the inverse temperature and $P(\Phi, \Psi)$ is the probability of occurence in a given state described by the dihedral angles.

Alanine Dipeptide

- Difference between the ergodic and non-ergodic trajectories.

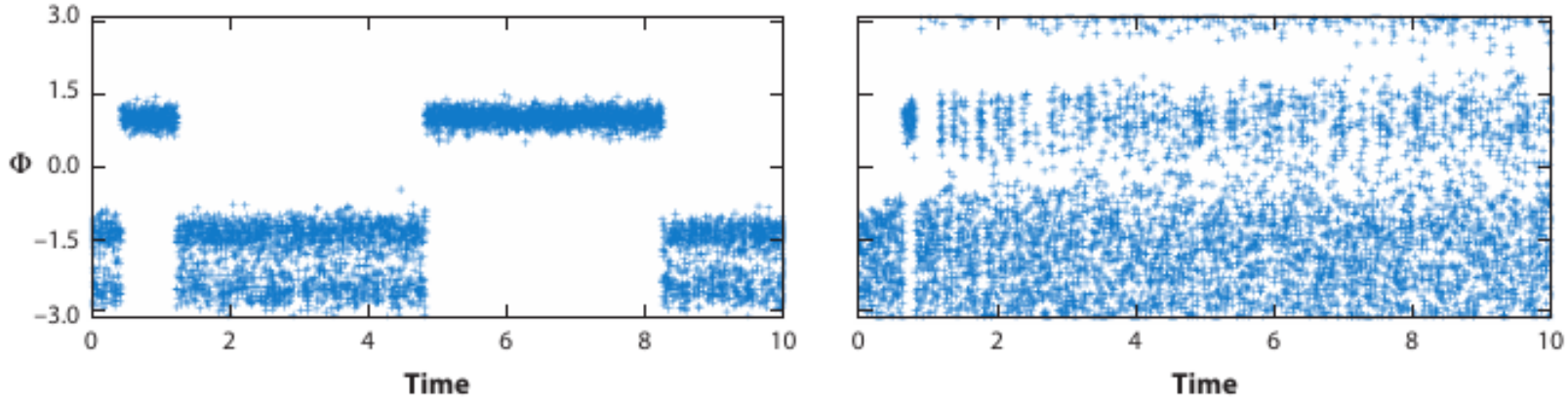


Figure 25: Time evolution of alanine dipeptide in vacuum.

- In the left, the transitions between two energy basins are *rare* or *infrequent*.
- In the right, the trajectory jumps between the states many times.

Proteins in MD

- Proteins, DNA, viruses, membranes can be studied using MD.

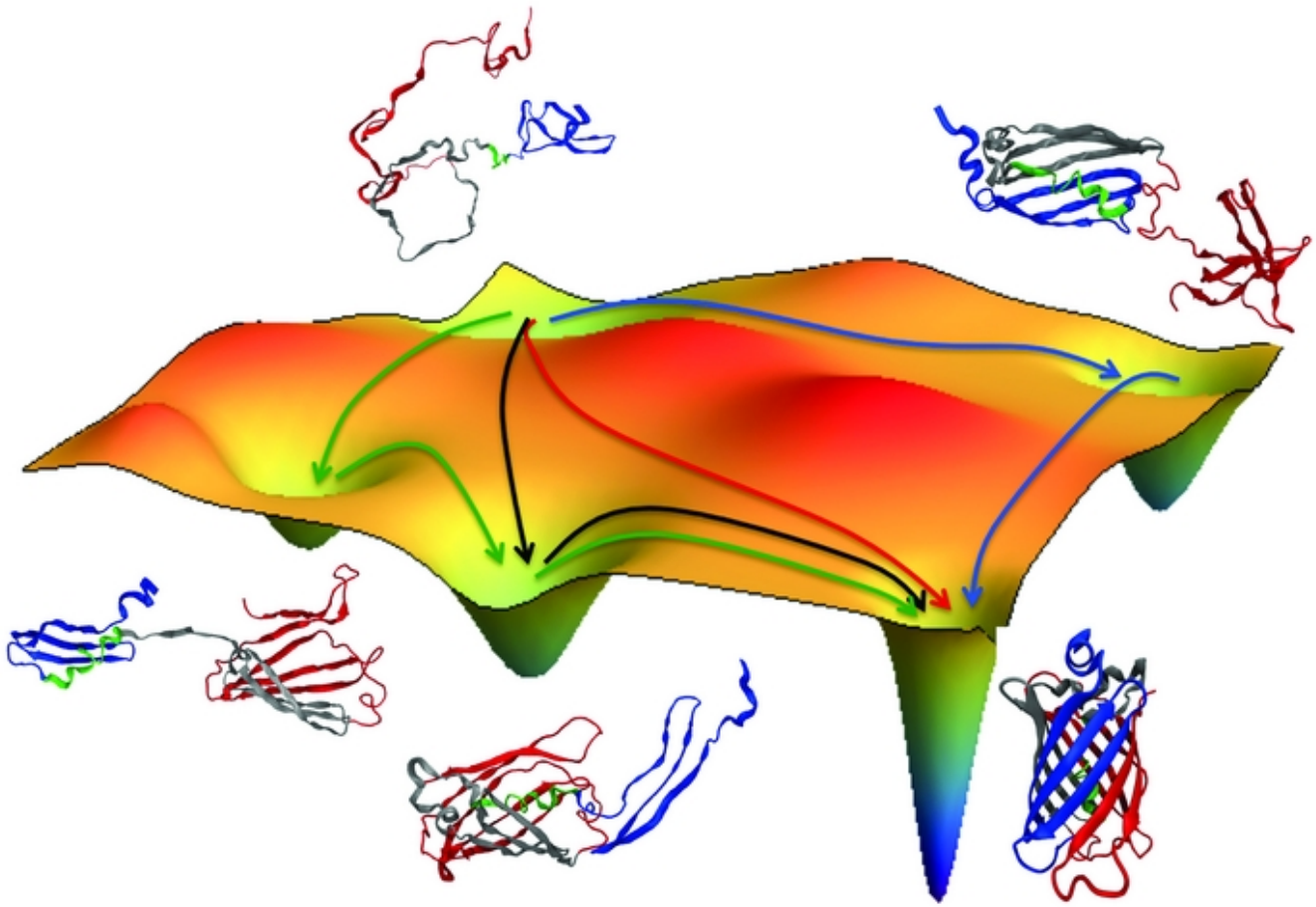
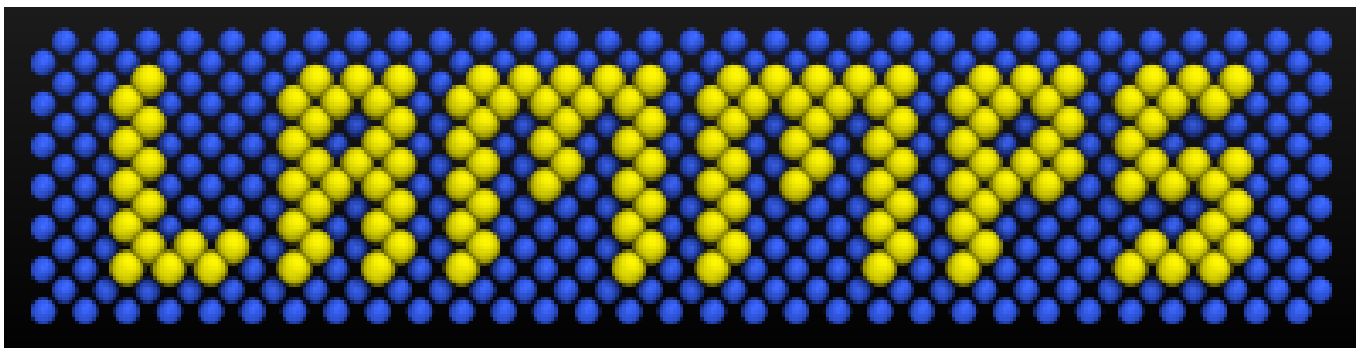


Figure 26: Protein in the phase space.

MD Engines

- Gromacs https://www.gromacs.org/
- NAMD https://www.ks.uiuc.edu/Research/namd/
- OpenMM https://openmm.org/
- LAMMPS https://lammps.sandia.gov/



Source: https://lammps.sandia.gov/

LAMMPS

LAMMPS is a classical molecular dynamics (MD) code that models ensembles of particles in a liquid, solid, or gaseous state. It can model atomic, polymeric, biological, solid-state (metals, ceramics, oxides), granular, coarse-grained, or macroscopic systems using a variety of interatomic potentials (force fields) and boundary conditions. It can model 2d or 3d systems with only a few particles up to millions or billions.[10]

[10] Source: LAMMPS Page: https://lammps.sandia.gov/

LAMMPS: Executables for Linux

- Pre-built Ubuntu Linux executable (`lammps-stable`):

```
$ sudo add-apt-repository ppa:gladky-anton/lammps
$ sudo apt-get update
$ sudo apt-get install lammps-stable
```

- Pre-built OpenSuse Linux executable (`lmp`):

```
$ zypper install lammps
```

- Archlinux build-script (`lmp`):

```
$ git clone https://aur.archlinux.org/lammps.git
$ cd lammps
$ makepkg -s
$ makepkg -i
```

- Also Conda:

```
$ conda config --add channels conda-forge
$ conda create -n lammps-env
$ conda activate lammps-env
$ conda install lammps
```

Source: https://lammps.sandia.gov/doc/Install.html

LAMMPS: cmake

- However, it is suggested to build LAMMPS yourself.
- You can do this executing `./build-lammps.sh`.

```
#!/bin/bash
set -e

LAMMPS_VERSION=3Mar2020
Make_CPUs=4
URL=https://github.com/lammps/lammps/archive
wget ${URL}/stable_${LAMMPS_VERSION}.zip
unzip stable_${LAMMPS_VERSION}.zip
rm -f stable_${LAMMPS_VERSION}.zip
cd lammps-stable_${LAMMPS_VERSION}

mkdir build; cd build
cmake -DPKG_MANYBODY=yes \
      -DPKG_KSPACE=yes \
      -DPKG_MOLECULE=yes \
      -DPKG_RIGID=yes \
            ../cmake
make VERBOSE=1 -j ${Make_CPUs}
```

LAMMPS

- The tarball for this project can be downloaded here: click.
- Contains the input file for the unbiased simulation.

```
irtg-school-mainz-2020/ $ tree
.
|-- data.nacl
|-- in.nacl
|-- nacl.psf
|-- run.sh
|-- start.lmp
|-- state.vmd
```

Source:
https://ves-code.github.io/doc-v2.7-irtg-school-2020/user-doc/html/ves_tutorials.html

LAMMPS

- The files comes from the TRR 146 IRTG School 2020 tutorial, and were prepared by Omar Valsson.
- We consider NaCl in aqueous solution.
- The dissociation barrier is expected to be around 2.5 $k_{\mathrm{B}}T$.
- Collective solvent motions play an important role in the transition.
- The system contains 1 Na, 1 Cl, and 106 water molecules (total 320 atoms).

LAMMPS

- The system:

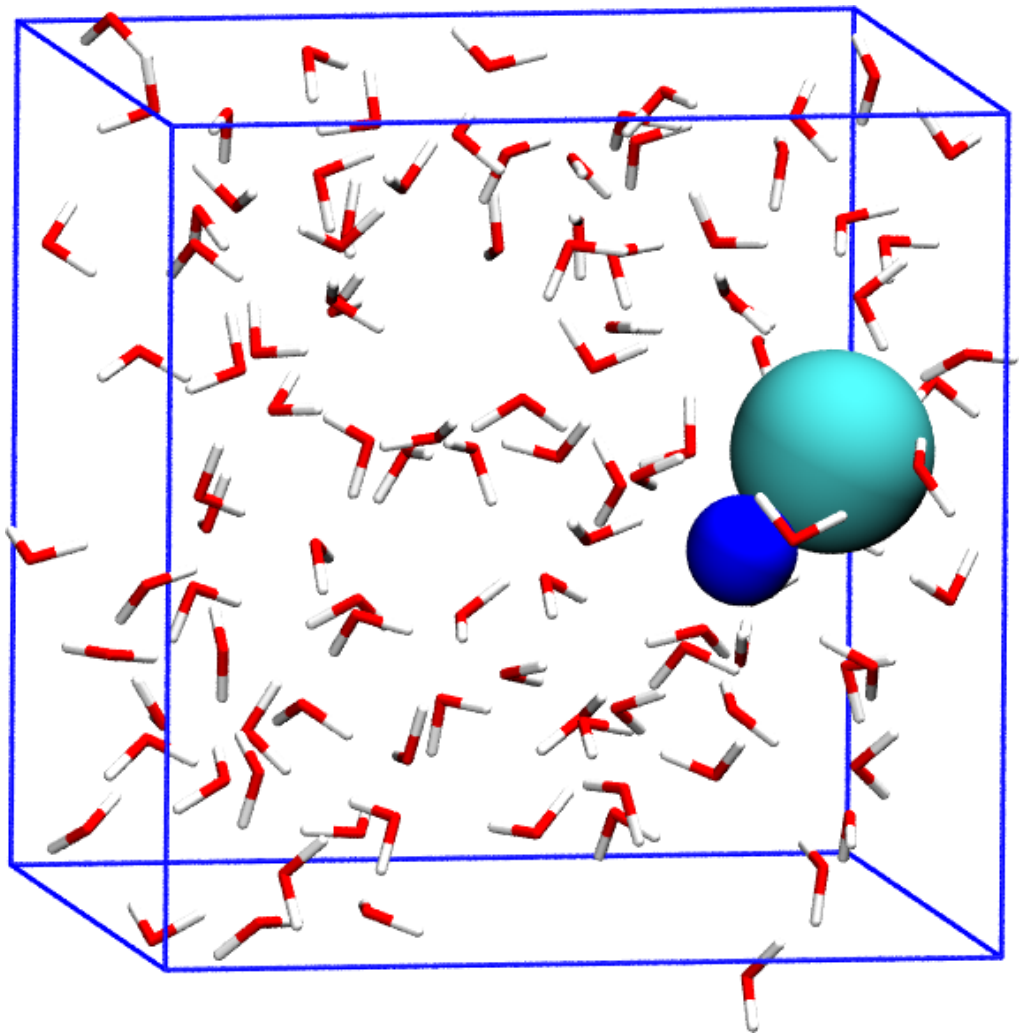


Figure 27: NaCl in water.

LAMMPS

- Download VMD (click)
- Load the LAMMPS data file (`data.nacl`).
- Open VMD → Extensions → TK console.

```
$ topo readlammpsdata data.nacl
$ play state.vmd
```

- We will run the simulation for 1 ns (see `start.lmp`)
- You can do this by executing `run.sh`.

```
#!/bin/bash

lmpexec=$HOME/lammps-stable_3Mar2020/build/lmp
ncor=2

mpirun -np ${ncor} ${lmpexec} < start.lmp > out.lmp
```

Source: https://www.plumed.org/

PLUMED

PLUMED is an open-source, community-developed library that provides a wide range of different methods, which include:

- enhanced-sampling algorithms,
- free-energy methods,
- tools to analyze the vast amounts of data produced by molecular dynamics (MD) simulations.

These techniques can be used in combination with a large toolbox of collective variables that describe complex processes in physics, chemistry, material science, and biology.[11]

[11] Source: https://www.plumed.org/

PLUMED: Build for Linux

- Run `build-plumed.sh` (plumed):

```
#!/bin/bash
set -e

PLUMED_VERSION=2.6.2
Make_CPUs=2
URL=https://github.com/plumed/plumed2/archive
wget ${URL}/v${PLUMED_VERSION}.zip
unzip v${PLUMED_VERSION}.zip
cd plumed2-${PLUMED_VERSION}
INSTALL_DIR=${PWD}/install

./configure --prefix=${INSTALL_DIR}
make -j ${Make_CPUs}
make install
```

- Or use Conda:

```
$ conda config --add channels conda-forge
$ conda create -n lammps-env
$ conda activate lammps-env
$ conda install -c conda-forge plumed
```

Source: https://www.plumed.org/doc-v2.7/user-doc/html/_installation.html

PLUMED

- After you have finished the simulation, you should have `out.dcd` which stores the 1-ns trajectory of our system.
- Now we will use PLUMED to analyze our simulation.
- The distance between Na and Cl can be calculated using the following PLUMED action:

```
distance: DISTANCE ATOMS=319,320
```

- And the coordination number which represents the collective motion of the solvent wrt Na can be computed as:

```
COORDINATION ...
GROUPA=319
GROUPB=1-318:3
SWITCH={RATIONAL R_0=0.315 D_MAX=0.5 NN=12 MM=24}
NLIST
NL_CUTOFF=0.55
NL_STRIDE=10
LABEL=coord
... COORDINATION
```

PLUMED

- The coordination number calculates the following rational switching function:

$$f(x) = \frac{1 - \left(\frac{x}{r_0}\right)^n}{1 - \left(\frac{x}{r_0}\right)^m}, \tag{84}$$

  where $r_0 = 0.315$, $n = 12$, and $m = 24$. You can see the function here.

- Use this command to calculate the defined variables along the trajectory (stored in `run-driver.sh`):

```
plumed driver --plumed plumed-driver.dat \
              --timestep 0.002 \
              --trajectory-stride 100 \
              --mf_dcd out.dcd
```

PLUMED

- You should obtain `COLVAR-driver` and see something like:

```
#! FIELDS time distance coord
 0.200000 0.568067 5.506808
 0.400000 0.500148 4.994588
 0.600000 0.449778 4.931140
 0.800000 0.528272 5.105816
 1.000000 0.474371 5.089863
 1.200000 0.430620 5.091551
 1.400000 0.470374 4.993886
 1.600000 0.458768 4.940097
 1.800000 0.471886 4.952868
 2.000000 0.489058 4.897593
.
.
```

LAMMPS

Exercises

1. Load the trajectory: in terminal `vmd out.dcd -psf nacl.psf`, then in VMD `pbc wrap -compound res -all`.
2. Change the visualization of atoms and render the system.
3. How the distance between Na and Cl changes in time?
4. Plot distances as a histogram (normalized). What distance value is the most probable?
5. From the histogram calculate $F = -\frac{1}{\beta} \log N$, where N is the histrogram.
6. Where the barrier is located?
7. Perform the same analysis (from step 2) for the coordination number.

LAMMPS

Exercises

1. Pick your tools – probably using `python` would be the easiest, but you can select anything you are used to.
2. Make an account on GitHub: https://github.com/.
3. Create a repository called `0800-fizobl-lammps`.
4. Sync your work.
5. Make the required plots and describe your results in `README.md`.
6. This project will be taken with 0.4 weight for your course grade.
7. Problems? Questions? Ask on our Slack channel: `#0800-fizobl-21`.

Enhanced Sampling

Enhanced Sampling: Source

- J. Rydzewski, and O. Valsson, *Multiscale Reweighted Stochastic Embedding (MRSE): Deep Learning of Collective Variables for Enhanced Sampling*, arXiv, 2021.
- O. Valsson, P. Tiwary, and M. Parrinello, *Enhancing Important Fluctuations: Rare Events and Metadynamics from a Conceptual Viewpoint*, Annual Review of Physical Chemistry, 67:159-184, 2016.

Long Timescales

- Modeling the long-timescale behavior of complex dynamical systems is a fundamental task in physical sciences.
- MD simulations allow us to probe the spatiotemporal details of molecular processes, but the so-called sampling problem severely limits their usefulness in practice.
- This sampling problem comes from the fact that a typical free energy landscape is characterized by many metastable states separated by free energy barriers that are much higher than the thermal energy $k_{\mathrm{B}}T$.

$k_{\mathrm{B}}T$

- $k_{\mathrm{B}}T$ is used in physics as a scale factor for energy values in molecular-scale systems.
- Inverse of $k_{\mathrm{B}}T$ is called the inverse temperature and generally as $\beta = \frac{1}{k_{\mathrm{B}}T}$.
- For a system in equilibrium in canonical ensemble, the probability of the system being in state with energy U is proportional to $\mathrm{e}^{-\beta U}$.
- $k_{\mathrm{B}}T$ is the amount of heat required to increase the thermodynamic entropy of a system by k_{B}.

$k_{\mathrm{B}}T$

Approximate values of kT at 298 K	Units
$kT = 4.11 \times 10^{-21}$	J
$kT = 4.114$	pN·nm
$kT = 9.83 \times 10^{-22}$	cal
$kT = 25.7$	meV
kT=-174	dBm/Hz
Related quantities (also at 298 K)	
$kT/hc \approx 207$ [1]	cm^{-1}
$kT/e = 25.7$	mV
$RT = kT \cdot N_A = 2.479$	$kJ \cdot mol^{-1}$
$RT = 0.593$	$kcal \cdot mol^{-1}$
$h/kT = 0.16$	ps

Figure 28: $k_{\mathrm{B}}T$ in different units.[12]

[12] Source: https://en.wikipedia.org/wiki/KT_(energy)

Metastable States

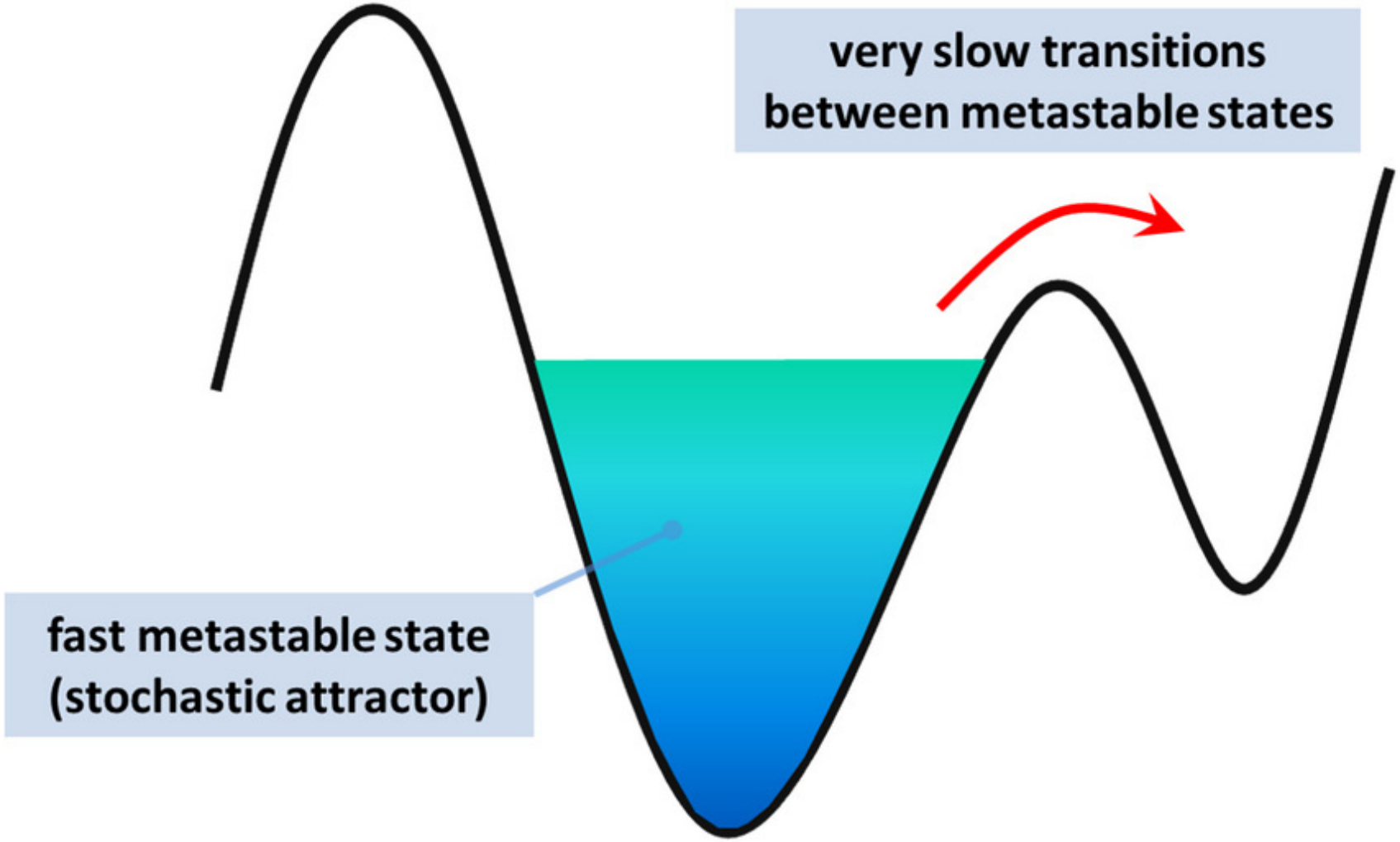


Figure 29: Slow transitions between metastable states.

Sampling Problem

- On the timescale one can simulate, barrier crossings are rare events, and the system remains kinetically trapped in a single metastable state.

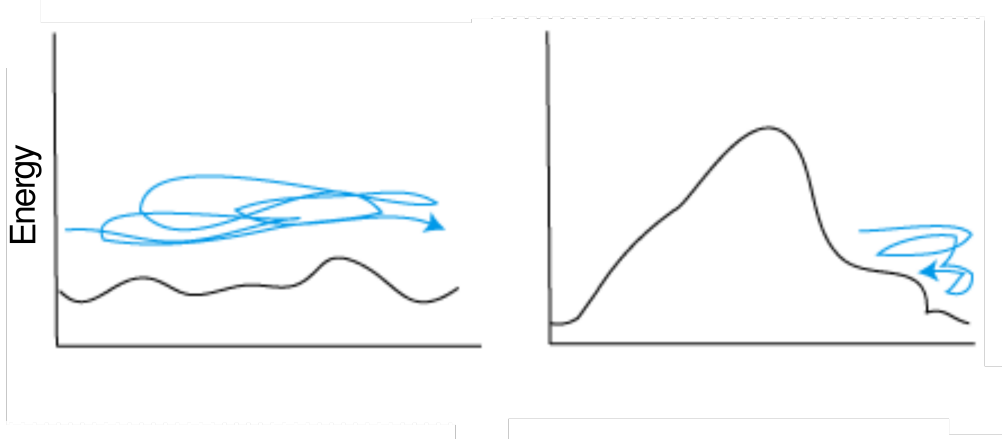


Figure 30: Ergodic and infrequent trajectories.

- ES can be used to alleviate the sampling problem.
- Here, we focus on ES methods that work by identifying coarse variables and enhancing their fluctuations.

Theory of Enhanced Sampling

- Consider a molecular system, described by microscopic coordinates $\mathbf{R}$ and a potential energy function $U(\mathbf{R})$.
- We limit our theory to the canonical ensemble (NVT).
- At equilibrium, the microscopic coordinates follow the Boltzmann distribution:

$$P(\mathbf{R}) = \mathrm{e}^{-\beta U(\mathbf{R})} / \int \mathrm{d}\mathbf{R}\, \mathrm{e}^{-\beta U(\mathbf{R})}, \tag{85}$$

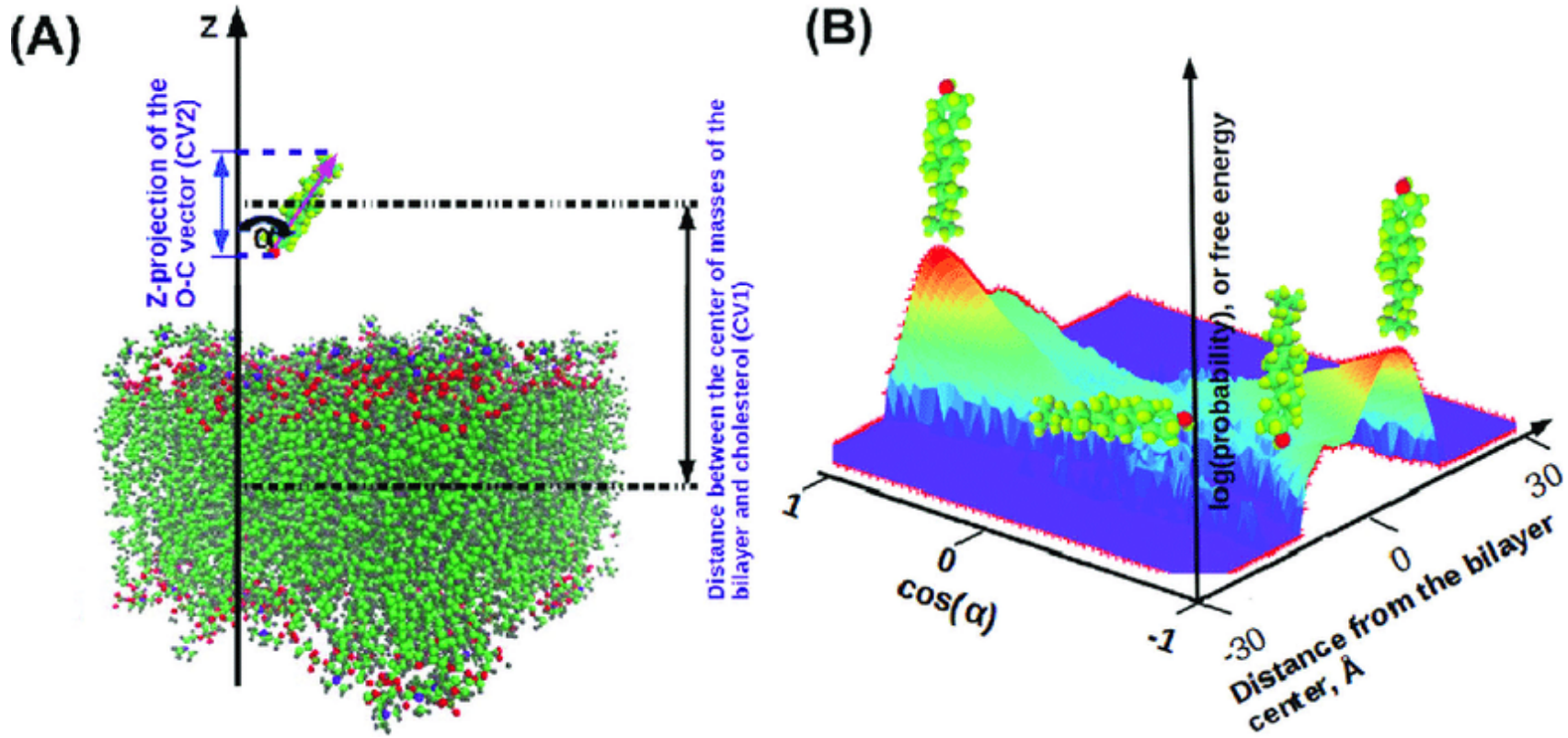
  where β is the inverse of the thermal energy.
- We identify a small set of coarse-grained order parameters called *collective variables* (CVs):

$$\mathbf{s}(\mathbf{R}) = [s_1(\mathbf{R}), s_2(\mathbf{R}), \ldots, s_d(\mathbf{R})]. \tag{86}$$

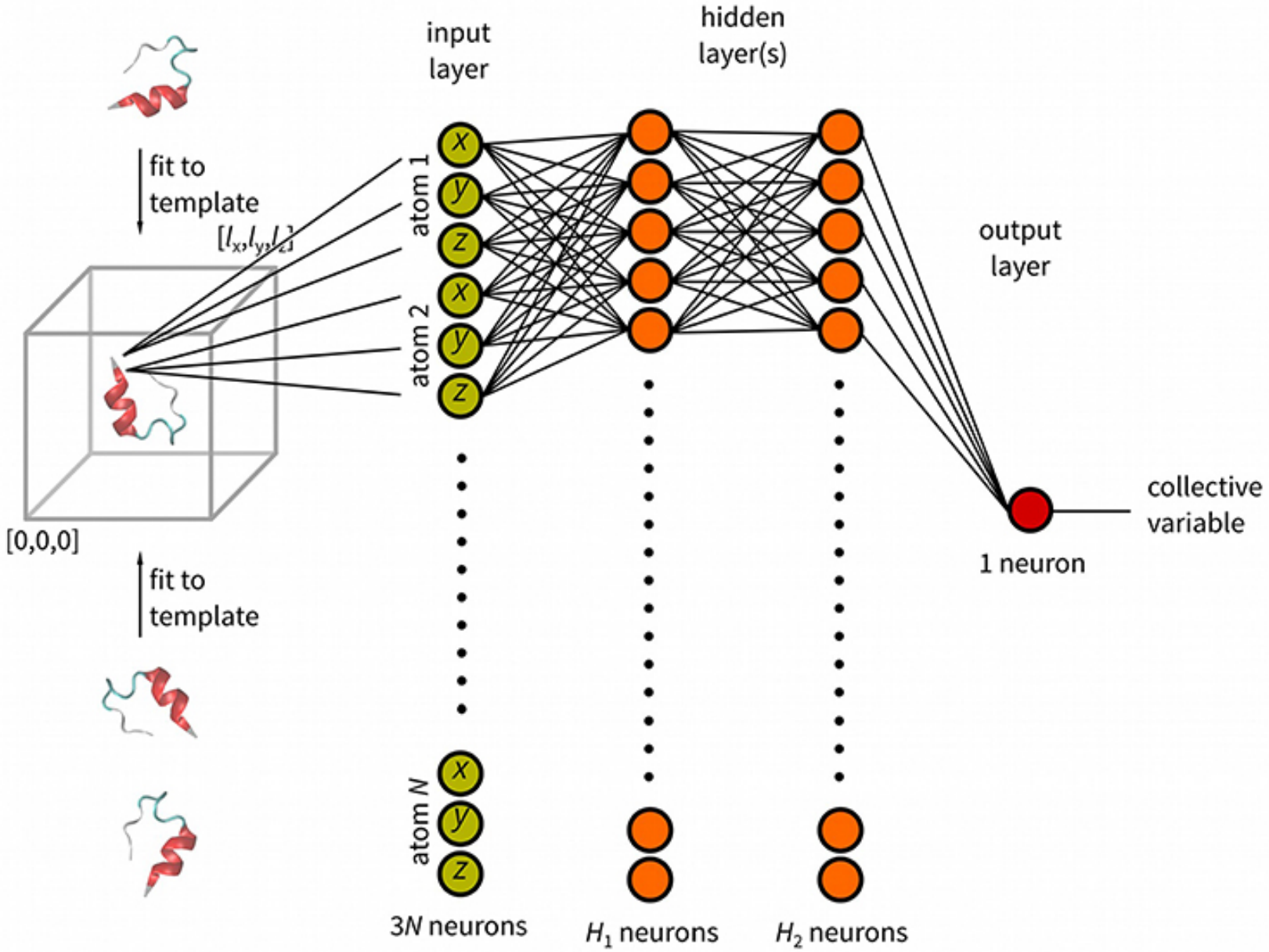
Collective Variables

- Also called *order parameters* or *coarse-grained variables* in some publications.
- Should be of lower dimension than microscopic coordinates $\mathbf{R}$.
- Can be any function of microscopic coordinates $s(\mathbf{R})$.
- For instance: angles, distances, contact map, dihedral angles, index along a predefined reaction coordinate, position, and many others.
- Can be periodic.
- Any differentiable function (to calculate forces) that discriminates between metastable states.
- Usually selecting CVs is a very difficult task.
- Nowadays machine learning can be used to find CVs: dimensionality reduction methods based on neural networks.

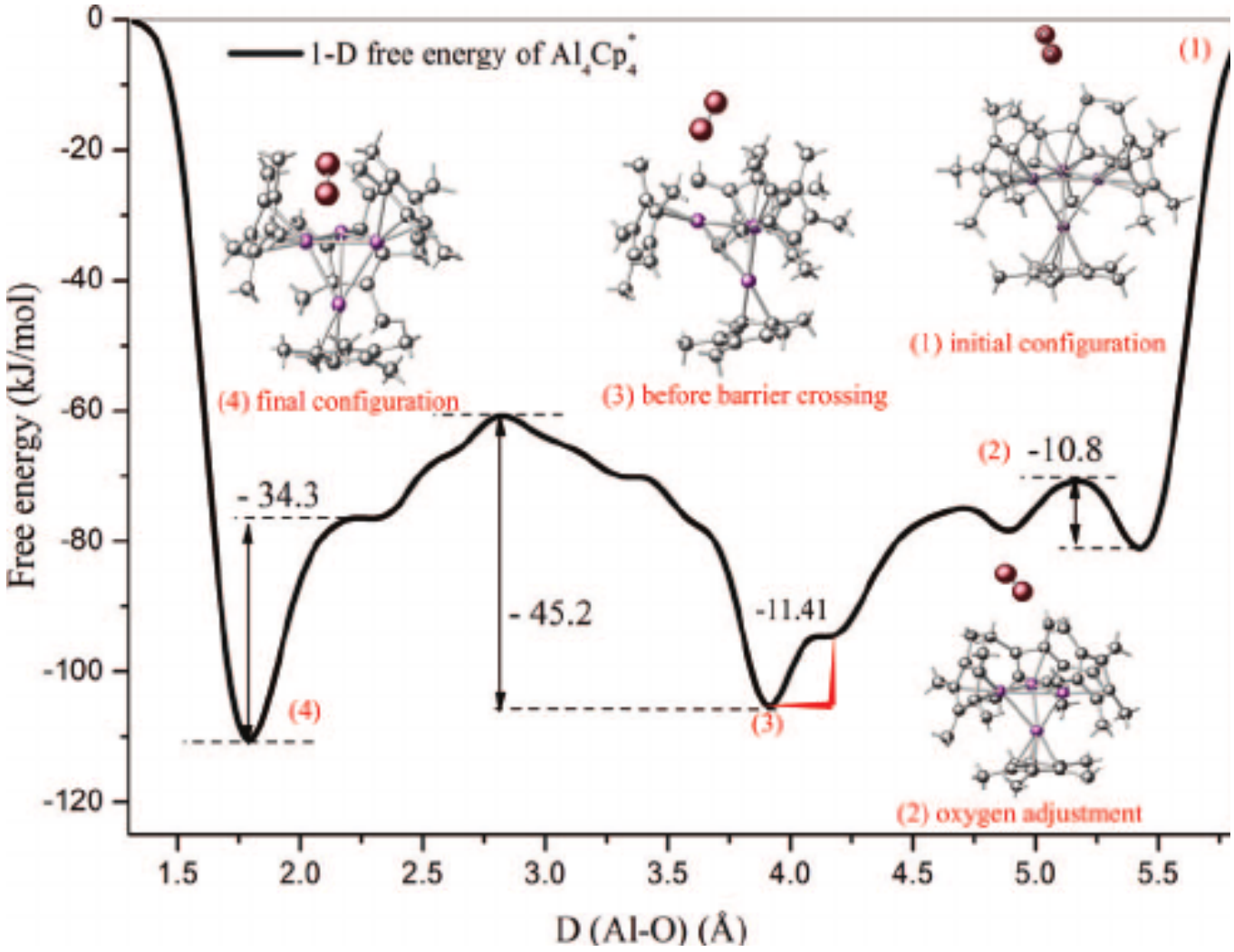
CVs: Examples



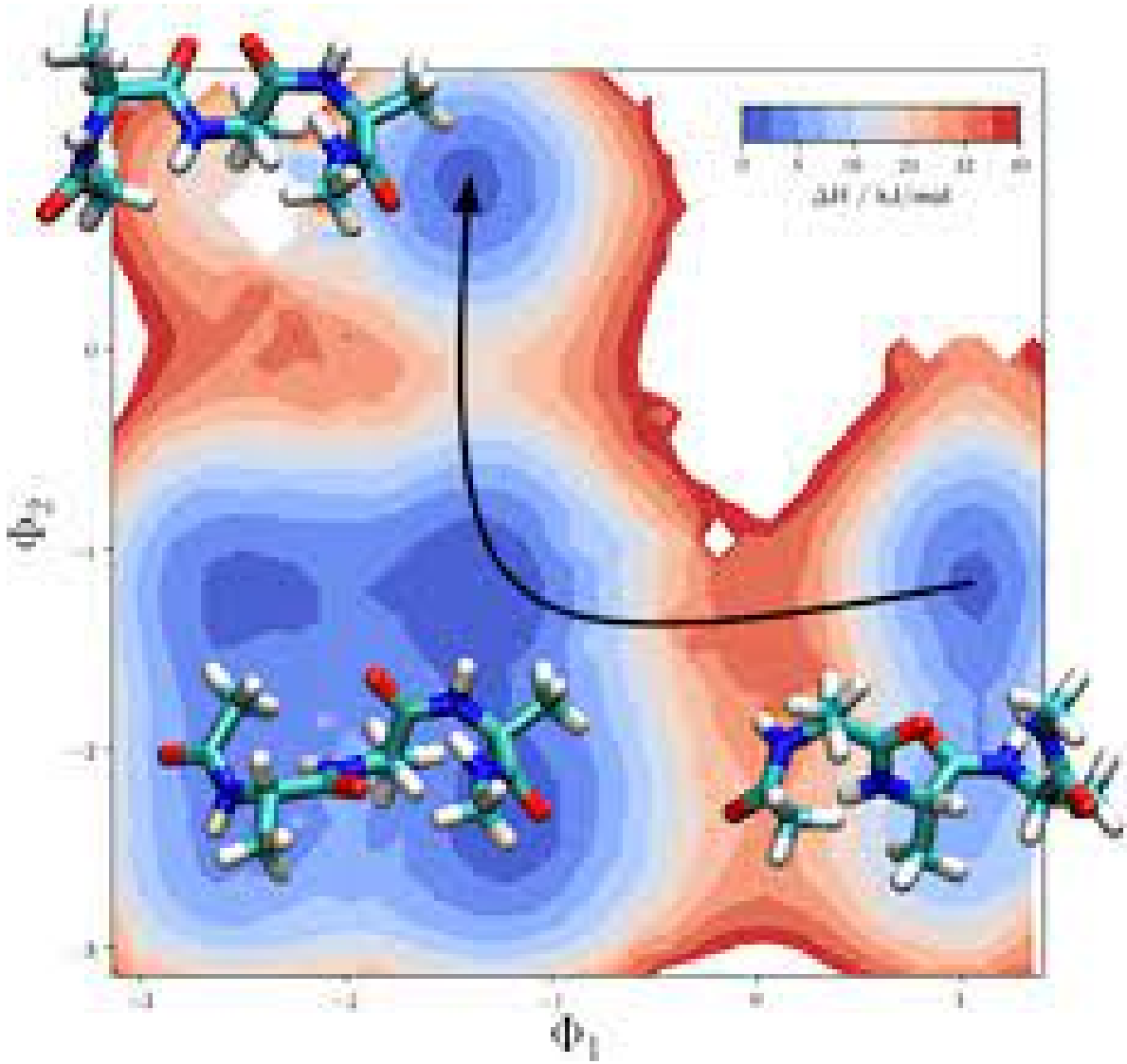
CVs: Examples



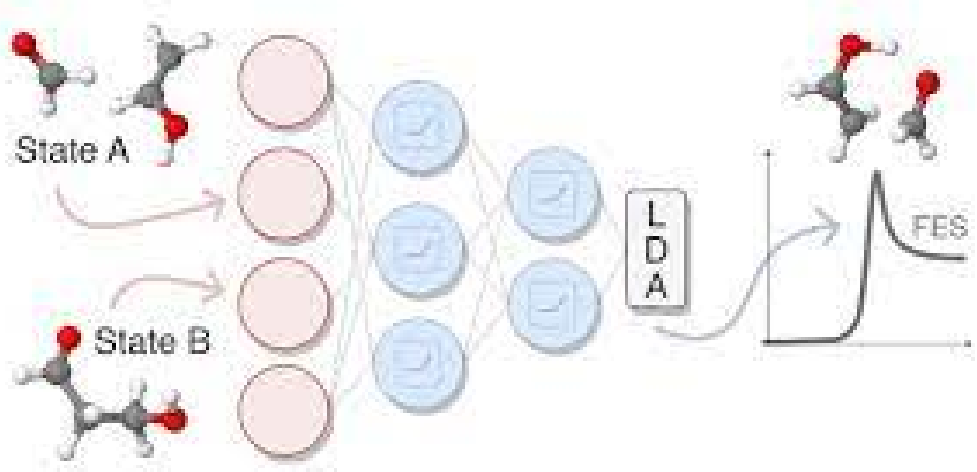
CVs: Examples



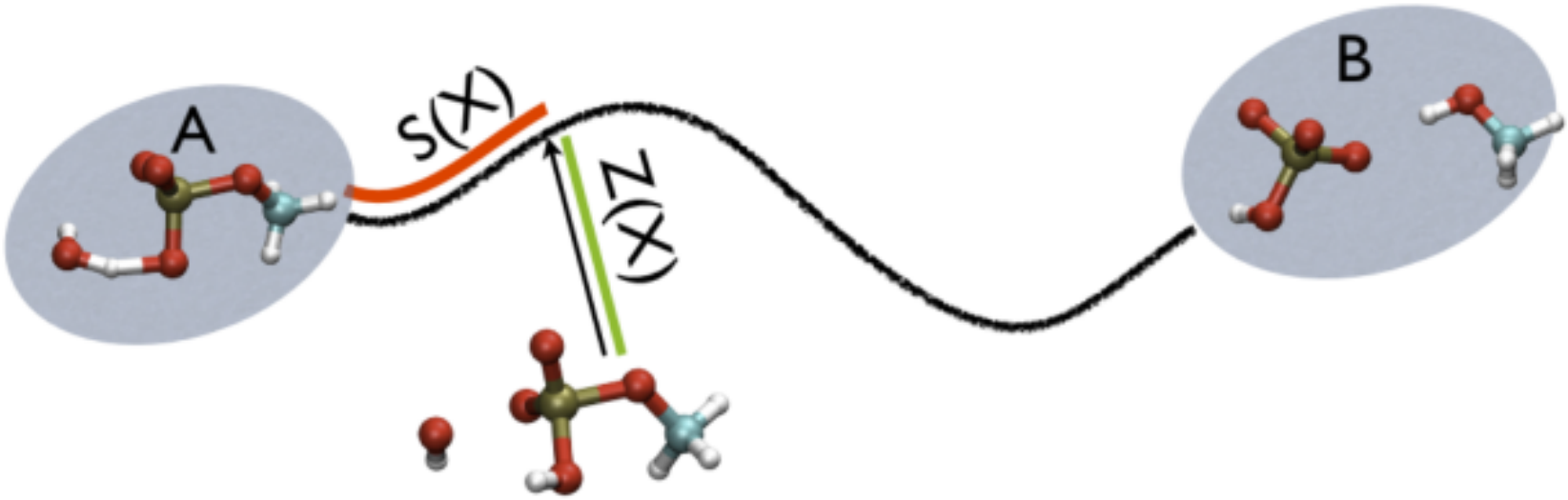
CVs: Examples



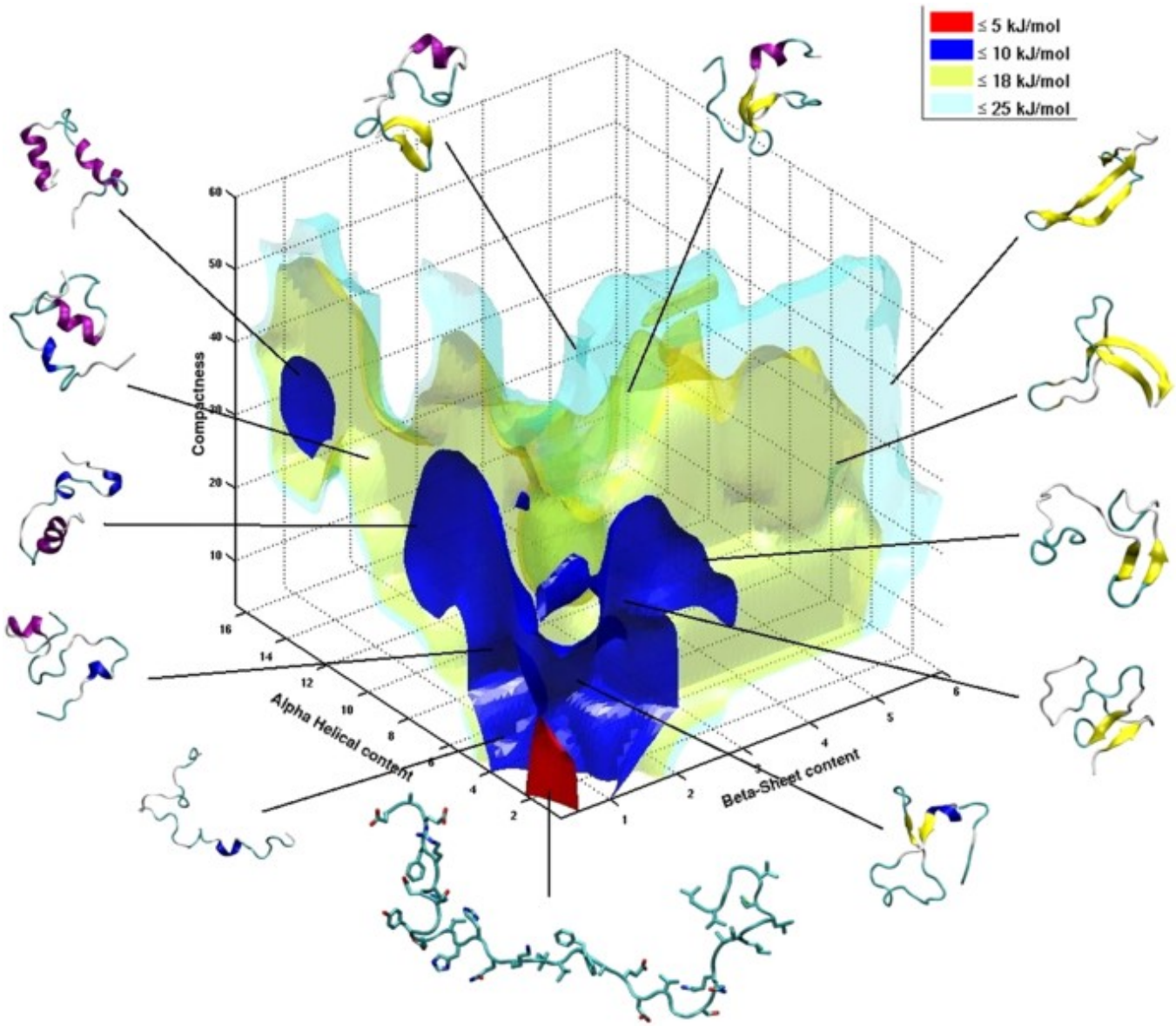
CVs: Examples



CVs: Examples



CVs: Examples



Theory of Enhanced Sampling

- The equilibrium marginal distribution of CVs is given by integrating out all other degrees of freedom:

$$P(\mathbf{s}) = \int \mathrm{d}\mathbf{R}\, \delta\left[\mathbf{s} - \mathbf{s}(\mathbf{R})\right] P(\mathbf{R}), \tag{87}$$

  where we use the sifting property of the Dirac δ-function.

- Eq. 87 is equivalent to:

$$P(\mathbf{s}) = \Big\langle \delta[\mathbf{s} - \mathbf{s}(\mathbf{R})] \Big\rangle, \tag{88}$$

  where $\big\langle \cdot \big\rangle$ denotes an ensemble average.

- Up to an unimportant constant, the *free energy* surface (FES) is given by:

$$F(\mathbf{s}) = -\beta^{-1} \log P(\mathbf{s}). \tag{89}$$

Theory of Enhanced Sampling

- FES contains all the relevant information about the system, such as the location and relative stability of the metastable states and the free energy barriers separating them.
- The relative free energy difference between two metastable states A and B can be determined from $F(\mathbf{s})$ by integrating the probabilities of the two states:

$$\Delta F_{A,B} = -\frac{1}{\beta} \log \frac{\int_A \mathrm{d}\mathbf{s} \mathrm{e}^{-\beta F(\mathbf{s})}}{\int_B \mathrm{d}\mathbf{s} \mathrm{e}^{-\beta F(\mathbf{s})}}, \tag{90}$$

where the integrations are performed over the two regions in CV space that define the two metastable states A and B, respectively.

Theory of Enhanced Sampling

- For an ergodic system, the FES can equivalently be computed as:

$$F(\mathbf{s}) = -\frac{1}{\beta} \lim_{t\to\infty} N(\mathbf{s}, t), \tag{91}$$

where the normalized histogram is given as:

$$N(\mathbf{s}, t) = \frac{\int_0^t \mathrm{d}t' \; \delta[\mathbf{s} - \mathbf{s}(\mathbf{R}(t'))]}{\int_0^t \mathrm{d}t'}. \tag{92}$$

Theory of Enhanced Sampling

- CV-based enhanced sampling methods overcome the sampling problem by introducing an external bias potential $V(\mathbf{s}(\mathbf{R}))$ acting in CV space.
- This leads to sampling according to a *biased* distribution:

$$P_V(\mathbf{R}) = \frac{\mathrm{e}^{-\beta[U(\mathbf{R})+V(\mathbf{s}(\mathbf{R}))]}}{\int \mathrm{d}\mathbf{R}\, \mathrm{e}^{-\beta[U(\mathbf{R})+V(\mathbf{s}(\mathbf{R}))]}} \tag{93}$$

- Compare it to Eq. 85.
- Idea of Torrie and Valleau, 1977.

Theory of Enhanced Sampling

- At convergence, CVs follow a biased distribution:

$$P_V(\mathbf{s}) = \int \mathrm{d}\mathbf{R}\, \delta\left[\mathbf{s} - \mathbf{s}(\mathbf{R})\right] P_V(\mathbf{R}) = \frac{\mathrm{e}^{-\beta[F(\mathbf{s})+V(\mathbf{s})]}}{\int \mathrm{d}\mathbf{s}\, \mathrm{e}^{-\beta[F(\mathbf{s})+V(\mathbf{s})]}}, \tag{94}$$

  which is easier to sample.

- The standard reweighting from importance sampling applies here:

$$P(\mathbf{R}) \propto P_V(\mathbf{R}) w(\mathbf{s}(\mathbf{R})) = P_V(\mathbf{R}) \mathrm{e}^{\beta V(\mathbf{s}(\mathbf{R}))}. \tag{95}$$

- CV-based methods differ in how they construct the bias potential and which kind of biased CV sampling they obtain at convergence.

Theory of Enhanced Sampling

- A non-exhaustive list of modern CV-based enhanced sampling techniques includes multiple windows umbrella sampling, adaptive biasing force, Gaussian-mixture umbrella sampling, metadynamics, variationally enhanced sampling, on-the-fly probability-enhanced sampling (OPES), and ATLAS.
- But let's start with umbrella sampling, proposed by the seminal work of Torrie and Valleau.

Umbrella Sampling

- TODO