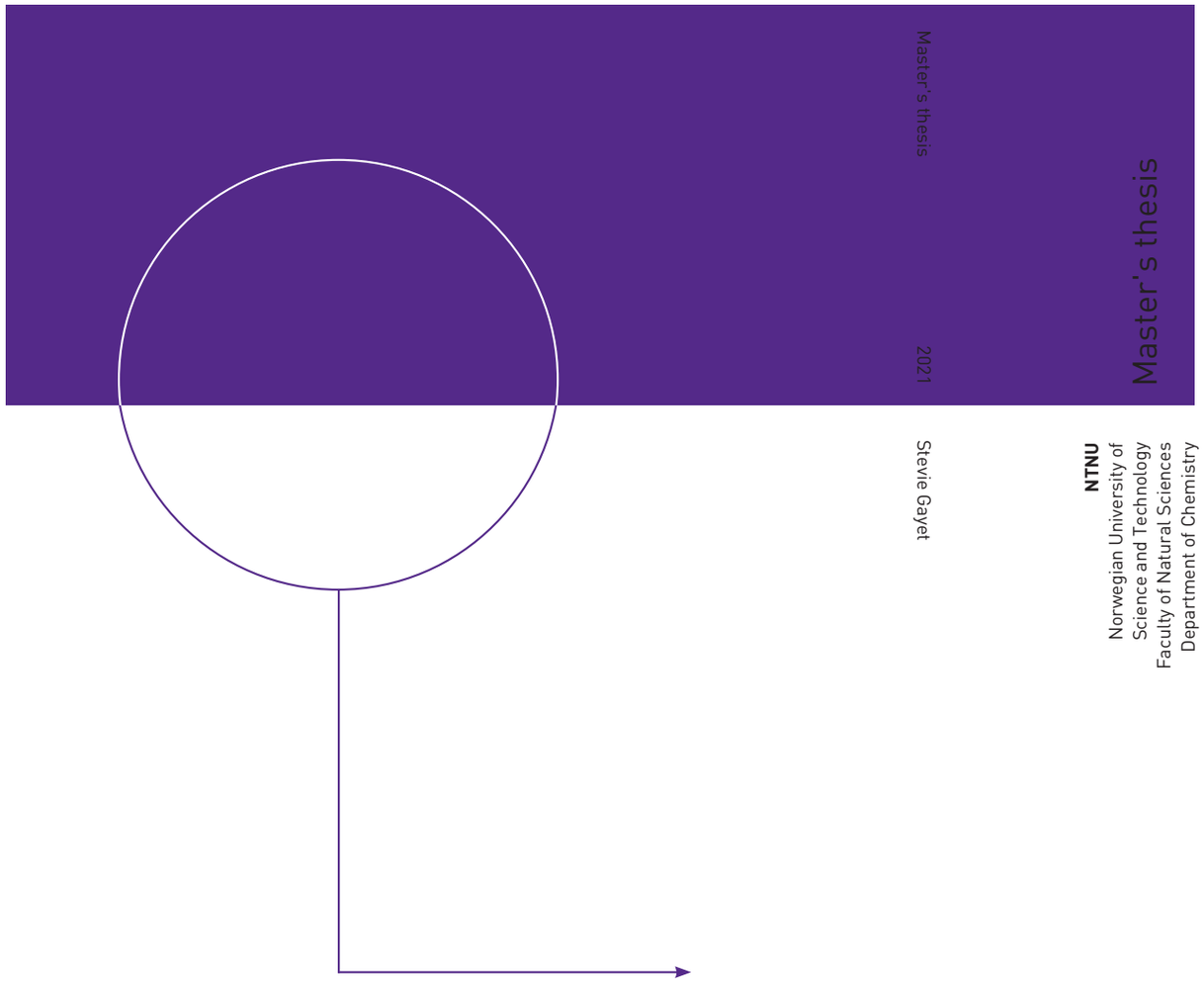

Stevie Gayet

Investigation of a parameter-free density estimation to analyze complex mechanisms using PyRETIS

May 2021

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

NTNU

Norwegian University of
Science and Technology

Investigation of a parameter-free density estimation to analyze complex mechanisms using PyRETIS

Stevie Gayet

Master in Chemistry
Submission date: May 2021
Supervisor: Titus van Erp
Co-supervisor: Sander Roet

Norwegian University of Science and Technology
Department of Chemistry

Abstract

Path sampling methods are successful in studying rare events. In chemical reaction, it is the process where the system moves from the reactant state, crosses a barrier, and reaches the product state.

Although the primary goal of these methods has been the computation of the reaction rate of a chemical reaction, recent developments focus on extracting more insights about the reaction mechanism and hence find a way to drive a reaction from the reactant state to the product state. PyRETIS is a Python library for rare events simulations based on transition interface sampling (TIS) and replica exchange TIS (RETIS).

As of now, PyRETIS uses handcraft parameters to process RETIS data.

This work asks whether it is possible to use automatic methods to tune these parameters and eliminate the need for manual tweaking. This willingness merges with the goal of PyRETIS of being tailored for non-expert users, as opposed to OpenPathSampling, another path sampling library, which is more suitable for users eager to choose themselves the parameters.

This work introduces a method based on kernel density estimation for reactive and nonreactive density functions and assesses its performance. The main idea of this work is to simulate fake data to decide whether this kind of method would be usable in a near future for PyRETIS.

Acknowledgements

This master thesis was carried out under the supervision of Professor Titus van Erp and co-supervision of Ph.D. Candidate Sander Roet. Both of them have been of great help for understanding the theory behind rare events, transition path sampling, and analysis. They have been of great help when dealing with new ideas and deciding whether they were realistic and usable.

I would also like to thank Anders Lervik for helping me with understanding and running analysis scripts.

Path Sampling simulations and, more generally, theoretical chemistry is a broad subject and is hard to get hold of at first hand, but very rewarding at the end of the day.

Contents

Chapter 1

Introduction

A rare event is defined as an event occurring very infrequently compared to other relaxation processes involved in the phenomenon. Rare events are of significant interest in many cutting-edge fields and applications, such as climate change, economics, or disease spreading[4]. In the vast majority of chemical reactions, crossing the free energy barrier between the reactant state A and the product state B happens very infrequently.

Molecular dynamics (MD) can be used to model a reactive event. However, these methods have to perform well and be accurate using a relatively small number of particles (up to 100 000 molecules) and a total simulation time from nanoseconds to microseconds, using a time step of a few femtoseconds, constrained by molecular vibrations[8].

Hence, applying "vanilla" MD is only possible for systems with a relatively small energy barrier separating the reactant and the product state, as the probability of crossing the energy barrier decreases exponentially with the height of the barrier. The system will stay most of the time in either A or B and rarely jump from one state to the other state: this is the separation of the time scales, stable states as opposed to short transitions.

Path sampling methods are successful in studying rare events. These methods mainly focused on the computation of the reaction rate but another

major topic of interest is studying and analyzing reaction mechanism [6].

Chapter 2

Theory

2.1 Transition State Theory and Bennett-Chandler approach

2.1.1 Statistical mechanics definitions

- $x = r, p$: phase space point with r Cartesian coordinates and p corresponding momenta of all N particles.
- $\rho(x) = \exp(-\beta\mathcal{H}(x))/Z$: equilibrium distribution with $\mathcal{H}(x)$ the Hamiltonian of the system, $\beta = 1/k_B T$ and $Z = \int \exp(-\beta\mathcal{H}(x))dx$ the partition function.
- The ensemble average of an observable O is $< O >= \int O(x)\rho(x)dx$.
- The free energy is computed by projecting the phase space onto a continuous function $\lambda(r)$ the reaction coordinate:

$$\exp(-\beta F(\lambda^*)) \equiv< \delta(\lambda(r) - \lambda^*) >= \int \rho(r)\delta(\lambda(r) - \lambda^*)dr$$

The probability histogram $P(\lambda)$ can be seen as a series of delta func-

tions and the free energy can be computed using $F(\lambda) = -k_B T \ln(P(\lambda))$. The probability distribution can be computed using umbrella sampling for example.

2.1.2 Rate Constants

Very often in a rare event, the molecular transition time τ_{mol} is much smaller than the time spent by the system in one of the two states, τ_{stable}: $\tau_{mol} \ll \tau_{stable}$. The rate constant for a system made up of two stables states ,A and B, separated by a significant activation barrier, could be defined as the number $N_{A\to B}(\mathcal{T})$ of transitions from A to B during time $\mathcal{T}$, given that the system is initially in state A:

$$k_{AB} = \lim_{\mathcal{T}\to\infty} \frac{N_{A\to B}(\mathcal{T})}{t^A_{tot}}$$

with

$$t^A_{tot}(\mathcal{T}) = \sum_i t^A_i$$

the total time spent in A during $\mathcal{T}$.

2.1.3 Rate Constants from Transition State Theory

The transition state theory (TST) evaluates the reaction rate by exploiting the properties of the free energy minima and the activation barrier. In the simplest systems, the activation barrier can be related to the first-order saddle points on the potential landscape: if the potential energy is regular enough such that the saddle points are easy to enumerate and related to the transition state, computing these points gives access the reaction rate[5].

However, in multi-dimensional many-body complex systems, the potential energy is not smooth anymore: the number of saddle points is growing exponentially with the number of degrees of freedom, and most of these

points are the order of k_BT. Entropy has a significant contribution and is not negligible anymore. A solution is to replace the potential energy with the free energy and compute the free energy barrier as a function of the reaction coordinate: this is the central point of the TST methods.

TST methods involve reactive flux methods: first, the free energy as a function of the reaction coordinates is computed using for instance umbrella sampling. The rate is proportional to the probability density to be at the top of the free energy barrier. If the probability density is multiplied by an analytical flux term, one obtains the TST estimate of the reaction rate. If needed, this estimate can be made exact by the computation of a transmission coefficient, obtained by starting many short trajectories from the top of the barrier. Nevertheless, the efficiency of this procedure depends on the reaction coordinate chosen; it might be challenging to find such a coordinate for high-dimensional systems.

We define

$$h_\Omega(r) = \begin{cases} 1 & \text{if } r \in \Omega, \\ 0 & \text{otherwise.} \end{cases}$$

So,

$$\langle h_\Omega(r) \rangle = \int_\Omega \rho(r) dr$$

In the TST, these state definitions usually depends on a single dividing surface, and therefore $h_A + h_B = 1$ so $\langle h_A \rangle + \langle h_B \rangle = 1$

We can define the mean time residence in A as before:

$$t_A^{mr} = \lim_{\mathcal{T} \to \infty} \frac{2}{N(\mathcal{T})} \int_0^T h_A(r(t)) dt$$

with $N(\mathcal{T})$ the number of times that, during $\mathcal{T}$, a very long dynamic trajectory crosses ∂A, the boundary between A and B. The factor 2 accounts for outgoing and ingoing crossings.

Using a reaction coordinate $\lambda(r)$, $A = \{r \in \mathbb{R}^n | \lambda(r) < \lambda^*\}$ and $\partial A = \{r \in \mathbb{R}^n | \lambda(r) = \lambda^*\}$, we have:

$$h_A(r) = \theta(\lambda^* - \lambda(r))$$

and

$$h_A(r) = \theta(\lambda(r) - \lambda^*)$$

Finally,

$$k_{AB}^{TST} = \frac{\left\langle \delta(\lambda(r) - \lambda^*)\dot{\lambda}\theta(\dot{\lambda}) \right\rangle}{\langle \theta(\lambda^* - \lambda(r)) \rangle}$$

and, using, the free energy,

$$k_{AB}^{TST} = \left\langle \dot{\lambda}\theta(\dot{\lambda}) \right\rangle_{\lambda=\lambda^*} \frac{\exp -\beta F(\lambda^*)}{\int_{-\infty}^{\lambda^*} \exp -\beta F(\lambda') d\lambda'}$$

TST also assumes that trajectories crossing ∂A do not recross it moments later, so localizing the dividing surface λ^* is crucial: the TST constant rate is the right one only if $\{x | \lambda(x) = \lambda^*\}$ is the separatrix, i.e the true transition state dividing the surface. However, it is impossible to know the exact location of the separatrix for complex systems.

2.1.4 The Bennett-Chandler approach

This reaction rate is sensitive to the choice of the reaction coordinate, and the computed rate is correct if $\{x | \lambda(x) = \lambda^*\}$ corresponds to the true separatrix (i.e., the transition state dividing system). The central assumption is that any trajectory coming from A and crossing λ^* will not recross λ^* afterward. In complex systems, it is impossible to know where the separatrix is located.

Fortunately, the k_{TST} rate constant can be corrected with a transmission coefficient in the Bennett-Chandler approach[1]. We can multiply the TST rate constant with $\kappa(t)$, the transmission coefficient to obtain the true rate constant.

$$k_{AB}(t) = k_{AB}\kappa(t)$$

with

$$\kappa(t) = \frac{\left\langle \dot{\lambda}(x_0)\theta(\lambda(x_t) - \lambda^*) \right\rangle_{\lambda=\lambda^*}}{\left\langle \dot{\lambda}(x_0)\theta(\dot{\lambda}(x_0)) \right\rangle_{\lambda=\lambda^*}}$$

After a short molecular time t_{mol}, the trajectories are committed to a stable state, and κ becomes a constant, leading to k_{AB}.

In $\kappa(t)$, recrossings are effectively taken into account: untrue trajectories $B \to B$ are not taken into account because positive and negative terms cancel, and a $A \to B$ trajectory with multiple λ^* crossings is counted only once. However, λ^* has to be close to the true transition state, otherwise κ will be very low, this requires a priori knowledge about the system.

2.2 Transition Path Sampling

In transition path sampling (TPS) methods, we care about path ensembles, which include different paths from state A to state B. One significant advantage is that only defining state A and state B carefully is required, it does not require a priori knowledge such as a reaction coordinate, for example[3].

TPS simulations, which aim at collecting all likely transition pathways between A and B, requires the following assumptions:

1. The stable states are characterized by an order parameter, if $\lambda < \lambda_A$, the system is in A and if $\lambda > \lambda_B$, the system is in B.

2. The system spends most of the time in states A and B, with rapid transitions between the two states.

3. There are no other stable states than A and B.

4. Stable states are separated by an unknown rough energy barrier or free energy barrier.

2.2.1 Path Probability

A path is described as a sequence of states $\{x_0, x_{\Delta t}, x_{2\Delta t}, \ldots, x_{\mathcal{T}}\}$ separated by Δt, $x = \{r, p\}$ represents the position and the momenta of all the particles in the system and $\mathcal{T}$ is the trajectory length. $x_{i\Delta t}$ is also called a snapshot of the system or a time slice at $i\Delta t$.

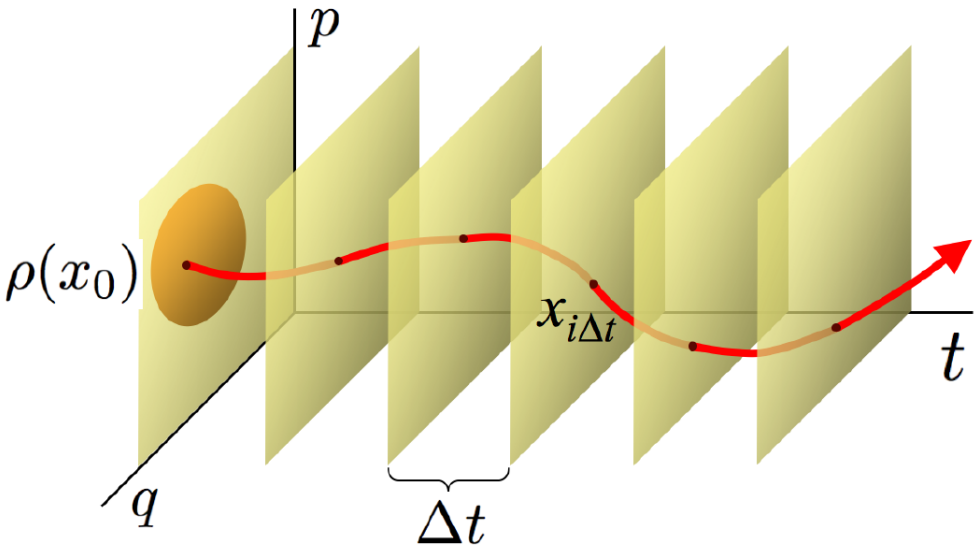

Figure 2.1: Illustration of a path. Figure taken from article of Dellago et al [5]

The path weight or probability of a path $x(\mathcal{T})$, $\mathcal{P}[x(\mathcal{T})]$, is given by

$$\mathcal{P}[x(\mathcal{T})] = \rho(x_0) \prod_{i=0}^{\frac{\mathcal{T}}{\Delta t}-1} \rho(x_{i\Delta t} \to x_{(i+1)\Delta t})/Z(\mathcal{T})$$

where $\rho(x_0)$ is the probability density of the initial conditions and $\rho(x_{i\Delta t} \to x_{(i+1)\Delta t})$ is the single time step transition probability.

2.2.2 Reactive Pathways

Now we are looking at the paths that start in the region A and end in the region B: the reactive pathways. In this ensemble, the path weight reads

$$\mathcal{P}_{AB}[x(\mathcal{T})] = h_A(x_0)\mathcal{P}[x(\mathcal{T})]h_B(x(\mathcal{T}))/Z_{AB}(\mathcal{T})$$

with

$$Z_{AB}(\mathcal{T}) = \int \mathcal{D}x(\mathcal{T})h_A(x_0)\mathcal{P}[x(\mathcal{T})]h_B(x(\mathcal{T}))$$

where $\int \mathcal{D}x(\mathcal{T})$ is a summation over all pathways.

2.2.3 Path Probabilities for Deterministic and Stochastic Dynamics

For our use case, the system is in contact with a heat bath, the right statistics to use is the canonical ensemble:

$$\rho(x) = \exp\left(-\beta\mathcal{H}(x)\right)/Z$$

with Z the partition function

Then we would like to use the Newtonian dynamics. The following relations hold:

$$\dot{r} = \frac{\partial\mathcal{H}(r,p)}{\partial p}$$

and

$$\dot{p} = -\frac{\partial\mathcal{H}(r,p)}{\partial r}$$

As the system is deterministic, its evolution is fully determined by the initial condition x_0, so we can define a function ϕ_t, called *the propagator of the system*, such that

$$x_t = \phi_t(x_0)$$

Using this function,

$$\rho(x_t \to x_{t+\Delta T}) = \delta[x_{t+\Delta t} - \phi_{\Delta t}(x_t)]$$

and finally

$$\mathcal{P}_{AB}[x(\mathcal{T})] = \rho(x_0)h_A(x_0)\prod \delta[x_{(i+1)\Delta t} - \phi_{\Delta t}(x_{i\Delta t})]h_B(x_{\mathcal{T}})/Z_{AB}(\mathcal{T})$$

with

$$Z_{AB}(\mathcal{T}) = \int dx_0 \rho(x_0)\mathcal{P}_{AB}[x(\mathcal{T})]h_A(x_0)h_B(x_{\mathcal{T}})$$

2.2.4 Defining the Stable States A and B

The TPS method doesn't require the reaction coordinate knowledge, but require a careful definition of the states A and B.

These states can be characterized by an *order parameter*, denoted q, however finding this order parameter is not trivial.

A good order parameter should be such that A and B are large enough

A and B should not overlap

2.3 Sampling Path Ensembles

We would like now to explore the path ensemble and find all reactive trajectories inside it.

2.3.1 Monte Carlo in Path Space

As said before, we would like to collect reactive trajectories $x(\mathcal{T})$ according to their weight $\mathcal{P}_{AB}[x(\mathcal{T})]$ in the transition path ensemble. Namely, we will use the Monte Carlo importance sampling: a new path $x^{(n)}(\mathcal{T})$ is generated from an old one $x^{(o)}(\mathcal{T})$ with a generating probability $P_{\text{gen}}[x^{(o)}(\mathcal{T}) \to x^{(n)}(\mathcal{T})]$.

The new path is then accepted with an acceptance probability $P_{\text{acc}}[x^{(o)}(\mathcal{T}) \rightarrow x^{(n)}(\mathcal{T})]$.

In order for pathways to be visited with a frequency proportional to the weight in the path ensemble $\mathcal{P}_{AB}[x(\mathcal{T})]$ (i.e maintain the path ensemble distribution), the detailed balance condition holds:

$$\begin{aligned}\mathcal{P}_{AB}[x(\mathcal{T})]P_{\text{gen}}[x^{(o)}(\mathcal{T}) \rightarrow x^{(n)}(\mathcal{T})]P_{\text{acc}}[x^{(o)}(\mathcal{T}) \rightarrow x^{(n)}(\mathcal{T})] = \\ \mathcal{P}_{AB}[x(\mathcal{T})]P_{\text{gen}}[x^{(o)}(\mathcal{T}) \rightarrow x^{(n)}(\mathcal{T})]P_{\text{acc}}[x^{(o)}(\mathcal{T}) \rightarrow x^{(n)}(\mathcal{T})]\end{aligned}$$

Since $h_A(x_0^{(o)}) = 1$ and $h_B(x_{\mathcal{T}}^{(o)}) = 1$ as the old path is reactive, using the definition of $\mathcal{P}_{AB}[x(\mathcal{T})]$ yields *the Metropolis-Hastings acceptance rule*:

$$P_{\text{acc}}[x^{(o)}(\mathcal{T}) \rightarrow x^{(n)}(\mathcal{T})] = h_A(x_0^{(n)})h_B(x_{\mathcal{T}}^{(n)}) \min\left[1, \frac{\mathcal{P}[x^{(n)}(\mathcal{T})]P_{\text{gen}}[x^{(n)}(\mathcal{T}) \rightarrow x^{(o)}(\mathcal{T})]}{\mathcal{P}[x^{(o)}(\mathcal{T})]P_{\text{gen}}[x^{(o)}(\mathcal{T}) \rightarrow x^{(n)}(\mathcal{T})]}\right]$$

2.3.2 Shooting Move

To generate a new path from an old one, the most common procedure used is the *shooting move*:

1. Select a time slice from the old path, $x_{t'}^{(o)}$
2. Randomly modify the momenta from $x_{t'}^{(o)}$ according to a Maxwell-Boltzmann distribution to get $x_{t'}^{(n)}$
3. From this new point, integrate forward to time $\mathcal{T}$ and backward to time 0

Using previous expressions, assuming microscopic reversibility and stationary and equilibrium distribution for the initial points, one can show that the acceptance probability reduces to

$$P_{\text{acc}}[x^{(o)}(\mathcal{T}) \rightarrow x^{(n)}(\mathcal{T})] = h_A(x_0^{(n)})h_B(x_{\mathcal{T}}^{(n)})$$

2.3.3 Shifting Move

In the *shifting move*, a new path is generated from the old one by translating the trajectory backward or forward. For the forward shift, a segment of a certain length is removed from the beginning of the path, then a segment of the same length is grown at the end of the path by integrating the equations of motion

This shifting move is computationally inexpensive and the new path partially overlaps the old path, leading to a highly correlated move.

One can show that the acceptance probability for this move reads

$$P_{\text{acc}}[x^{(o)}(\mathcal{T}) \to x^{(n)}(\mathcal{T})] = h_A(x_0^{(n)})h_B(x_{\mathcal{T}}^{(n)})$$

2.3.4 The Initial Pathway

To use the previous moves, an initial reactive pathway has to be available. Although using a plain MD simulation is still possible, it is computationally expensive as we are trying to get a rare event. A better solution is to produce an atypical trajectory with a low weight, then equilibrate the pathway towards more important regions of trajectory space: this can be done repeating MC steps and then begin the TPS simulation. Other solutions have been provided, such as running a high temperature MD and then run the TPS simulation at the temperature of interest, protein folding uses this process.

2.4 Reaction Mechanisms: Analyzing Transition Pathways

2.4.1 Rate constants

TPS can be used to compute reaction rate, and the reaction rate expression is obtained by comparing macroscopic and microscopic kinetic equations.

A time-dependent correlation function, $C(t)$, is then used to compute the reaction rate using TPS method, and reads as follows:

$$C(t) = \frac{\langle h_A(x_0) h_B(x_t) \rangle}{\langle h_A \rangle}$$

For $\tau_{mol} < t \ll \tau_{rxn}$, $C(t)$ is a linear function of time and the slope is given by k_{AB}. Methods such as Umbrella Sampling can be used to compute the correlation function, but this is beyond the scope of this work.

2.4.2 Reaction Coordinate vs Order Parameter

TPS requires a proper definition of the stable states A and B. Defining the stable states can be achieved using an *order parameter*, a variable that discriminate configurations belonging to A or B.

But this is not suitable for characterization of reaction mechanism, which would use the reaction coordinate (RC), a variable capable of describing the dynamical bottleneck separating the reactants from products A good RC should capture the essence of the dynamics and allow us to predict what a trajectory starting from a given configuration will most likely do: this is the committor.

2.4.3 Committor

In simple systems, transition state is described by finding the saddle points of the potential energy, which correspond to unstable states that can evolve to A and B in response to a perturbation. The *committor* is a generalization of this concept : it tells with which likelihood a certain configuration is committed to one of the two states. Thus, this is a direct statistical indicator of the progress of the reaction: the committor appears to be the ideal reaction coordinate. However, it is highly non trivial to get a small number of variables to parametrize the committor.

The committor of the state B at the time t is by definition

$$p_B(r,t) \equiv \frac{\int \mathcal{D}x(t)\mathcal{P}[x(t)]\delta(r_0 - r)h_B(x_t)}{\int \mathcal{D}x(t)\mathcal{P}[x(t)]\delta(r_0 - r)}$$

where r_0 is the initial configuration of the system from which the trajectory $x(t)$ is integrated.

Hence, the committor p_B is a statistical measure for how committed a given configuration is to the product state.

A value of $p_B = 0$ indicates no commitment to the state B whereas a value $p_B = 1$ shows a fully B committed configuration

2.4.4 Transition State Ensemble

A configuration r is a transition state if both states A and B are equally accessible from that configuration. This is equivalent to require that

$$p_A(r) = p_B(r)$$

Hence, the transition state defined in a statistical way is different from particular features of the potential energy (ex: first-order saddle points). Any r such that $p_A(r) = p_B(r)$ is the *separatrix*, also called *isocommittor surface*. If all trajectories end up in state A or B, then the transition state is defined by $p_A(r) = p_B(r) = 0.5$

2.5 Transition Interface Sampling

The main change using transition interface sampling is the introduction of overall states[7] $\mathcal{A}$ and $\mathcal{B}$ instead of regular ones, A and B. These overall states are complementary.

$\mathcal{A}$ includes all phase points in A and all phase points that were most recently in A than in B based on their history.

$\mathcal{B}$ includes all phase points in B and all phase points that were most recently in B than in A based on their history.

Hence, $h_{\mathcal{A}}$ and $h_{\mathcal{B}}$, the characteristic function of $\mathcal{A}$ and $\mathcal{B}$, are not very sensitive to the stable state definition.

Now, the time-dependent correlation function is:

$$C(t) = \frac{< h_{\mathcal{A}}(x_0) h_{\mathcal{B}}(x_t) >}{< h_{\mathcal{A}} >}$$

Recrossings of the phase space hypersurface separating the overall state are eliminated and k_{AB} is the slope of $C(t)$ at time 0:

$$k_{AB} = \frac{< h_{\mathcal{A}}(x_0) \dot{h_{\mathcal{B}}}(x_0) >}{< h_{\mathcal{A}} >}$$

This expression can be further rewritten to:

$$k_{AB} = \frac{< \phi_{AB} >}{< h_{\mathcal{A}} >}$$

TIS methods also aim at computing the rate constant k_{AB} for the reaction $A \to B$. One can prove that:

$$k_{AB} = f_A \mathcal{P}_A(\lambda_B \mid \lambda_A)$$

where f_A is the initial flux: it measures how often trajectories start off at the foot of the reaction barrier from the reaction side λ_A and $\mathcal{P}_A(\lambda_B \mid \lambda_A)$ is the crossing probability: it's the probability of reaching λ_B before λ_A given λ_A has just been crossed.

One major issue is that $\mathcal{P}_A(\lambda_B \mid \lambda_A)$ is very small and therefore impossible to compute without having to wait for a long time. One solution is the TIS method: we set n non-intersecting interfaces λ_i, called interfaces, with $\lambda_0 = \lambda_A$ and $\lambda_N = \lambda_B$. So, now,

$$k_{AB} = f_A \prod_{i=0}^{N} P_A(\lambda_{i+1} \mid \lambda_i)$$

with

$$\mathcal{P}_A(\lambda_{i+1} \mid \lambda_i)$$

the probability of a path crossing λ_{i+1} given it started from λ_A, ended in λ_A or λ_B, and at least crossed λ_i in the past.

λ_i defines the path ensemble $[i^+]$: it includes all the trajectories that start at λ_A, end in λ_A or in λ_B, and reached λ_i at some point: with these ensembles we can compute $\mathcal{P}_A(\lambda_{i+1} \mid \lambda_i)$, this is the fraction of paths in the $[i^+]$ ensemble that cross λ_{i+1}. To generate trajectories, the shooting move as seen before can be used as well as a new move: the time reversal move.

- Shooting move

    1. Pick a random step from the MD simulation in the current trajectory
    2. Modify the velocities at this phase point
    3. Generate a new trajectory from this point by integrating backward and forward until A or B is reached
    4. We accept the new trajectory if:

       - The detailed balance condition is fulfilled (the maximum allowed length path to obey the detailed balance is equal to the length of the old path length divided by a random number between 0 and 1 drawn from a uniform distribution)

       - It starts at A

       - It has at least one crossing with λ_i before ending in A or B

- In a time-reversal move, a new trajectory is generated by changing the time direction of the current path.

The shooting move is more efficient than the method to generate trajectories in the classical TPS.

2.5.1 Replica Exchange TIS

The RETIS approach[4] is similar to the TIS one, except that we include a new move, called the swapping move, and a new ensemble, $[0^-]$, which describes trajectories that explore the reactant state. The swapping is described as follows: if, in two different path ensembles $[(i-1)^+]$ and $[i^+]$, two valid trajectories are valid for each other's ensemble, we can swap these trajectories, i.e the trajectory currently belonging to $[(i-1)^+]$ now belongs to $[i^+]$ and vice versa.

The shooting move is the most time consuming because it requires force calculations to do the MD steps, the least time consuming are time reverse and swapping moves. Moreover, swapping move can efficiently decorrelate consecutive paths.

2.5.2 Analyze complex mechanisms using RETIS

The use of a committor function is, as described above, considered as the best reaction coordinate. However, computing the committor is very expensive and find helpful chemical insights about the reaction is a non-trivial task.

The following approach[6] considers the reaction coordinate as a simple order parameter, but considers also in addition other variables, called collective variables.

$\lambda(x)$ is a progress coordinate, a function of phase-space point x, it could be the length of a bond that needs to be broken or radius of gyration for protein folding. The collection of phase-space points x having a specific value λ_i form the interfaces. There are $M+1$ interfaces: $\lambda_0 = \lambda^A$ is placed within the reactant well, $\lambda_M = \lambda^B$ is placed within the product well and the interfaces in between are placed in the barrier region.

From now on, X denotes a path of $L+1$ time slices: $X = \{x_0, x_1, \ldots, x_L\}$: L is the flexible path length and x_k is the k-th phase point of the path

$X \in [i^+]$ if $\lambda(x_0) < \lambda_0$, $\lambda(x_L) < \lambda_0$ or $\lambda(x_L) > \lambda_M$, $\lambda_0 < \lambda(x_k) < \lambda_M$ for $k = 1, 2, \ldots, L-1$ and $\lambda_{max} \equiv \max[\lambda(x_1), \lambda(x_2), \ldots, \lambda(x_L)] > \lambda_i$. x^{λ^c} is the first crossing point with interface λ^c: $x^{\lambda^c}(X) = x_k \in X$ if $\lambda(x_k) \geq \lambda^c$ and $\lambda(x_l) < \lambda^c$ for all $l < k$.

Naturally $\lambda(x^{\lambda^c}(X)) \geq \lambda_c$, but there are other collective variables (CVs) that can characterize this point: $\Psi_1, \Psi_2, \ldots, \Psi_N$. We denote $\Psi^N(x) = \{\Psi_1(x), \Psi_2(x), \ldots, \Psi_N(x)\}$ the vector corresponding to the N CVs.

Now, we consider 3 important interfaces: λ^A: the reactant interface, $\lambda^c > \lambda^A$: the crossing interface, and $\lambda^r > \lambda^c$: the partial reaction interface, that characterizes the reactive and unreactive trajectories: the reactive ones cross λ^r, the unreactive recross λ^A without crossing λ^r. If $\lambda^r = \lambda^B$, reactive trajectories are fully reactive, and for $\lambda^r < \lambda^B$, we can get useful information about the reaction mechanism at intermediate stages of the reaction. We can shift λ^c and λ^r to the desired region and construct a grid in the CVs space Ψ^N and define bins covering the full accessible surface of the λ^c interface.

q is the index of the bins, t_q is the fraction of all crossing trajectories passing through bin q in the λ^c interface, r_q is the fraction of all crossing trajectories passing through bin q and crossing λ^r, and u_q is the fraction of all crossing trajectories passing through bin q and not crossing λ^r.

The following relations hold:

$$t_q = u_q + r_q$$

$$\sum_q t_q = 1$$

$$\sum_q r_q = \mathcal{P}_A(\lambda^r \mid \lambda^c)$$

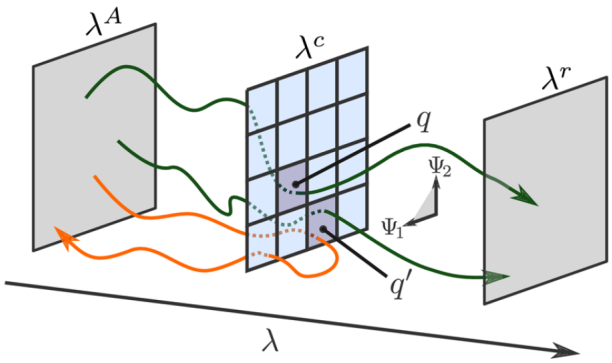

Figure 2.2: Illustration of reactive and unreactive trajectories passing through bins with two collective variables Ψ_1 and Ψ_2. Figure taken from article of van Erp et al [6].

$$\sum_q u_q = 1 - \mathcal{P}_A(\lambda^r \mid \lambda^c)$$

Depending on the CVs and the grid spacing, we would get different values of r_q and u_q. If we could partition the first crossing points such that $\frac{r_q}{t_q} = 1$ or $r_q = 0$ for each bin, the predictive ability is optimal; each time that λ^c is crossed for the first time, we check through which bin it passes and, then, we would be able to say whether it will cross λ^r or not. But this is not achievable as it is too difficult to find the right CVs.

For each bin q, the reactive ratio $\frac{r_q}{t_q}$ can have a value between 0 and 1. The overall measure of predictive power must then be a weighted average of $\frac{r_q}{t_q}$ over q: This measure must reflect the fraction of reactive trajectories of the bin q over the total number of reactive trajectories: the measure $\mathcal{T}$ is defined as follow:

$$\mathcal{T} \equiv \sum_q \left(\frac{r_q}{\sum_v r_v} \right) \frac{r_q}{t_q}$$

We can rewrite using the properties mentioned above:

$$\mathcal{T} = 1 - \frac{1}{\mathcal{P}_A(\lambda^r \mid \lambda^c)} \sum_q \frac{r_q u_q}{t_q} \equiv 1 - \mathcal{S}$$

In continuous space, $\mathcal{S}$ is the overlap integral of the reactive and nonreactive distributions:

$$\mathcal{S}_A^{\lambda^c,\lambda^r}[\Psi^N] = \frac{1}{\mathcal{P}_A(\lambda^r \mid \lambda^c)} \int \left(\frac{r^{\lambda^c,\lambda^r}(\Psi^N) u^{\lambda^c,\lambda^r}(\Psi^N)}{t^{\lambda^c}(\Psi^N)} \right) d\Psi^N$$

The overlap depends on the CVs chosen, which are functions of phase-space point x. The goal is to minimise $\mathcal{S}$:

$$\mathcal{S}_{A,0}^{\lambda^c,\lambda^r}[\Psi^N] = \frac{1}{\mathcal{P}_A(\lambda^r \mid \lambda^c)} \min_{\Psi^N} \left[\int \left(\frac{r^{\lambda^c,\lambda^r}(\Psi^N) u^{\lambda^c,\lambda^r}(\Psi^N)}{t^{\lambda^c}(\Psi^N)} \right) d\Psi^N \right]$$

and we call the corresponding CVs Ψ_{min}^N such that

$$S_A^{\lambda^c,\lambda^r}[\Psi_{min}^N] = S_{A,0}^{\lambda^c,\lambda^r}$$

Ψ_{min}^N is not unique: the aim is to minimize $S_A^{\lambda^c,\lambda^r}$ and to get intuitive properties to find out how to drive the reaction from the reactants to the products.

$\mathcal{T}_A^{\lambda^c,\lambda^r}$ is a measure of predictive capacity:

$$\mathcal{T}_A^{\lambda^c,\lambda^r} \leq 1$$

$\dfrac{\mathcal{T}_A^{\lambda^c,\lambda^r}}{\mathcal{P}(\lambda^r \mid \lambda^c)}$ is a measure of the enhancement of predictive capacity due to the information on the selected CVs:

$$\frac{\mathcal{T}_A^{\lambda^c,\lambda^r}}{\mathcal{P}(\lambda^r \mid \lambda^c)} \geq 1$$

With path sampling data, one can compute $S_A^{\lambda^c,\lambda^r}(\Psi^N)$ for different λ^c and λ^r between λ^A and λ^B and so we can get information information about the predictive power of the CVs at each stage of the reaction.

Practically, the results from all path ensembles $[i^+]$ are merged together using the weighted histogram analysis method (WHAM)[9, 13]. In substance, WHAM enables merging all path ensembles data $[0^+]$, $[1^+]$, ..., $[(M-1)^+]$ to compute $\mathcal{T}_A^{\lambda^c,\lambda^r}$ in a way to reduce statistical errors and being able to use values for λ^c and λ^r that are not being part of the interface values.

Chapter 3

Enhance the density probability estimation of reactive and nonreactive distributions R and U using non parametric kernel density estimation

3.1 Motivation

Recall from the previous part that we set up a grid in the CVs space using bins, so we have to define a bin width. Following a discussion with Titus Van Erp and Anders Lervik, finding the right bin width is challenging. It has to be carefully chosen by the user in order to process mechanism analysis. This raises some issues:

- if the bin size is too small, the situation where too many bins are empty can happen, leading to falsely non-overlapping R and U distributions.
- if the bin size is too large, the resolution is not sufficient enough to

compute the overlap integral.

Hence, when choosing the bin size, a trade-off between acceptable accuracy and wrong R and U distributions. This requires a priori knowledge about the collective variable used for the analysis and the studied reaction, which can be challenging to get, even if impossible. When studying a reaction, many tests are required to check if a slight variation of the bin width dramatically changes R and T distributions.

Moreover, this collective variable analysis could be integrated into a future update of PyRETIS, and since this library globally aims at being usable by non-expert users, not having to choose this parameter is a significant advantage.

3.2 General framework

Estimation of a density probability function given a discrete set of points has always been a tedious task. Two approaches have been mainly studied in the literature: the parametric and the non-parametric methods. In the parametric approach, one assumes a distribution shape described by a set of parameters (i.e., a Gaussian distribution is described using two parameters, the mean and the standard deviation) and then tries to fit this given distribution to sampled data. In contrast, the non-parametric approach does not require assumptions about the density function.

The most straightforward method is to plot a histogram, but more complex and robust methods have been developed, such as the kernel density estimation (KDE). This method evaluates the proper density by using a sum of kernel functions which are centered on the data points:

$$f_{KDE}(x) = \frac{1}{hN} \sum_{j=1}^{N} K\left(\frac{x - X_j}{h}\right)$$

where K is the kernel function and h is the bandwidth.

Usually, non-parametric methods require the user to choose parameters (bin size in case of a histogram, bandwidth in case of KDE) in order for the method to be successful. Most of these parameters are used to smooth the density estimation. It would be beneficial for the user to have the least number of parameters involved in a method or have an automatic procedure for choosing these parameters. Bernacchia et al.[2] developed a self-consistent method that gives an optimal density function.

An optimal convolution kernel can be expressed as a function of the power spectrum of the density to be estimated, the optimization is made by minimizing the mean integrated square error. The estimate is first expressed as follows:

$$f(x) = \frac{1}{N}\sum_{j=1}^{N} K(x - X_j)$$

.

The Fourier transform of the optimal kernel is given by

$$\kappa_{opt}(t) = \frac{N}{N - 1 + \mid \phi(t) \mid^{-2}}$$

where $\phi(t)$ is the characteristic function of the true density $f(x)$, i.e the Fourier transform:

$$\phi(t) = \int_{-\infty}^{+\infty} \exp(itx) f(x) dx$$

Obviously, the true density is unknown, so is the characteristic function and this procedure is not usable. Self consistent method has been developed to get rid of this issue.

The Fourier transform of the true density can be written as follows:

$$\phi(t) = \Delta(t)\kappa_{opt}(t) = \frac{\Delta(t)N}{N - 1 + \mid \phi(t) \mid^{-2}}$$

where $\Delta(t) = \frac{1}{N}\sum_{j=1}^{N} \exp(itX_j)$

The iterative procedure uses a first estimation, ϕ_0, and this estimation yields ϕ_1. Then we can find an improved estimate ϕ_2 using a kernel which is optimal for ϕ_1, and so on so forth. More formally we try to find the fixed point for the following suite defined by:

$$\hat{\phi}_{n+1} = \Delta(t)\kappa_{opt}(t) = \frac{\Delta(t)N}{N - 1 + \mid \hat{\phi}_n \mid^{-2}}$$

and the fixed point is by definition:

$$\hat{\phi}_{sc} = \frac{\Delta N}{N - 1 + \mid \hat{\phi}_{sc} \mid^{-2}}$$

It can be shown that:

$$\hat{\phi}_{sc} = \frac{N\Delta(t)}{2(N-1)}[1 + \sqrt{1 - \frac{4(N-1)}{N^2|\Delta(t)|^2}}]I_A(t)$$

where $I_A(t)$ is the indicator function that equals to 1 if $t \in A$ and 0 otherwise, and A is the set of accepted frequencies, i.e frequencies giving a non-zero contribution to the estimate. To be a stable solution, the set A must be contained in B where $t \in B$ if and only if

$$|\Delta(t)|^2 \geq \frac{4(N-1)}{N^2}$$

.

This condition sets a threshold below which $\hat{\phi}_{sc}(t) = 0$.

The choice of A is still at the user's discretion, but one can prove that any bounded set A where the bound grows with N will make the estimate converge to the true density. Hence, the authors used a default value for A that works in many cases.

The inverse Fourier transformation of $\hat{\phi}_{sc}$ will lead to the estimate density $\hat{f}_{sc}$.

However, the method implemented by Bernacchia et al. has not been applied to multidimensional KDEs. O'Brien et al.[12] augmented the original method to multivariate KDEs, called *fastKDE*. In essence, the theory lying behind fastKDE is the same as before, but it relies on optimized Fourier computations to achieve fast computations.

This method is closely related to histograms and does not interrupt the already existing workflow of PyRETIS analysis scripts, resulting in minor modifications in the codebase.

The code was implemented in Python 3.8, using Numpy[10] and built-in Python functions. The fastKDE method has been used using the fastKDE package available released by the authors at `https://pypi.org/project/fastkde/`.

Chapter 4

Results and discussion

4.1 Kernel Density Estimation

4.1.1 Toy model using Gaussian distributions

To investigate the possibility of using the fastKDE method on PyRETIS data, we used a model made up of Gaussian distributions, faking R and U distributions. Different values of μ and σ were used to simulate different cases. Two reasons lead us to use Gaussian as testing distributions:

- The ability to compute an exact value for the overlap coefficient between two Gaussian distributions.
- The ability to shape fake R and U distributions to mimic real-world counterparts' behavior.

The central quantity is the overlap coefficient between two distributions, which is defined using a simplified notation as follows:

$$\mathcal{S}[\Psi^N] = \int \frac{R(\Psi^N) \times U(\Psi^N)}{R(\Psi^N) + U(\Psi^N)} d\Psi^N$$

The current issue with density estimation is that, in the desired case

where R and U distributions have a slight overlap value, i.e., the chosen CVs are highly discriminating, computing these values is tricky.

The number of collective variables in Ψ^N is usually low, for instance, 2, 3, or 4, as we would like to explain the outcome of a reaction using a small number of CVs[11].

The first step was to use 1D Gaussian distributions as fake R and U distributions:

$$R(\Psi) = \frac{1}{\sigma_R\sqrt{2\pi}} \exp\left(-\frac{(\Psi - \mu_R)^2}{2\sigma_R^2}\right)$$

and

$$U(\Psi) = \frac{1}{\sigma_U\sqrt{2\pi}} \exp\left(-\frac{(\Psi - \mu_U)^2}{2\sigma_U^2}\right)$$

We define the mean difference to be the difference of the mean value of the R distribution and the U distribution and is equal to $\mu_R - \mu_D$. Similarly, we define the standard deviation ratio to be the ratio of the standard deviation value of the R distribution and the U distribution and is equal to $\frac{\sigma_R}{\sigma_U}$.

In a real-case situation, as in [11], the reactive distribution can be of several order of magnitude less than the unreactive distribution (10^7 for instance) and shifted away from the unreactive distribution. We can translate these properties of the real R and U distribution into the faked counterparts by adjusting the mean and the standard deviation.

The overlap value was computed with a mean difference value ranging from 0 to 5, simulating the different cases in a real dataset. The number of generated points drawn from Gaussian distribution is equal to 10000. Numeric integration was then carried out using the integrate.quad function from the scipy module.

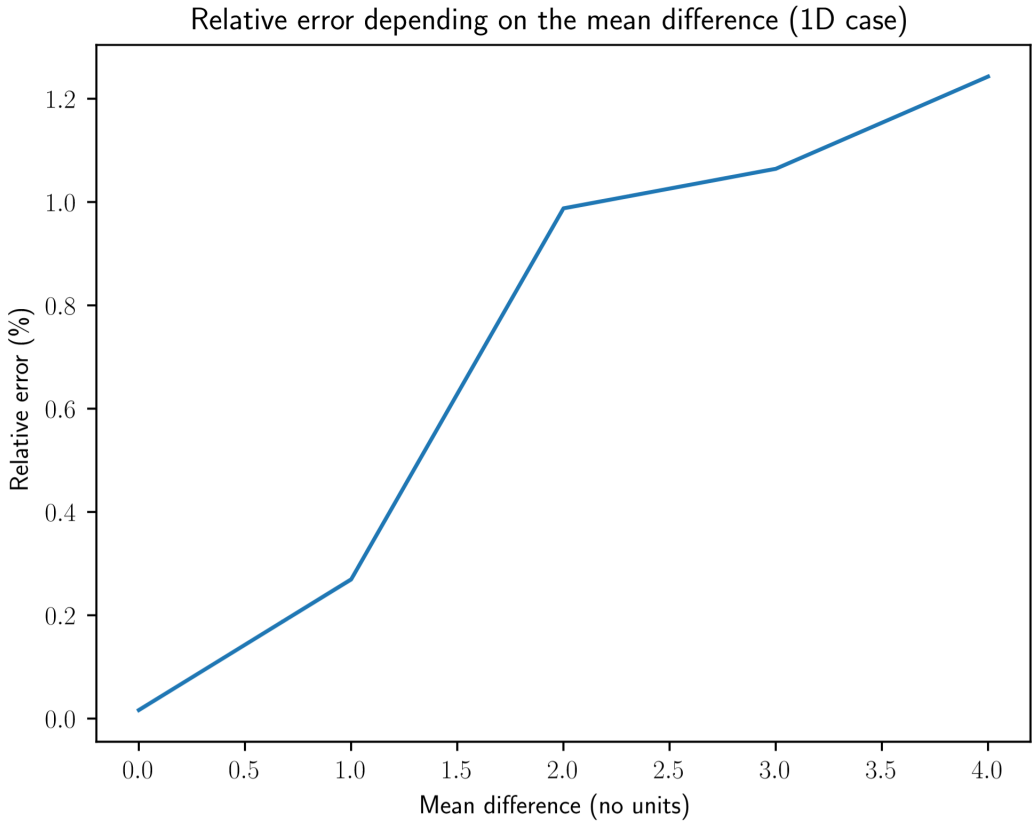

Figure 4.1: Overlap value depending on the mean difference of the two distributions (1D case)

It can be seen that the relative error ranges from 0% to 1.2% (with a mean difference equal to 4).

The same was done using this time a difference in standard deviation values. The overlap value was computed with a standard deviation ratio ranging from 1 to 10000, simulating the different cases in a real dataset. The number of generated points drawn from Gaussian distribution is equal to 10000. Numeric integration was then carried out on estimated densities using the integrate.quad function from the scipy module.

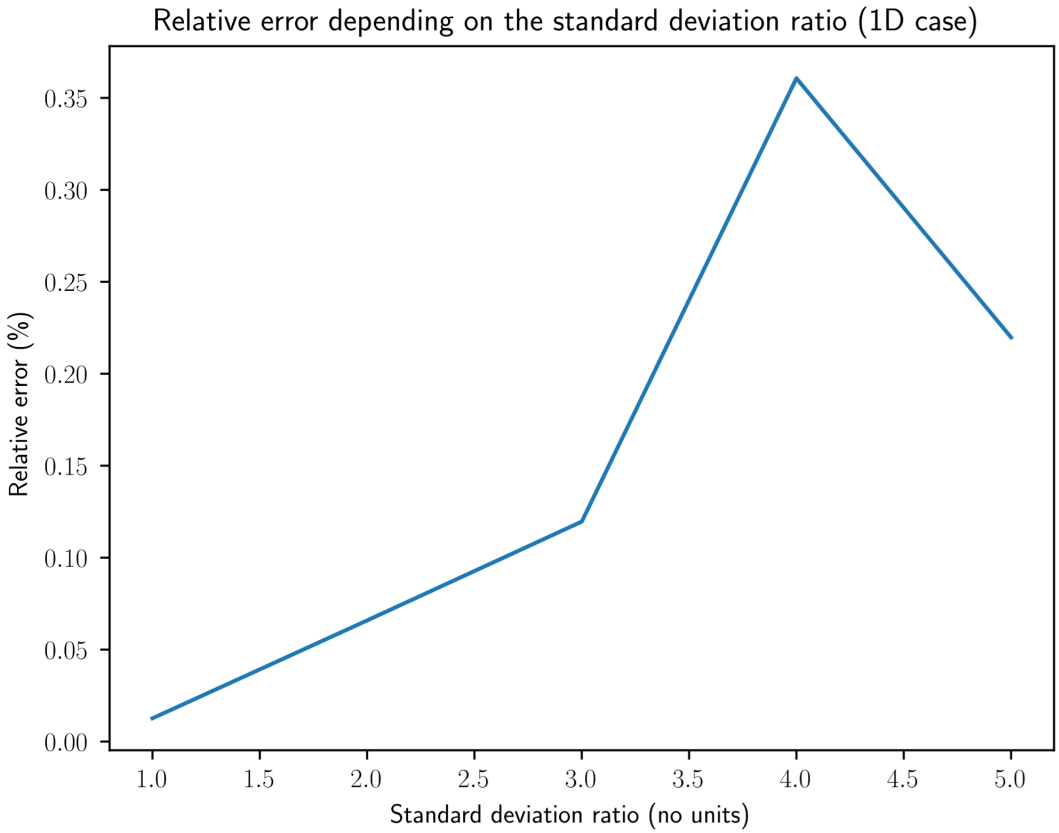

Figure 4.2: Overlap value depending on the standard deviation ratio of the two distributions (1D case)

It can be seen that the relative error ranges from 0% (i.e the two distributions are the same) to 100% (with a standard deviation ratio equal to 10000). We switched over to 2D Gaussian distributions and used the same process.

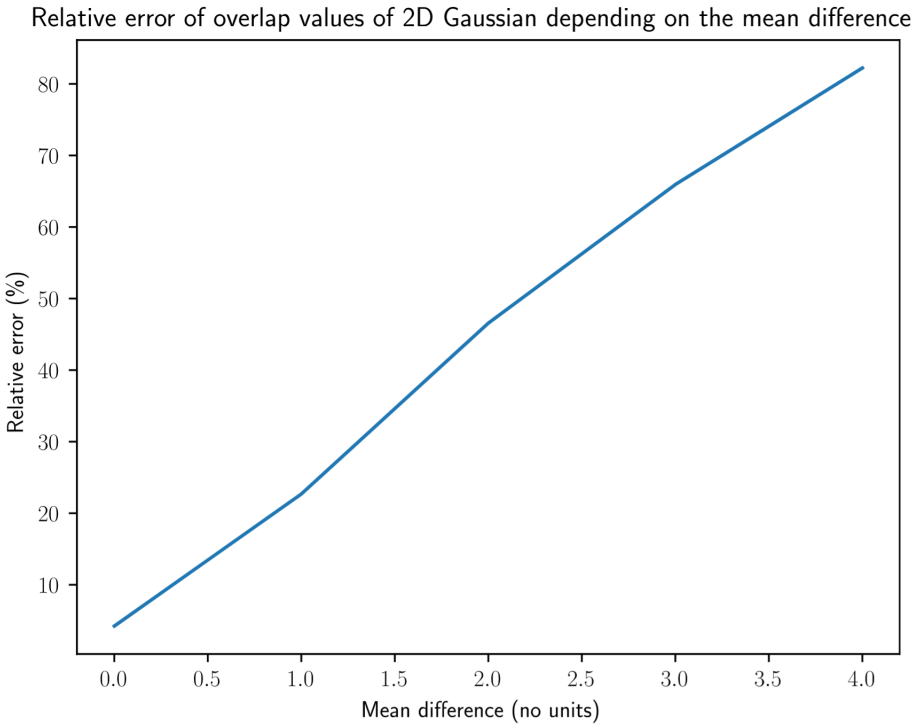

Figure 4.3: Overlap value depending on the mean difference of the two distributions (2D case)

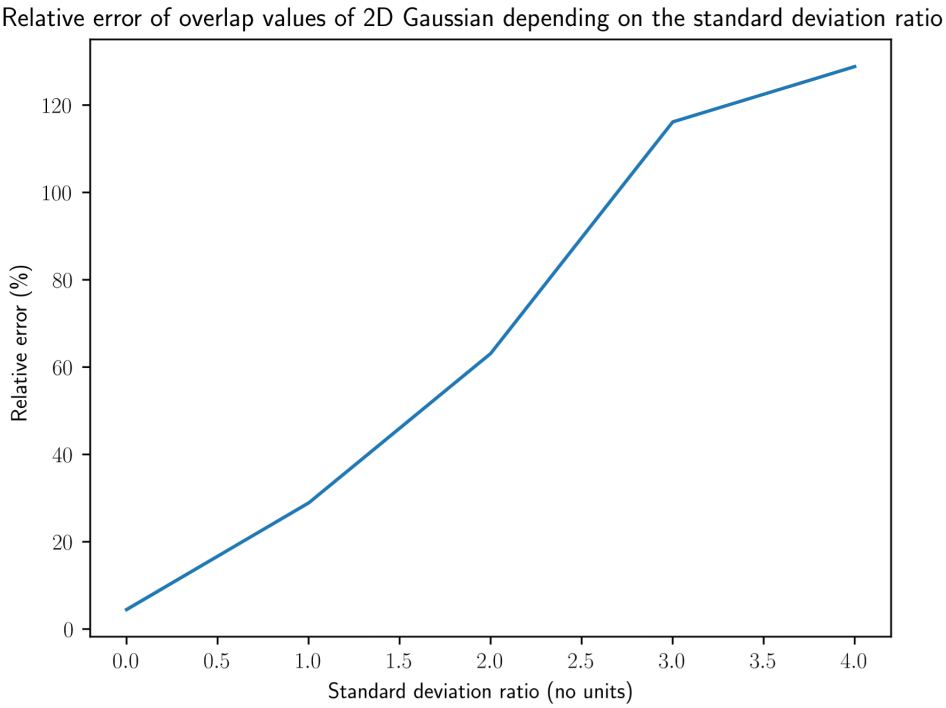

Figure 4.4: Overlap value depending on the standard deviation ratio of the two distributions (2D case)

When the number of dimensions increases, the relative error dramatically increases. The following section will discuss the obtained results here.

4.2 Discussion

The general framework around the fastKDE method can be seen as the framework around most methods for estimating density, in the sense that the estimated density is a sum of kernels. Hence, testing this method enables us to confirm the interest for kernel density estimation to be integrated into PyRETIS or indicates a no-go and that we should not put further efforts into the investigation of a similar method.

From the results above, using toy models made up of Gaussian distributions leads to the conclusion that fastKDE methods will likely not be suitable to use for real-case PyRETIS data. Indeed, analyzing the results shows a dramatic loss of precision using the fastKDE method. The histogram method has already led to some interesting practical results, for instance in [11].

Further analysis led to a possible issue with fastKDE: it is optimized to reduce the mean integrated squared error, which might not be the optimum for computing overlaps as done in the predictive capacity method. Therefore, method improvements could aim for optimizing the bandwidth based on other criteria.

Chapter 5

Future work

Even if the method is not really successful on the toy system, it would be interesting to work with a real system, as the water dataset. It could be that the numerical difficulties that we tested are not representative of an actual system, and KDE might still do equally well or better for such a realistic case. The Python scripts written for this purpose are still under active development at the time of finalizing this work and ongoing work includes applying the fastKDE method to another RETIS simulation that has been running in the past year.

Currently, the WHAM method is the statistical tool used to combine all path ensembles and compute histograms with reduced statistical errors. Investigating more into weighted analysis could be a key to find a more sophisticated and accurate version.Ongoing work includes applying the fastKDE method to a water dataset similar to the one used in [11] and another RETIS simulation running in the past year.

Chapter 6

Conclusion

In this work, we tried to apply the fastKDE method, a parameter-free density estimation, to estimate fake distributions simulating reactive R and unreactive U distributions. Having applied the fastKDE method only on toy models but simulating real cases, the conclusion is partial but gives a strong signal that these kind of methods are challenging to apply to this kind of computations where we would like to compute a tiny quantity resulting from the overlap of two distributions in regions with a small number of data points. The WHAM is still the method of choice and the best one available at the moment.

Bibliography

[1] C.H Bennett. *Algorithms for Chemical Computations (ACS Symposium, Series No. 46) ed R Christofferson*. 1977.

[2] Alberto Bernacchia and Simone Pigolotti. "Self-consistent method for density estimation". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73.3 (Apr. 2011), pp. 407–422. ISSN: 1369-7412. DOI: 10.1111/j.1467-9868.2011.00772.x. URL: http://dx.doi.org/10.1111/j.1467-9868.2011.00772.x.

[3] Peter G. Bolhuis, David Chandler, Christoph Dellago, and Phillip L. Geissler. "TRANSITION PATH SAMPLING: Throwing Ropes Over Rough Mountain Passes, in the Dark". In: *Annual Review of Physical Chemistry* 53.1 (2002). PMID: 11972010, pp. 291–318. DOI: 10.1146/annurev.physchem.53.082301.113146. eprint: https://doi.org/10.1146/annurev.physchem.53.082301.113146. URL: https://doi.org/10.1146/annurev.physchem.53.082301.113146.

[4] Raffaela Cabriolu, Kristin M. Skjelbred Refsnes, Peter G. Bolhuis, and Titus S. van Erp. "Foundations and latest advances in replica exchange transition interface sampling". In: *The Journal of Chemical Physics* 147.15 (2017), p. 152722. DOI: 10.1063/1.4989844. eprint: https://doi.org/10.1063/1.4989844. URL: https://doi.org/10.1063/1.4989844.

[5] Christoph Dellago, Peter G. Bolhuis, and Phillip L. Geissler. "Transition Path Sampling". In: *Advances in Chemical Physics*. John Wiley & Sons, Ltd, 2003. Chap. 1, pp. 1–78. ISBN: 9780471231509. DOI: 10.1002/0471231509.ch1. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/0471231509.ch1. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/0471231509.ch1.

[6] Titus S. van Erp, Mahmoud Moqadam, Enrico Riccardi, and Anders Lervik. "Analyzing Complex Reaction Mechanisms Using Path Sampling". In: *Journal of Chemical Theory and Computation* 12.11 (2016). PMID: 27732782, pp. 5398–5410. DOI: 10.1021/acs.jctc.6b00642. eprint: https://doi.org/10.1021/acs.jctc.6b00642. URL: https://doi.org/10.1021/acs.jctc.6b00642.

[7] Titus S. van Erp, Daniele Moroni, and Peter G. Bolhuis. "A novel path sampling method for the calculation of rate constants". In: *The Journal of Chemical Physics* 118.17 (2003), pp. 7762–7774. DOI: 10.1063/1.1562614. eprint: https://doi.org/10.1063/1.1562614. URL: https://doi.org/10.1063/1.1562614.

[8] Titus S. [van Erp] and Peter G. Bolhuis. "Elaborating transition interface sampling methods". In: *Journal of Computational Physics* 205.1 (2005), pp. 157–181. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2004.11.003. URL: http://www.sciencedirect.com/science/article/pii/S0021999104004620.

[9] Alan M. Ferrenberg and Robert H. Swendsen. "Optimized Monte Carlo data analysis". In: *Phys. Rev. Lett.* 63 (12 Sept. 1989), pp. 1195–1198. DOI: 10.1103/PhysRevLett.63.1195. URL: https://link.aps.org/doi/10.1103/PhysRevLett.63.1195.

[10] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus,

Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[11] Mahmoud Moqadam, Anders Lervik, Enrico Riccardi, Vishwesh Venkatraman, Bjørn Kåre Alsberg, and Titus S. van Erp. "Local initiation conditions for water autoionization". In: *Proceedings of the National Academy of Sciences* 115.20 (2018), E4569–E4576. ISSN: 0027-8424. DOI: 10.1073/pnas.1714070115. eprint: https://www.pnas.org/content/115/20/E4569.full.pdf. URL: https://www.pnas.org/content/115/20/E4569.

[12] Travis A. O'Brien, Karthik Kashinath, Nicholas R. Cavanaugh, William D. Collins, and John P. O'Brien. "A fast and objective multidimensional kernel density estimation method: fastKDE". In: *Computational Statistics & Data Analysis* 101 (2016), pp. 148–160. ISSN: 0167-9473. DOI: https://doi.org/10.1016/j.csda.2016.02.014. URL: https://www.sciencedirect.com/science/article/pii/S0167947316300408.

[13] Benoıt Roux. "The calculation of the potential of mean force using computer simulations". In: *Computer Physics Communications* 91.1 (Sept. 1995), pp. 275–282. DOI: 10.1016/0010-4655(95)00053-I.