DEVELOPMENT OF ENHANCED SAMPLING METHODS FOR MOLECULAR SIMULATIONS

—

WAVELETS AND BIRTH-DEATH

Dissertation zur Erlangung des Grades
“**Doktor der Naturwissenschaften**”
am Fachbereich Physik, Mathematik und Informatik
der Johannes Gutenberg-Universität Mainz

Mainz, 2022

BENJAMIN PAMPEL
geboren in Nürnberg

Arbeitskreis Theorie der Polymere
am Max-Planck-Institut für Polymerforschung Mainz

Für L.

1. Gutachter: [redacted]
2. Gutachterin: [redacted]

Tag der Einreichung: 01. März 2022
Tag der Prüfung: 09. Juni 2022

ABSTRACT

To overcome the time scale problem of molecular simulations, methods that enhance phase space sampling are developed. Two complementary enhanced sampling approaches are investigated: Improving the bias representation in the variationally enhanced sampling method, as well as a novel sampling algorithm using birth-death moves.

The variationally enhanced sampling method is based on a variational principle, where a bias potential is constructed in the space of a few slow degrees of freedom by minimizing a convex functional. Typically, the bias potential is taken as a linear expansion in some set of basis functions. In this work, properties of good basis functions for the method are identified to subsequently propose new basis functions and assess their performance. In particular, Daubechies wavelets are investigated, which construct orthogonal and localized bases that exhibit an attractive multiresolution property. Their theory is studied and they are implemented into the PLUMED2 software, together with other new basis functions. The parameters of the new basis sets are tuned. Benchmarking studies on systems of increasing complexity are performed, from the simulation of the movement of a single particle in a one-dimensional potential to the study of the association process of calcium carbonate in water. The wavelet bases are found to exhibit excellent performance and yield much better convergence of the bias potential than the previously existing basis functions.

Also, a novel sampling algorithm that augments Langevin dynamics with birth-death moves is investigated. This is a modification of a previously proposed algorithm that provides an approximation of a stochastic birth-death process for a particle-based implementation. The method connects multiple parallel Langevin dynamics simulations of the same system with a birth-death scheme to facilitate global sampling according to the equilibrium distribution.

The algorithm is investigated theoretically, implemented into a custom molecular simulation code, and tested via numerical simulations. The behavior of the algorithm under change of parameters is investigated. In this process, the desired sampling is observed for all tested systems. It is found that the performance of the method is independent of the intrinsic time scales and barriers of the system, which is favorable for systems with processes on long time scales.

ZUSAMMENFASSUNG

Es werden Methoden entwickelt die besseres Sampling des Phasenraums ermöglichen um Zeitskalenprobleme bei molekularen Simulationen zu überwinden. Zwei komplementäre Enhanced-Sampling-Ansätze werden untersucht: Eine verbesserte Darstellung des sogenannten Bias-Potentials in der Variationally-Enhanced-Sampling-Methode, sowie ein neuartiger Sampling-Algorithmus, der Birth-Death-Schritte verwendet.

Die Variationally-Enhanced-Sampling-Methode basiert auf einem Variationsprinzip, bei dem ein zusätzliches Potential durch Minimierung eines konvexen Funktionals konstruiert wird. Dieses Bias-Potential verbessert das Sampling im Unterraum einiger langsamer Freiheitsgrade und wird typischerweise mittels einer Linearkombination von Basisfunktionen dargestellt. In dieser Arbeit werden die Eigenschaften guter Basisfunktionen für die Methode ermittelt, um darauf aufbauend neue Basisfunktionen vorzuschlagen und deren Verhalten zu evaluieren. Insbesondere werden die von Daubechies entwickelten Wavelets untersucht, aus denen orthogonale und lokalisierte Basen konstruiert werden können, die zudem eine attraktive Mehrskalen-Eigenschaft aufweisen. Ihre Theorie wird untersucht und sie werden zusammen mit anderen neuen Basisfunktionen in die PLUMED2-Software implementiert. Optimale Parameter der neuen Basissätze werden empirisch bestimmt. Es werden Benchmarking-Studien an Systemen mit zunehmender Komplexität durchgeführt, von der Simulation der Bewegung eines Teilchens in einem eindimensionalen Potential, bis hin zur Untersuchung des Assoziationsprozesses von Calziumcarbonat in Wasser. Es wird festgestellt, dass die Wavelet-Basen ausgezeichnete Ergebnisse produzieren. Insbesondere wird deutlich bessere Konvergenz des Bias-Potenzials beobachtet als mit zuvor gebräuchlichen Basisfunktionen.

Weiterhin wird eine neuartige Methode entwickelt, die Langevin-Dynamik-Simulationen mit Birth-Death-Schritten ergänzt. Dabei handelt es sich um eine Modifikation eines zuvor vorgeschlagenen Algorithmus, welcher eine mathematische Beschreibung eines stochastischen Birth-Death-Prozess mittels einer partikelbasierte Implementierung als Annäherung annähert. Die Methode verbindet mehrere parallele Langevin-Dynamik-Simulationen desselben Systems mit einem Birth-Death-Schema, um das globale Sampling gemäß der Gleichgewichtsverteilung zu verbessern. Der Algorithmus wird theoretisch untersucht, in einen eigens angefertigten molekularen Simulationscode implementiert und durch numerische Simulationen getestet. Das Verhalten des Algorithmus unter Variation verschiedener Parameter wird untersucht. Das gewünschte Sampling wird dabei für alle getesteten Systeme beobachtet. Es wird zudem festgestellt, dass die Methode unabhängig von den systemeigenen Zeitskalen und Barrieren arbeitet, was vorteilhaft für Systeme mit Prozessen auf langen Zeitskalen ist.

CONTENTS

I Background and Fundamentals

II Daubechies Wavelets as Basis Functions for Variationally Enhanced Sampling

III Birth-Death Augmented Sampling

IV Conclusion

LIST OF ABBREVIATIONS

Adam	Adaptive Moment Estimation
CV	Collective Variable
FES	Free Energy Surface
LD	Langevin Dynamics
MD	Molecular Dynamics
MDF	Mei-Davenport-Fernando
PPPM	particle-particle particle-mesh
RMS	Root Mean Square
VES	Variationally Enhanced Sampling
WTMetaD	Well-tempered Metadynamics

LIST OF SYMBOLS

$\vec{x}$	vector of carthesian coordinates
$\boldsymbol{x}$	set of objects, $\boldsymbol{x} = \{x_1, x_2, \ldots, x_N\}$
∇	Nabla operator, if ambiguous with respect to the specified subscript $\nabla_{\vec{x}} = \sum_i \vec{e}_i \frac{\partial}{\partial x_i}$
$\dot{x}$	time derivative of x
$\langle \cdot \rangle$	(ensemble) average
$\langle f, g \rangle$	inner product
$\lfloor \cdot \rfloor$	floor function
$\bigcup$	union of sets
$\bigcap$	intersection of sets
$\mathcal{A} \subset \mathcal{B}$	$\mathcal{A}$ is subset of $\mathcal{B}$
$\overline{\mathcal{S}}$	closure of the set $\mathcal{S}$

β	inverse thermal energy $1/k_B T$
$\delta(\cdot)$	dirac delta
δt	time step
$\vec{f}$	force
F	Helmholtz free energy
γ	friction coefficient
$\mathcal{L}$	Fokker-Planck operator
L^p	Lebesgue space
m	particle mass
$\mathcal{O}(\cdot)$	order of
P	probability
$\vec{p}$	momentum
$\vec{r}$	atomic coordinates
ρ	transition density
s	collective variable
U	potential (energy)
V	bias potential
ψ	(mother) wavelet
ϕ	scaling function or father wavelet
ζ	bias factor or biasing factor
k_B	Boltzmann constant, $k_\mathrm{B} = 1.380649 \times 10^{-23}\,\mathrm{J\,K^{-1}}$
ϵ_0	vacuum permittivity, $\epsilon_0 = 8.854188 \times 10^{-12}\,\mathrm{F\,m^{-1}}$

1

INTRODUCTION

Molecular simulations are routinely used in chemistry, biology, and materials science to obtain a microscopic understanding of physical processes, offering insight into equilibrium and kinetic properties and possible pathways [1–6]. However, one major problem impeding conventional molecular dynamics (MD) [7] or Langevin dynamics (LD) [8, 9] simulations is the time scale or rare event problem. For many systems, the molecular process of interest occurs on a much longer time scale than one can simulate in practice; in other words, it is a rare event. Thus, the system stays in a metastable state during the simulation, and one does not observe transitions to other metastable states. Despite impressive developments in specialized hardware [10–12] and simulation codes [13–15] that make very efficient use of modern graphics processing units, it is unlikely that accessible time scales will increase significantly in the near future. The speedup of processing units has come to an end and high-performance computing relies on the usage of massive parallelization [16], and time is not easily parallelizable. Thus, there has been considerable interest in developing advanced methods that enhance phase space sampling and overcome this time scale problem.

A popular family of such enhanced sampling methods introduces a bias potential as a function of a few essential slow degrees of freedom, the so-called collective variables (CVs). By flattening barriers, this bias potential allows for transitions between metastable states on much shorter timescales and results in faster and more comprehensive exploration of the configurational space. Variationally enhanced sampling (VES) is a recently developed method of this family that is based on a variational principle [17, 18]. It introduces a convex functional of the bias potential that is related to the relative entropy and the Kullback-Leibler divergence. To minimize the functional, the bias potential is generally taken as a linear expansion in some basis function set. So far, the employed basis sets have been primarily sets of orthogonal functions that are delocalized in the CV space, such as plane waves, Chebyshev, or Legendre polynomials, although Gaussian basis functions have also been used [19, 20]. Additionally, bias potentials constructed from neural networks [21] or free energy models [22–25] are considered in the literature. However, there has not been an extensive study of how the choice of the basis functions affects the convergence behavior. In particular, it remains an open question if basis functions that are localized in the CV space might perform better. While Gaussian basis functions might be the type of localized basis functions that first comes to mind, they have the disadvantage of not forming orthogonal basis sets. Instead, a more appealing option might be Daubechies wavelet-based basis sets [26], as they are orthogonal and exhibit an attractive multiresolution property. Daubechies wavelets have recently been used as basis functions for other applications within molecular simulations, such as density functional theory [27, 28] or coarse-grained potentials [29].

Another possibility to obtain better sampling is offered by multiple replica methods. In their basic form, molecular simulation algorithms (such as MD or LD) generate snapshots by moving a single representation of the system through configuration space. To get more comprehensive sampling, multiple copies of the system can be used; each exploring the same underlying energy landscape but set up with different initial conditions. These copies can also be interpreted as a set of imaginary particles, the "walkers" or "replicas", exploring the phase space of the system. Using snapshots from multiple independent short simulations instead of a single long one allows for the usage of parallel hard- and software but does not overcome the time scale problem. Therefore, interactions between the replicas can be introduced to realize additional benefits. In the family of cloning methods [30–37], replicas are duplicated and killed via birth-death processes. The replicas then typically do not sample the desired distribution individually, but the correct sampling is obtained by selected replicas, collectively, or can be calculated in a post-processing step after the simulation. Enforcing an equilibrium distribution of the replicas in the relevant states via a birth-death mechanism results in sampling the desired distribution collectively via the ensemble average of the samples from all replicas. This allows obtaining accurate estimates of rare event systems even if the configurational space was only partially covered and no transitions between the states occurred from the molecular sampling scheme.

1.1 OBJECTIVES OF THE THESIS

In this thesis, two complementary enhanced sampling approaches are investigated: Improving the bias representation in the variationally enhanced sampling method, as well as a novel sampling algorithm using birth-death moves.

Building on the existing framework of variationally enhanced sampling, a thorough analysis of the bias representation in this method is performed. Crucial aspects of good basis functions are identified to subsequently propose new basis functions that might be better suited for the method. Because Daubechies wavelets are found as a promising candidate, their theoretical framework is studied before they are implemented as basis functions into the PLUMED2 software [38, 39], together with cubic B-splines and Gaussians as other new basis functions that are localized in the CV space. Good sets of parameters for the new basis functions are obtained through empirical studies. The performance of the new basis functions is assessed in numerical simulations by comparison with the performance of established basis functions. Benchmarking studies on systems of increasing complexity are performed, from the simple movement of a particle in a one-dimensional potential to the study of the association process of calcium carbonate in water.

Also, a novel sampling algorithm that augments Langevin dynamics with birth-death moves is investigated. This research is part of a project within the collaborative research center TRR146, which is funded by the Deutsche Forschungsgesellschaft. In this project, a modification of an algorithm proposed by Lu, Lu, and Nolen [30] is investigated theoretically, implemented into molecular simulation software, and tested via numerical simulations. The original algorithm provides an approximation of a stochastic birth-death process for a particle-based implementation. Alternative approximations of the birth-death term with desirable properties are proposed in the current research, together with modifications of the algorithm for better computational efficiency. Additionally, extending the applicability of the algorithm is investigated: While the original proposal and theory are only for the combination of the birth-death scheme with overdamped Langevin dynamics, using the algorithm with more general dynamics is of interest for future application to realistic chemical or biological systems. Thus, the behavior of the algorithm when augmenting general (or underdamped) Langevin dynamics with the birth-death scheme is also studied. The main objective of this work for the research project is to implement the modified algorithm and assess its behavior through numerical simulations. The theoretically derived sampling is verified by application to different model systems and the behavior of the algorithm is investigated under change of various parameters.

1.2 OUTLINE

This thesis is organized into several parts, to account for the different research presented here.

Part I presents the theoretical background and some fundamental principles used in this thesis. After the basics of two popular molecular simulation methods, molecular and Langevin dynamics, are introduced in Chapter 2, the focus of Chapter 3 is on enhanced sampling methods. There, fundamental principles of bias-based enhanced sampling methods are provided, as well as the details of the metadynamics and variationally enhanced sampling methods. Additionally, multiple replica methods and selected performance measures for molecular simulations are introduced.

The investigation of Daubechies wavelets as basis functions for the variationally enhanced sampling method is presented in Part II. Chapter 4 discusses the properties of good basis functions, first in general and then specifically for the VES method. The fundamental theory of Daubechies wavelets is presented in Chapter 5, with a focus on algorithms used for their construction. Afterward, Chapter 6 provides details about the properties of the new localized basis functions (wavelets, Gaussians, cubic B-splines) and their implementation into the VES module of PLUMED2. Short studies of the optimal parameters for each basis set are also included. In Chapter 7, the performance of the different types of basis functions is compared in benchmarking studies. Starting with simple model systems to assess the general behavior, simulations with increasing complexity are performed. Concludingly, the performance of the VES method with a wavelet-based bias potential is investigated on a realistic chemical system, the association process of calcium carbonate in water, where it is compared to existing basis functions as well as the metadynamics method. The short Chapter 8 summarizes the findings and gives some perspective on further applications and improvements of the method.

Part III provides the research on the novel birth-death augmented Langevin sampling scheme. A short motivation is given in Chapter 9. After providing fundamentals of the theory of stochastic processes, Chapter 10 introduces the theoretical framework of the algorithm as well as the different approximations of the birth-death term. Chapter 11 presents details on the implementation of the algorithm. Numerical studies of the algorithm are provided in Chapter 12. There, the behavior of the algorithm with respect to the different approximations, as well as several other parameters, is investigated. In Chapter 13, the state of the research is recapitulated and an outlook on possible subsequent steps of the project is given.

Finally, Part IV provides a summary of the findings of the thesis and presents perspectives on how both research projects might be combined into a joint enhanced sampling method.

Part I

BACKGROUND AND FUNDAMENTALS

"In the right light, study becomes insight"

— Zacharias Manuel de la Rocha [40]

2

SIMULATION METHODS FOR MOLECULAR SYSTEMS

To simulate molecular systems, different methods have been developed. In the following, a system with a fixed number of atoms N with coordinates $\vec{\boldsymbol{r}} = \{\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N\}$, momenta $\vec{\boldsymbol{p}} = \{\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_N\}$, and masses $\boldsymbol{m} = \{m_1, m_2, \ldots, m_N\}$ is considered. Usually, the goal is to obtain a series of configurations of the system ("samples") that are sampled according to the underlying physical principles. In the canonical ensemble, the probability of exploring a specific point of the phase space $P(\vec{\boldsymbol{r}}, \vec{\boldsymbol{p}})$ is given by

$$P(\vec{\boldsymbol{r}}, \vec{\boldsymbol{p}}) = \frac{\exp(-\beta U(\vec{\boldsymbol{r}}))\exp(-\beta \sum_i p_i^2/2m_i)}{\mathcal{Z}}. \tag{2.1}$$

Here, $\mathcal{Z} = \int\int \mathrm{d}\vec{\boldsymbol{r}}\,\mathrm{d}\vec{\boldsymbol{p}}\,\exp(-\beta U(\vec{\boldsymbol{r}}))\exp(-\beta \sum_i p_i^2/2m_i)$ is the partition sum, $\beta = (k_\mathrm{B}T)^{-1}$ is the inverse of the thermal energy, and $U(\vec{\boldsymbol{r}})$ denotes a potential energy function depending on the position of all particles. The potential incorporates all interactions between the particles, as well as possible interactions with the surroundings. While the full probability density of eq. (2.1) defines the dynamics of the system, often only the distribution of the atoms $P(\vec{\boldsymbol{r}})$ is of interest, which is proportional to the Boltzmann factor

$$P(\vec{\boldsymbol{r}}) \propto \exp(-\beta U(\vec{\boldsymbol{r}})). \tag{2.2}$$

Limiting the discussion to the atomic coordinates or configurations of the system, the expected value of an observable O is given by

$$\langle O \rangle = \int \mathrm{d}\vec{\boldsymbol{r}}\, O(\vec{\boldsymbol{r}}) P(\vec{\boldsymbol{r}}) \tag{2.3}$$

and can be approximated from the N_s independent samples drawn from $P(\vec{\boldsymbol{r}})$ by the ensemble average

$$\langle O \rangle \approx \frac{1}{N_s} \sum_{n=1}^{N_s} O(\vec{\boldsymbol{r}}_n). \tag{2.4}$$

In Monte-Carlo methods [41, 42] the sampling happens in a purely stochastic manner. The obtained series of configurations have no temporal meaning and do not contain information about the actual dynamics of the system. On the contrary, Molecular dynamics (MD) [7] and Langevin dynamics (LD) [8, 9] are simulation methods that generate time series by evolving the system according to equations of motion. This results in trajectories that can also be used to study time-dependent processes. The following will give an introduction to these dynamic methods. The section on molecular dynamics was inspired by Refs. 43–46, and for Langevin dynamics studying Refs. 46–48 gives further insight.

2.1 MOLECULAR DYNAMICS

In molecular dynamics simulations, the particles are evolved according to Newton's equations of motion

$$\dot{\vec{r}}_i(t) = \frac{\vec{p}_i(t)}{m_i}, \tag{2.5a}$$

$$\dot{\vec{p}}_i(t) = \vec{f}_i(t) = -\nabla_{\vec{r}_i} U(\vec{\boldsymbol{r}}). \tag{2.5b}$$

Here, the force acting on particle i at time t was introduced as $\vec{f}_i(t)$. To solve this system of ordinary differential equations, a finite-difference approach can be used. Given the state of the system at time t, including the particle positions, velocities, and possibly other information, the state at a later time $t + \delta t$ is computed. Desirable algorithms should match the real trajectory given by eqs. (2.5) as closely as possible while allowing a large time step δt.

A very popular algorithm to integrate Newton's equation was introduced by Verlet [49]. It generates positions that are accurate up to $\mathcal{O}(\delta t^4)$. The later introduced velocity Verlet variant [50] also directly calculates the velocities of the particles at each time step. The new positions and velocities when advancing the particles by a time step δt are given by

$$\vec{r}_i(t+\delta t) = \vec{r}_i(t) + \frac{\delta t}{m_i}\vec{p}_i(t) + \frac{\delta t^2}{2m_i}\vec{f}_i(t), \tag{2.6a}$$

$$\vec{p}_i(t+\delta t) = \vec{p}_i(t) + \frac{\delta t}{2}\left(\vec{f}_i(t) + \vec{f}_i(t+\delta t)\right). \tag{2.6b}$$

Implementations of the algorithm typically use an equivalent three-step procedure since eq. (2.6b) would require to store the forces both at time t and $t+\delta t$ in memory. The three steps come in the form of the symmetric propagation triplet

$$\vec{p}_i\left(t+\frac{1}{2}\delta t\right) = \vec{p}_i(t) + \frac{\delta t}{2}\vec{f}_i(t), \tag{2.7a}$$

$$\vec{r}_i(t+\delta t) = \vec{r}_i(t) + \frac{\delta t}{m_i}\vec{p}_i\left(t+\frac{1}{2}\delta t\right), \tag{2.7b}$$

$$\vec{p}_i(t+\delta t) = \vec{p}_i\left(t+\frac{1}{2}\delta t\right) + \frac{\delta t}{2}\vec{f}_i(t+\delta t), \tag{2.7c}$$

where the force evaluation happens between the second and third step.

Force evaluation

For systems with many atoms, the evaluation of the forces is usually the most time-consuming part of the algorithm. While the true physical interactions would require quantum mechanical calculations, classical molecular dynamics simulations model the approximately correct behavior via empirical force fields such as AMBER [51, 52] or CHARMM [53, 54]. Typically, the interactions are split into bonded and non-bonded interactions.

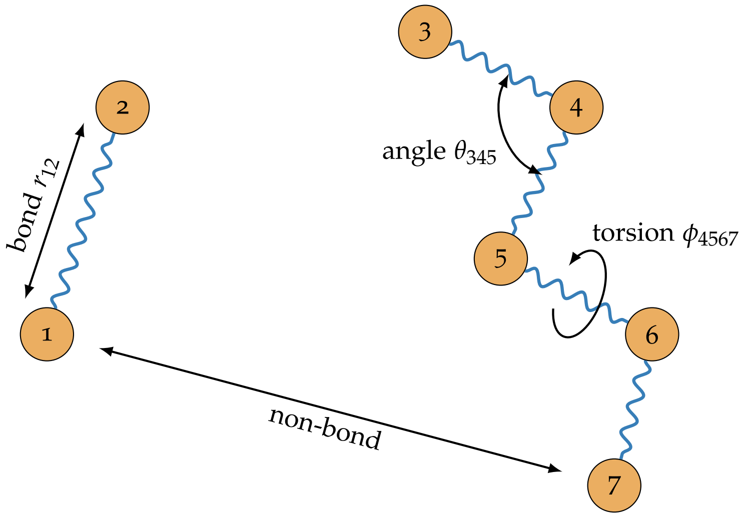


Figure 2.1: Illustration of force field terms and the respective variables used in eq. 2.8 on exemplary atoms of two molecules. Numbered circles denote atom positions and molecular bonds are shown as blue curvy lines. The bonding distance r_{ij} is shown for atoms 1 and 2, and the bond angle θ_{ijk} for atoms 3, 4, and 5. The angle between the two planes defined by atoms 4, 5, 6 and 5, 6, 7 is the torsion angle ϕ_{4567}.

The bonded interactions restrain for example bond distances and angels between atoms within a molecule. A simple model can have the following form, which is illustrated in Figure 2.1:

$$\begin{aligned} U_{\text{bonded}} &= \sum_{\text{bonds}} \frac{1}{2} k_r \left(r_{ij} - r_0\right)^2 \\ &+ \sum_{\text{angles}} \frac{1}{2} k_\theta \left(\theta_{ijk} - \theta_0\right)^2 \\ &+ \sum_{\text{torsions}} \sum_n k_{\phi,n} \left[\cos\left(n\phi_{ijkl} + \delta_n\right) + 1\right] \end{aligned} \tag{2.8}$$

The first sum is over all bonded atom pairs. Here, a harmonic restrain is used to keep the bond distances close to the equilibrium value r_0 with a spring constant k_r. The second term is a sum over all bonding angles θ_{ijk} between three connected atom positions $\vec{r}_i, \vec{r}_j, \vec{r}_k$ in a molecule, with a harmonic spring constant k_θ enforcing the equilibrium value θ_0. The third term sums over all torsions ϕ_{ijkl} of four connected atoms $ijkl$ and is an expansion in trigonometric functions with

n minima per full rotation. The parameters $k_{\phi,n}$ and δ_n are used to model the specific form of the torsion potential. Some force fields also include "improper" torsions of atoms not directly connected. The presented model assumed the same parameters for all given sets of atoms, while in practice the parameters are not the same across different atom types. For example, the constants k_r and r_0 of the bonds have to be adjusted for every pair of different atom types directly connected by bonds in a system.

Additionally to these interactions of the atoms within a molecule, non-bonded interactions between all atoms have to be taken into account. Often, the atoms already interacting via bonds and angles are excluded and the atoms connected via torsions have a modified set of parameters for the non-bonded interactions [55]. Many force fields use potentials modeling the interaction of neutral atoms and an additional term for the electrostatic interactions. Considering two particles i, j with charges q_i, q_j, they often have a general functional form similar to

$$U_{\text{nonbonded}} = U_{\text{repulsive}} - \left(\frac{C}{r_{ij}}\right)^6 + \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}}, \tag{2.9}$$

where the first term models the repulsive interactions at short distances, the second term accounts for attractive Van-der-Waals interactions, and the third term is the electrostatic Coulomb potential. A requirement for the first two terms on the right hand side of eq. (2.9) is that the repulsive interactions have to go to zero faster than the attractive interactions. The most common choice is the Lennard-Jones(12-6) potential [56]

$$U_{LJ} = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right]. \tag{2.10}$$

In the presented form, which is illustrated in Figure 2.2, the two parameters σ and ϵ have a descriptive meaning: The distance at which the potential switches from repulsive to attractive behavior is at σ, while the minimal energy of the potential is given by $-\epsilon$ at distance $\sqrt[6]{2}\sigma$. Improving on this purely phe-

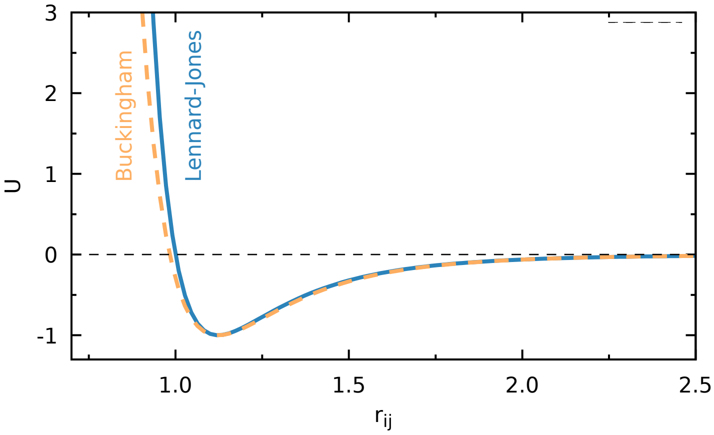


Figure 2.2: Exemplary form of the Lennard-Jones and Buckingham potentials. The parameters of the Lennard-Jones potential were set to $\sigma = \epsilon = 1$. For the Buckingham force field, the parameters were adjusted to match the form of the Lennard-Jones potential at large distances.

nomenological approach at modeling, Buckingham proposed an exponential form for the repulsive interaction based on the theory of overlapping electronic densities [57]:

$$U_{\text{Buckingham}} = A \exp(-Br_{ij}) - \frac{C}{r_{ij}^6}. \tag{2.11}$$

The three parameters A, B, and C offer more possibilities of adjusting the potential to experimental data. A choice that approximates the Lennard-Jones potential for large distances can be seen in Figure 2.2.

The non-bonded interactions need to be calculated for every pair of particles. The number of force computations for these interactions is therefore proportional to the number of distinct pairs, $\mathcal{O}(N^2)$, and becomes increasingly large for systems with many atoms. Because the strength of non-bonded interactions is going to zero for long distances, usually only particles within some distance to each other are taken into account for the force calculations. The exact value of this spherical cutoff r_{cut} depends on the force field being used and the system under investigation. Usually it is on the order of a few σ for Lennard-Jones-like force fields.

Additional speedup can be achieved by using neighbor lists. To find the particles that have to be considered for non-bonded interactions, the distances between all atoms would have to be calculated at every time step, and then decided if they are within the cutoff distance. Instead, Verlet proposed the use of neighbor lists that store all particles within a certain skin distance r_{skin} slightly larger than the cutoff. Only the distances of particles inside the lists are calculated at every step. The list of particles to consider is updated every couple of time steps. This update time for the lists has to be chosen such that particles do not travel all the way through the skin region when coming from the outside between subsequent list updates.

Periodic boundary conditions

The size of the system to be simulated limits the time scales that can be observed within a given computational time. While modern customized hardware is able to simulate up to hundred microsecond of trajectory per day for systems with as many as 100 million atoms [12], broadly available hardware is typically only used for much smaller systems. On the typically available simulation clusters, simulations of small isolated systems, such as liquid droplets or microcrystals, are possible without further modifications of the scheme. But when simulating extended systems, such as solvated molecules in bulk liquid, the number of atoms required to capture the correct behavior quickly grow beyond what is feasible to simulate.

Therefore, the simulation box is often modified to have periodic boundary conditions: The box is replicated in space to form an infinite lattice. When an atom moves in the simulation box, its periodic images move in exactly the same ways. Thus, when an atom crosses the boundary and leaves the box, a periodic image at the opposite side of the box enters. In practice, only the atoms in the central box need to be stored and the atoms crossing the box reenter at the opposite face of the box with the same momentum. This allows to significantly reduce the number of atoms to be simulated, although care has to be taken that the box is chosen large enough to avoid finite-size effects. It is therefore crucial to study the effect of the box size to obtain the desired macroscopic physics from the small periodic system.

When using a periodic box, the forces between the atoms are calculated based on the "minimal image convention" [58], which means that for each atom only the force of the periodic image with the smallest distance is considered. The periodic boundary conditions are now combined with neighbor lists and cutoff distances for the non-bonded interactions to obtain an efficient simulation protocol. To satisfy the minimal image convention intrinsically, the cutoff distances must not be larger than half the size of the periodic box. While this is typically satisfied for the short-ranged repulsive and van-der-Waals interactions, the electrostatic interactions are of much longer range and can quickly extend outside the simulation box. While extending the box would solve the problem, this comes at high additional costs for the simulation. Therefore, methods that approximate the long-range electrostatic interactions are common, for example the Ewald sum [59]. The Ewald sum does not use only the minimal image but attempts to calculate the electrostatic interactions of all the periodic images up to a given layer of boxes around the center by using a lattice. Even more efficient are modern extensions that do the calculations via fast Fourier transforms, such as the particle-particle particle-mesh (PPPM) method [60, 61].

Sampling from other thermodynamic ensembles

When simulating the motion of atoms through Newton's equation and classical force fields, the total energy of the system

$$E_{\text{tot}} = E_{\text{kin}} + U(\vec{\boldsymbol{r}}) = \sum_{i=1}^{N} \frac{p_i^2}{2m_i} + U(\vec{\boldsymbol{r}}) \tag{2.12}$$

is conserved.

Using a fixed number of atoms and a simulation box of fixed volume, this results in sampling from the microcanonical (NVE) ensemble where the temperature and pressure are not fixed. This does not resemble realistic experimental conditions for most applications. To perform simulations in the canonical (NVT) or isothermal-isobaric (NPT) ensemble, the MD algorithm has to be extended to keep the temperature, and in the second case also the pressure, constant on average.

To keep the temperature constant, one simple idea is to rescale the velocities of the atoms in the system based on the equipartition theorem

$$k_\mathrm{B}T = \frac{\langle 2E_{kin}\rangle}{N_f}, \tag{2.13}$$

for a system with N_f degrees of freedom.

Other thermostatting approaches couple the system by random interactions with imaginary heat bath particles, that is they reselect at certain times the atomic velocities of a random particle from the Maxwell-Boltzmann distribution

$$P(\vec{p}_i) = \frac{\exp\left(\frac{-\beta(\vec{p}_i)^2}{2m}\right)}{\int \mathrm{d}\vec{p}\ \exp\left(\frac{-\beta(\vec{p})^2}{2m}\right)}. \tag{2.14}$$

While this Andersen thermostat [62] does indeed sample the NVT ensemble, the random momentum changes do not result in correct momentum transport in the system and dynamic properties such as the diffusion or viscosity cannot be obtained correctly. Therefore, deterministic methods that modify the equations of motion have become more common than statistical approaches. Popular variants are the ones by Berendsen et al. [63], Nosé and Hoover [64, 65], or Bussi et al. [66]. The thermostat developed by Nosé and Hoover [64, 65] adds an additional "thermal reservoir" variable or "friction coefficient" η to rescale the momenta [58]:

$$\dot{\vec{r}}_i(t) = \frac{\vec{p}_i(t)}{m_i} \tag{2.15a}$$

$$\dot{\vec{p}}_i(t) = \vec{f}_i(t) - \eta\vec{p}_i \tag{2.15b}$$

$$\dot{\eta}(t) = v_T^2\left(\frac{\mathcal{T}}{T} - 1\right) \tag{2.15c}$$

The friction coefficient varies during the simulation depending on the ratio between the instantaneous mechanical temperature $\mathcal{T}$, calculated from the kinetic energy of the atoms, and the desired temperature T. For $\mathcal{T} > T$ the value of η increases, where the parameter v_T can be seen as relaxation rate for the thermal fluctuations. If ζ is positive, the momenta of all particles will be scaled down at every time step and the system becomes gradually "colder". For a too cold system, that is $\mathcal{T} < T$, the opposite happens. Under certain circumstances, such as small or very stiff systems, the proposed Nosé-Hoover dynamics can be slow at sampling the full configurational space and non-ergodic behavior might be observed. This can be rectified by thermostat chains that also drive the fluctuations of the thermostat variables [67].

For simulations that keep the pressure constant, similar approaches in the form of barostats are used. In the NPT-ensemble, a common set of equations of motions based on the Nose-Hoover dynamics given in eq. (2.15) were proposed by Martyna et al. [68].

2.2 LANGEVIN DYNAMICS

Langevin dynamics was originally developed to describe Brownian motion, that is, free diffusion of particles in a fluid. When investigating the behavior of the solute particles, interesting processes can often be observed on much longer time scales than the ones related to the solvent. The idea is therefore to separate the scales and approximate the fast degrees of freedom from the solvent only as stochastic contribution to the dynamics of the solute particles.

The Langevin equations to be considered here are the stochastic differential equations

$$\mathrm{d}\vec{r}_i(t) = \frac{\vec{p}_i(t)}{m_i}\,\mathrm{d}t, \tag{2.16a}$$

$$\mathrm{d}\vec{p}_i(t) = \vec{f}_i(t)\,\mathrm{d}t - \gamma\vec{p}_i(t)\,\mathrm{d}t + \xi_i\,\mathrm{d}\overrightarrow{W}_i, \tag{2.16b}$$

where the second and third terms of eq. (2.16b) act as linear dissipation with friction coefficient γ and stochastic force with strength ξ_i, respectively. The noise of the stochastic term is uncorrelated between all particles i and directions α and interpreted as a Wiener process in the Itô convention. Assuming a time step of length δt, each component $G_{i,\alpha}$ of $\mathrm{d}\overrightarrow{W}_i = \overrightarrow{W}_i(t+\delta t) - \overrightarrow{W}_i(t) = \sqrt{\delta t}\,\vec{G}_i$

is an independent Gaussian random variable with mean zero, $\langle G_{i,\alpha} \rangle = 0$, and variance one, $\langle G_{i,\alpha}(t) G_{j,\beta}(t') \rangle = \delta_{i,j}\delta_{\alpha,\beta}\delta(t-t')$. Of note is, that the stochastic force is completely independent of the particle positions or momenta as well as time.

In the case of pure Brownian motion, the deterministic force $\vec{f}_i(t)$ is zero. Then Einstein's expression for the diffusion coefficient,

$$D = \frac{1}{\gamma \beta m}, \tag{2.17}$$

holds. The strength of the random forces and the friction coefficient are related by the fluctuation-dissipation theorem

$$\xi_i = \sqrt{\frac{2\gamma m_i}{\beta}}. \tag{2.18}$$

The Langevin equations can be interpreted as the movement of Newtonian particles in contact with a thermal reservoir at temperature T [47]. Although the temperature does not directly appear in the equations of motion (2.16), the thermal energy of the system connects the friction and noise term in the fluctuation dissipation theorem.

An often studied special case is the overdamped limit of the Langevin dynamics. This represents a system where the inertia of the particles is much smaller than the friction: $\gamma/m_i \to \infty$. In this limit, the left side of eq. (2.16b) vanishes and the equations of motion reduce to

$$\mathrm{d}\vec{r}_i(t) = D\beta \vec{f}_i(t)\,\mathrm{d}t + \sqrt{2D}\,\mathrm{d}\vec{W}, \tag{2.19}$$

where the diffusion coefficient is used for shorter notation.

Integrators

To obtain particle trajectories, the stochastic differential equations of eq. (2.16) or (2.19) need to be integrated. A multitude of algorithms has been developed to this end. For general Langevin dynamics, early attempts did not result in a stable molecular dynamics method in the overdamped limit. Later suggestions solved this issue by using a Verlet-like scheme. Modern variants typically use operator-splitting to derive algorithms similar to the velocity Verlet method. Bussi and Parinello [48] introduced an integrator with the following steps:

$$\vec{p}_i(t^+) = c_1\vec{p}_i + c_2\vec{G}_i(t) \tag{2.20a}$$

$$\vec{p}_i\left(t+\frac{1}{2}\delta t\right) = \vec{p}_i(t^+) + \frac{\delta t^2}{2}\vec{f}_i(t) \tag{2.20b}$$

$$\vec{r}_i(t+\delta t) = \vec{r}_i(t) + \frac{\delta t}{m_i}\vec{p}_i\left(t+\frac{1}{2}\delta t\right) \tag{2.20c}$$

$$\vec{p}_i(t^- + \delta t) = \vec{p}_i\left(t+\frac{1}{2}\delta t\right) + \frac{\delta t}{2}\vec{f}_i(t+\delta t) \tag{2.20d}$$

$$\vec{p}_i(t+\delta t) = c_1\vec{p}_i(t^- + \delta t) + c_2\vec{G}'_i(t) \tag{2.20e}$$

The algorithm consists of a velocity Verlet triplet in steps (b) to (d) of the algorithm enclosed by two "thermostat" steps accounting for the stochastic force. The two coefficients

$$c_1 = \exp(-\gamma\frac{\delta t}{2}), \tag{2.21a}$$

$$c_2 = \sqrt{(1-c_1^2)\frac{m}{\beta}}, \tag{2.21b}$$

were introduced for a shorter notation and $\vec{G}_i, \vec{G}'_i$ denote vectors of random and independent numbers drawn from a Gaussian probability distribution with mean 0 and standard deviation 1.

While this algorithm is also valid in the overdamped limit, the simpler Euler-Maruyama method [69] can be used to integrate eq. (2.19) directly. The algorithm consists of a one-step update rule for the particle positions [70]:

$$\vec{r}_i(t+\delta t) = \vec{r}_i(t) + D\beta\,\delta t\vec{f}_i(t) + \sqrt{2D}\,\delta t\,\vec{G}_i \tag{2.22}$$

3

ENHANCED SAMPLING

To calculate meaningful estimates from atomistic simulations via eq. (2.4), the obtained samples need to cover all relevant parts of the configuration space sufficiently. Formally, this is expressed as ergodicity: given infinite time all accessible configurations of the system are explored. Additionally, the dynamics of the system have to be sensitive to the starting conditions, such that different initial states lead to uncorrelated trajectories. This also results in a behavior of the system at late times that is completely uncorrelated to the initial state. In other words, the system has to display an approach towards equilibrium. In ergodic theory, this condition is named “mixing” [71].

Ergodicity is hard to prove in systems with many degrees of freedom and it will often be assumed blindly that ergodic trajectories were obtained. Indeed, typical systems that are studied with molecular simulations are ergodic, although the simulation time required to visit all configurations might be quite long. Especially for systems where the energy landscape possesses high barriers between metastable states (“rare event” systems), obtaining ergodic simulation trajectories is usually very difficult in practice [72]. The system might become “trapped” in some state of the system and be unable to transition to other parts of the configurational space within time scales feasible for simulation. Methods that improve the sampling in rare event systems, such that all releveant parts are visited, are also called “enhanced sampling” methods.

Collective variables

Many methods do not enhance sampling globally, but rather focus on enhancing fluctuations along a few collective variables (CVs), which are sometimes called reaction coordinates. These CVs usually correspond to critical slow degrees of freedom [73]. Ideally, they should describe all characteristics of the physical process of interest. Additionally, they should be able to clearly distinguish between all the relevant states, which means that their values at the different states do not coincide. Most methods also put an upper limit on the number of CVs, usually not more than three to five, either because of theoretical restrictions or simply for computational efficiency.

Clearly, identifying a set of CVs is not an easy task. Missing a crucial degree of freedom yields suboptimal sampling and might lead to misinterpretation of results due to hidden barriers and hysteresis effects. On the other hand, the limited number of allowed CVs does not allow to include more and more degrees of freedom if a sparse set cannot be found. In the literature, the committor function is sometimes proposed as the optimal reaction coordinate that is able to describe the process of interest by a single CV. Its definition in a system with two metastables states is given by the probability that a given configuration will evolve to the final product state before it reaches the initial reactant state [74]. This also allows for a simple definition of the transition state as the points where the commitor function is 0.5, that is, the configurations where it is equally likely that the system will reach the reactant or the product state first.

While the committor function indeed satisfies the requirements for a good CV, it is hard to determine in practice. Instead, CVs are typically selected by hand using physical and chemical intuition [75–77]. This is often an iterative process involving trial-and-error: Starting with a set of CVs obtained from, for example, experimental or theoretical information, a short enhanced sampling simulation is performed to assess the quality of the CVs and get further information on the system. With the additional information, the CVs can then be refined to describe the process of interest better and another short run of an enhanced sampling simulation is performed. This process is repeated until a satisfactory set has been determined.

Alternatively, methods such as principal component analysis [78], time-lagged component analysis [79], path collective variables [80], sketch map [81], spectral gap optimization of order parameters [82], reweighted autoencoded variational Bayes [83], Gaussian mixture variational encoders [84], or encoder map [85] have been used to facilitate the identification of slow degrees of freedom with algorithms, where especially algorithms based on machine learning [74, 83–87] are becoming popular recently.

In the following sections, it is assumed that an appropriate set of CVs, $\boldsymbol{s} = \{s_1(\vec{\boldsymbol{r}}), s_2(\vec{\boldsymbol{r}}), \ldots, s_N(\vec{\boldsymbol{r}})\}$, has already been identified. The corresponding equilibrium probability distribution is then defined as the marginal density of the CVs

$$P(\boldsymbol{s}) = \int \mathrm{d}\vec{\boldsymbol{r}}\, \delta(\boldsymbol{s} - \boldsymbol{s}(\vec{\boldsymbol{r}}))\, P(\vec{\boldsymbol{r}}), \tag{3.1}$$

and the standard definition of the free energy is given by

$$F(\boldsymbol{s}) = -\beta^{-1} \log P(\boldsymbol{s}) + C. \tag{3.2}$$

This low-dimensional description of the free energy as a function of the CVs is also called the free energy surface (FES). The surface property can be made clear by rewriting the marginal density in eq. (3.1) as an integral over surface elements [88]. Often only relative differences in free energies are of interest and the additive constant C in eq. (3.2) is omitted.

Overview on enhanced sampling methods

Methods that enhance the sampling along the identified CVs include, but are not limited to, umbrella sampling [89, 90], free energy perturbation [91], thermodynamic integration [92], local elevation [93], adaptive biasing force [94, 95], energy landscape paving [96], conformational flooding [97, 98], Gaussian-mixture umbrella sampling [99], basis function sampling [100], Green's function sampling [101], on-the-fly probability enhanced sampling [102, 103], adaptive topography of landscape for accelerated sampling [104], metadynamics [73, 105, 106], and variationally enhanced sampling [17, 18].

While the present research focuses on CV-based methods, some alternatives that enhance the sampling globally rather than focussing on few relevant degrees of freedom are mentioned here for the sake of completeness. A non-exhaustive list of these methods consists of multicanonical ensemble simulations [107, 108], the Wang-Landau algorithm [109], hyperdynamics [110, 111], accelerated molecular dynamics [112], temperature-accelerated dynamics [113], and simulated tempering [114]. Additionally, some methods rely on simulating multiple interacting copies of the system. These will be considered separately in Section 3.3.

A typical approach to enhance sampling is adding a bias in form of an external potential to the system. This bias can be either constant during the simulation or updated adaptively in intervals from the already accumulated data. It is used to drive the system towards the parts of interest of the configuration space. The basic idea for CV-based methods can be traced back to the umbrella sampling method [89, 90], where the addition of a bias potential $V(\boldsymbol{s})$ leads to a biased CV distribution given by

$$P_V(\boldsymbol{s}) \propto \mathrm{e}^{-\beta[F(\boldsymbol{s})+V(\boldsymbol{s})]}. \tag{3.3}$$

A concept similar to enhancing sampling with a bias potential is the so-called importance sampling proposed originally for Monte Carlo simulations [115]. There, instead of sampling a difficult probability distribution directly, the idea is to draw samples from an alternative distrubtion that focuses sampling on the important parts of the problem. Although it is typically expressed quite differently, the overlap to enhanced sampling methods is significant. In fact, if certain conditions are satisfied an importance sampling method can be also seen as an enhanced sampling method and vice versa [116].

In enhanced sampling simulations the biasing introduces an additional biasing force. Because the bias potential is described in the space of CVs, the actual forces on the atoms have to be calculated by the chain rule

$$f_V(r_i) = -\sum_j \frac{\partial s_j}{\partial r_i} \frac{\partial V(\boldsymbol{s})}{\partial s_j}. \tag{3.4}$$

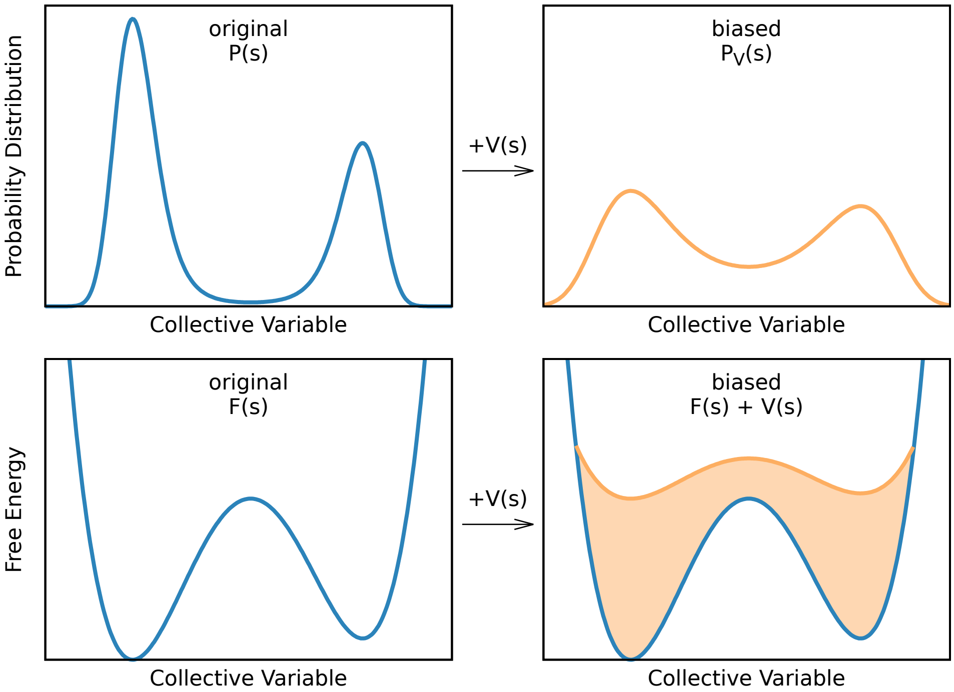


Figure 3.1: Effect of adding a bias potential $V(s)$ to a prototypical rare event system consisting of two states separated by a high barrier.
Left column: The original probability distribibution (*top*) and associated FES (*bottom*).
Right column: Probability distribution (*top*) and FES (*bottom*) with added bias potential that lowers the barrier and increases sampling in previously low-probability regions. The colored area in the right panel denotes the added bias potential while the orange line is the combination of the original FES and the bias potential that is now being sampled. In this example, the bias potential was taken as $V_G(s) = -\frac{3}{4}F(s)$.

In Figure 3.1 the effect of the biasing potential can be seen on an exemplary system consisting of two states separated by a high barrier. The original distribution in the top left panel has only neglectable probabilities in the area between the two states. Due to the missing overlap between the states, transitions between the states are unlikely and can be observed only on long time scales. On the other hand, by biasing the system some overlap between the distributions is introduced, which can be seen in the top right panel. The exact form of the bias potential depends highly on the system of interest and the method used: Taking again the example of a rare event system, desired is usually a distribution that facilitates sampling by reducing barriers or even flattening them completely. In the example shown in Figure 3.1, the barriers of the free energy surface that can be seen in the bottom left panel were only reduced but not completely flattened after the bias potential was added in the bottom right panel. This results in shorter transition times between the two states, which makes observation of the associated processes more likely within typical simulation times. The main differences between CV-based enhanced sampling methods are in how they construct the bias potential and which kind of biased sampling they lead to.

Two methods are introduced in more detail in the following sections: metadynamics and variationally enhanced sampling.

3.1 METADYNAMICS

Metadynamics is a highly popular enhanced sampling method originally introduced by Laio and Parrinello in 2002 [105]. It has been employed in a wide range of different applications, some examples include protein folding [117–119], ion association [105, 120], phase transitions [121–123], ion conduction [124–126], chemical reactions [127, 128], and diffusion [129–131].

Several extensions or modifications to the original algorithm have been proposed over time. Among them are well-tempered metadynamics (WT-MetaD) [106], multiple-walker metadynamics [132], metadynamics with adaptive Gaussians [133], infrequent metadynamics [134], combining metadynamics and hyperdynamics [135], parallel bias metadynamics [136], or combining metadynamics and free energy perturbation [137]

Notable is also the number of theoretical studies available of which two shall be explictily mentioned here: Laio et al. [138] derived how the accuracy of a metadynamics simulation depends on the method parameters, while Dama et al. [139] showed that the bias potential in the well-tempered variant converges exactly to a theoretical value, given infinite simulation time.

For further details, the reader is referred to one of the many review papers, for example by Valsson et al. [77], Bussi et al. [73], or Barducci et al. [140].

Following the ideas of these papers, the next sections will introduce the basic method as well as the well-tempered variant.

3.1.1 *Basic algorithm*

The original metadynamics algorithm is based on the idea of gradually filling the potential energy wells with a bias potential, such that the landscape becomes flat. Then the corresponding probability distribution $P(\boldsymbol{s})$ is approaching a uniform distribution and free diffusion in the chosen space of CVs is achieved. By looking at eq. (3.3), the desired bias potential for uniform sampling can be identified as exactly the negative of the free energy

$$V(\boldsymbol{s}) = -F(\boldsymbol{s}). \tag{3.5}$$

The metadynamics method iteratively builds a bias potential that approximates this ideal case by the successive addition of small repulsive Gaussian kernels $K(\boldsymbol{s}, \boldsymbol{s}')$ centered at the current CV value $\boldsymbol{s}' = \boldsymbol{s}(t')$

$$K(\boldsymbol{s}, \boldsymbol{s}') = h \exp\left(-\frac{(\boldsymbol{s} - \boldsymbol{s}')^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{s} - \boldsymbol{s}')}{2} \right). \tag{3.6}$$

Other choices than this multivariate Gaussian kernel are also possible. In eq. (3.6), the parameter h defines the height of the Gaussian while the covariance matrix $\boldsymbol{\Sigma}$, or rather its inverse $\boldsymbol{\Sigma}^{-1}$, corresponds to the spread of the Gaussian hill. For the covariance most of the time a diagonal matrix $\Sigma_{i,j} = \delta_{i,j}\sigma_i^2$ is used. Without loss of generality, the following will assume this case and refer to the standard deviation σ_i also as the "width" of the Gaussian for CV s_i. While the covariance matrix will be kept fixed for all applications presented here, the usage of adaptive Gaussians, where the optimal width is estimated on the fly, has also been proposed [133].

In the standard algorithm of metadynamics, Gaussians will be added to the bias potential at regular intervals, for example with frequency τ_G. The bias potential acting on the system at time t is now given by

$$V(\boldsymbol{s}) = \sum_{\substack{t'=\tau_G, 2\tau_G, \cdots \\ t' \leq t}} h \exp\left(- \sum_{i=1}^{N} \frac{(s_i - s_i(t'))^2}{2\sigma_i^2} \right), \tag{3.7}$$

where the first sum runs over all times where Gaussians were deposited and the inner sum over the CVs.

The Gaussians are added to the parts of the CV space that were already visited during the simulation and the resulting bias potential pushes the system toward the yet unexplored parts. Assuming a typical rare event system with several minima separated by high barriers, the algorithm will start filling the well of the minimum in which the simulation was initialized until the barrier is reduced enough to transit to the next minimum. This process is repeated until all the minima are filled and uniform sampling is achieved. The trajectory and time evolution of the bias potential for an exemplary metadynamics run can be seen in Figure 3.2.

In this example, a particle is moved in a three-minima potential, shown as thick black line in the bottom panel. The dynamics is generated by a Langevin equation with friction constant $\gamma = 10$, time step $\delta t = 0.01$, and temperature $T = 0.5/k_{\mathrm{B}}$, where the particle was initialized in the central minimum. As the barriers to the other minima are significantly higher than the thermal energy, transitions to other states are unlikely and would only be observed on long time scales without enhanced sampling. Adding Gaussians of width $\sigma = 0.25$ and height $h = 0.1$ to the metadynamics bias potential every 300 Langevin time steps results in the aforementioned filling of the wells. After 1250 deposited Gaussians all three minima are filled and the combined $F(\boldsymbol{s}) + V(\boldsymbol{s})$ is becoming flat. At this point, eq. (3.5) holds and an approximation to the free energy of the system can directly be determined from the bias potential up to a time-dependent constant:

$$F(\boldsymbol{s}) = -V(\boldsymbol{s}) + C(t) \tag{3.8}$$

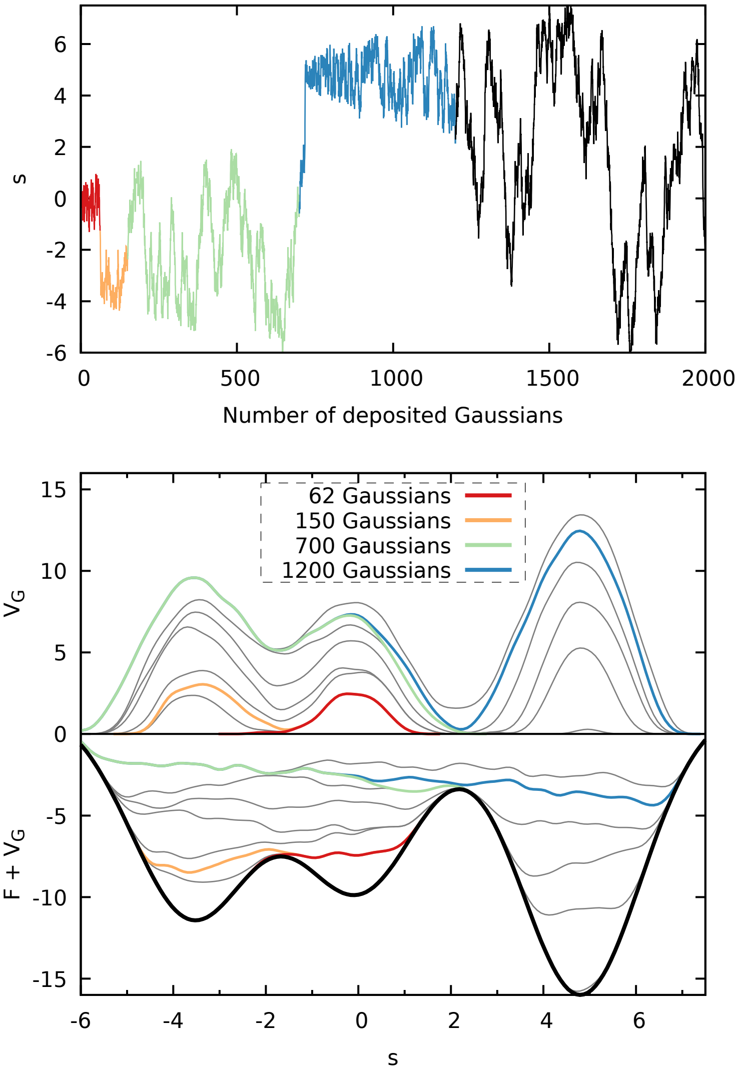


Figure 3.2: Illustration of the metadynamics method for a simple Langevin dynamics simulation.
Top panel: The trajectory of the Langevin particle moved in an external one-dimensional potential $U(s) = F(s)$ shown as a thick black line in the lower panel. The dynamics is biased with a metadynamics potential V_G created on the fly with parameters $\sigma = 0.25$, $h = 0.1$, and $\tau_G = 300\delta t$.
Middle panel: Time evolution of the bias potential.
Bottom panel: Time evolution of the sum of the external and the metadynamics potential. The gray lines in the lower two panels show snapshots after each 120 added Gaussians, the colored lines denote when the bias potential has filled a specific minimum. Red line: The first minimum around $s = 0$ is filled and the particle can escape to the second minimum. Orange line: The second minimum is filled and the particle diffuses freely in the first two minima. Green line: The bias potential has filled the combined well of the first two minima. Blue line: The last minimum is filled and the particle diffuses freely in the whole potential range. Graphic similar to Fig. 1 of Ref. [73].

Because Gaussians are still added to the bias potential, this can only be accurate up to the order of the height of the deposited Gaussians. Therefore sometimes not just the instantaneous value of the bias, but a time average over the last part of the simulation is taken [141]. Additionally, the stopping point of the simulation might be crucial: stopping too late might result in exploring undesired high energy regions of the system, while stopping before the free energy has been completely flattened by the bias potential makes analyzing the results prone to errors. To prevent sampling of undesired regions of the CV space, additional bias potentials in form of walls can be used. These static restraints push the system back when trying to cross the wall.

In summary, metadynamics provides a simple to implement method to drive the system towards unexplored parts also over high barriers. It does not require a-priori knowledge of the FES or the different metastable states and inherently explores the low energy regions of the system first. The choice of parameters depends on the system under investigation, though. An exemplary sequence of the algorithm steps in pseudocode can be found in Appendix A.1.

A way to ameliorate the accuracy of the free energy estimate is to decrease the height of the deposited Gaussians over time. Well-tempered metadynamics, the most popular variant using that principle, will be introduced in the next section.

3.1.2 *Well-tempered metadynamics*

For the basic metadynamics algorithm, the bias potential is never becoming stationary but continues to increase at a finite rate. This non-equilibrium behavior means that in practice $F(\boldsymbol{s}) - V(\boldsymbol{s}) = C(t)$ is not actually true because the constant $C(t)$ will fluctuate also in CV space.

Therefore, the well-tempered variant of metadynamics was introduced by Barducci et al. in 2008 [106]. The idea is to reduce the height of the added Gaussians by a factor that depends on the already existing bias at the current CV value

$$h = h_0 \exp\left(-\frac{1}{\zeta - 1}\beta V(\boldsymbol{s})\right), \tag{3.9}$$

where h_0 is the initial height of the Gaussians and $\zeta \geq 1$ is the so-called bias or biasing factor. The factor $(\zeta - 1)\beta^{-1}$ is sometimes also written as $k_B\Delta T$, where the parameter ΔT has the dimensions of a temperature. Clearly, for $\zeta \to \infty$ the non-tempered variant of metadynamics is recovered, while finite values of ζ result in a decreasing rate at which the bias potential grows over time. For $\zeta \to 1$ the height of the Gaussians tends to zero and an unbiased simulation is performed.

With Gaussians of decreasing heights added at the current CV value every τ_G time steps, the bias potential of the well-tempered metadynamics method at time t is given by

$$V(\boldsymbol{s}) = \sum_{\substack{t'=\tau_G,2\tau_G,\cdots \\ t' \leq t}} h_0 \exp\left(-\frac{1}{\zeta - 1}\beta V(\mathbf{s}, t')\right) \exp\left(-\sum_{i=1}^{N} \frac{(s_i - s_i(t'))^2}{2\sigma_i^2}\right), \tag{3.10}$$

where the first sum runs over all times t' at which Gaussians have been previously deposited.

In practice, calculating eq. (3.10) becomes increasingly time-consuming with more Gaussians being added. Evaluating thousands of possibly multiple dimensional kernel functions at each time step leads to significant overhead and might slow down the simulation a lot. Alternatively, one can express the time evolution of the bias by an update rule

$$V_n(\boldsymbol{s}) = V_{n-1} + K(\boldsymbol{s}, \boldsymbol{s}') \exp\left(-\frac{1}{\zeta - 1}\beta V_{n-1}(\boldsymbol{s}')\right), \tag{3.11}$$

where $V_0(\boldsymbol{s}) = 0$, and $\boldsymbol{s}'$ is again the CV value at the time of adding another Gaussian.

The bias does not change between Gaussian depositions. Therefore, to avoid unnecessary calculations, the bias can be stored on a grid. This is typically done for all but the shortest simulations. The evaluation of the bias becomes a simple grid lookup, and only a single kernel evaluation has to be done when adding another gaussian via the update rule of eq. 3.11. An exemplary implementation of the WTMetaD algorithm in pseudocode can be found in Appendix A.1.

The remarkable property of the WTMetaD variant is that the bias converges asymptotically. In Ref. 139 it was shown that the bias potential of eq. (3.10) converges exactly to

$$V(\boldsymbol{s}) = -\left(1 - \frac{1}{\zeta}\right) F(\boldsymbol{s}) + c(t) \tag{3.12}$$

in the limit of long simulation times and short times between added Gaussians. Thus, the bias is simply a tuneable fraction of the unbiased free energy. This holds regardless of the kernel being used [142] and the now time-independent constant is given by

$$c(t) = \beta^{-1} \log\left(\frac{\int \mathrm{d}\boldsymbol{s}\, \mathrm{e}^{-\beta F(\boldsymbol{s})}}{\int \mathrm{d}\boldsymbol{s}\, \mathrm{e}^{-\beta F(\boldsymbol{s}) + V(\boldsymbol{s})}}\right). \tag{3.13}$$

Inserting eq. (3.12) into the probability distribution of eq. (3.3) gives also a more intuitive meaning to the bias factor ζ and the associated parameter ΔT:

$$P_V(\boldsymbol{s}) \propto \exp\left[-\frac{\beta F(\boldsymbol{s})}{\zeta}\right] = \exp\left[-\frac{F(\boldsymbol{s})}{k_\mathrm{B}(T + \Delta T)}\right] \tag{3.14}$$

After convergence of the bias, the temperature of the sampling along the biased CVs is increased by ΔT. Alternatively, one could say that the simulation samples the unbiased FES scaled down by a factor of $1/\zeta$. The barriers are therefore not completely flattened but reduced by a factor of $1/\zeta$.

To obtain the free energy from a metadynamics simulation, one can estimate it directly from the bias up to some constant by inverting eq. (3.12)

$$F(\boldsymbol{s}) = -\left(\frac{\zeta}{\zeta - 1}\right) V(\boldsymbol{s}, t). \tag{3.15}$$

In Ref. 142 it was shown that this is only the zeroth-order estimator and it can be made more accurate by adding a first-order term. However, the higher-order terms vanish for sufficiently converged bias potentials, that is long enough simulation times, and are typically neglected.

Another method to estimate the free energy or any other observable from the biased simulations is to reweight the obtained trajectory data [143]. The unbiased probability distribution can be calculated from the biased distribution at time t by

$$P(\vec{\boldsymbol{r}}) = P_V(\vec{\boldsymbol{r}})\,\mathrm{e}^{\beta[V(\boldsymbol{s}(\vec{\boldsymbol{r}}),t)-c(t)]}, \tag{3.16}$$

which means that the average of an observable in the unbiased ensemble can be calculated from the trajectory in the biased ensemble via

$$\langle O(\vec{\boldsymbol{r}})\rangle = \left\langle O(\vec{\boldsymbol{r}})\,\mathrm{e}^{\beta[V(\boldsymbol{s}(\vec{\boldsymbol{r}}),t)-c(t)]}\right\rangle_V . \tag{3.17}$$

This relation requires the value of $c(t)$. It can be calculated during the simulation by

$$c(t) = \beta^{-1}\log\left(\frac{\int \mathrm{d}\boldsymbol{s}\exp\left[\frac{\zeta}{\zeta-1}\beta V(\boldsymbol{s})\right]}{\int \mathrm{d}\boldsymbol{s}\exp\left[\frac{1}{\zeta-1}\beta V(\boldsymbol{s})\right]}\right), \tag{3.18}$$

which was obtained by inserting eq. (3.15) into eq. (3.13). Reweighting makes it also possible to obtain the FES for other variables $\boldsymbol{s}'$ than the biased CVs $\boldsymbol{s}$, by calculating

$$F(\boldsymbol{s}') = -\beta^{-1}\log\left\langle\delta\big(\boldsymbol{s}'-\boldsymbol{s}'(\vec{\boldsymbol{r}})\big)\,\mathrm{e}^{\beta V(\boldsymbol{s}(\vec{\boldsymbol{r}}))}\right\rangle_V . \tag{3.19}$$

In practice, eq. (3.19) is often obtained by taking a weighted histogram of the trajectory data. Besides discrete histogramming of the data into bins, the usage of kernel density methods is found frequently. There, the delta function in eq. (3.19) is replaced by a kernel function, typically a Gaussian, and the ensemble average is then obtained by summing over all kernels. This can help to get a smoother estimate of the FES, although the choice of the kernel parameters is crucial to avoid oversmoothing of the relevant features or obtaining a sum of non-overlapping kernel functions. Other possibilities for obtaining smooth estimates from the discrete data include technices like the averaged shifted histogram [144–146].

3.2 VARIATIONALLY ENHANCED SAMPLING

Variationally enhanced sampling (VES) is another CV-based enhanced sampling method, which was first introduced in 2014 by Valsson and Parrinello [17]. While the working principles behind the method are quite different, many of the ideas that were described in the previous section on metadynamics apply also to it. The VES method can be applied in a similar fashion to the same kind of problems by biasing the sampling along a set of CVs. Exemplary applications described in the literature include nucleation processes [22, 147, 148], protein folding [149], kinetic rate calculations [150], phase transitions [24, 25], metal-organic frameworks [19], Monte Carlo renormalization group [151–153], crystallization processes [21, 154], characterizing high dimensional free energy landscapes [155], photoisomerization [156], catalytic reaction kinetics [157], or calculating the chemical potential in fluids [158].

In the following the fundamental principle of the method will be introduced, the reader is referred to the literature [17, 18, 77, 159] for further details.

3.2.1 *Variational principle*

The VES method is based on a variational principle, where the optimization problem is to find the minimum of a functional $\Omega[V]$ depending on the bias potential. This functional is defined as

$$\Omega[V] = \frac{1}{\beta} \log \frac{\int \mathrm{d}\boldsymbol{s}\, \mathrm{e}^{-\beta[F(\boldsymbol{s})+V(\boldsymbol{s})]}}{\int \mathrm{d}\boldsymbol{s}\, \mathrm{e}^{-\beta F(\boldsymbol{s})}} + \int \mathrm{d}\boldsymbol{s}\, p(\boldsymbol{s}) V(\boldsymbol{s}), \tag{3.20}$$

where $p(\boldsymbol{s})$ is a normalized probability distribution. This functional has some interesting properties: It is invariant under the addition of a constant to the bias potential $V(\boldsymbol{s})$, that is, $\Omega[V+c] = \Omega[V]$, and is a convex functional, which means $\Omega\left[\frac{V_1+V_2}{2}\right] \leq \frac{1}{2}\Omega[V_1] + \frac{1}{2}\Omega[V_2]$.

The stationary point of this functional is given up to a constant by

$$V(\boldsymbol{s}) = -F(\boldsymbol{s}) - \frac{1}{\beta} \log p(\boldsymbol{s}), \tag{3.21}$$

which, due to the convexity of $\Omega[V]$, is the global minimum. At this minimum, the sampling of the CVs is according to $p(\boldsymbol{s})$, which is consequently called a “target distribution”. This can be verified by inserting the bias potential into the biased probability distribution of eq. (3.3).

In Ref. 24 it was shown that the functional $\Omega[V]$ is related to the Kullback-Leibler divergence (or relative entropy). Using the Kullback-Leibler divergence [160]

$$D_{\mathrm{KL}}(P(x)\,|\,Q(x)) = \int \mathrm{d}x \log\left(\frac{P(x)}{Q(x)}\right) P(x), \tag{3.22}$$

which is a measure of the distance of the probability distribution $Q(x)$ from the distribution $P(x)$, or in other words, the relative entropy from Q to P, the functional can be rewritten as

$$\beta\Omega[V] = D_{\mathrm{KL}}(p\,|\,P_V) - D_{\mathrm{KL}}(p\,|\,P). \tag{3.23}$$

Because only the first term depends on the bias potential, minimizing the functional is equivalent to minimizing the Kullback-Leibler divergence from the biased distribution to the target distribution.

3.2.2 *Target distribution*

By minimizing $\Omega[V]$, a bias potential can be constructed that leads to a sampling of the CVs according to the specified target distribution $p(\boldsymbol{s})$. The most straightforward choice of the target distribution is a uniform target distribution, leading to completely flat sampling in CV space. Inserting a uniform distribution into eq. (3.21) yields the same expression as eq. (3.8). Similar to the basic metadynamics method, this means that the stationary bias potential fills all the wells of the underlying free energy surface and is given exactly by the negative of the free energy up to some constant. However, the constant is not time-dependent but static in this case.

Another choice that is inspired by the well-tempered metadynamics variant is the so-called well-tempered target distribution [18, 159] given by a normalized version of eq. (3.14)

$$P_V(\boldsymbol{s}) = \frac{\exp\left[-\frac{\beta F(\boldsymbol{s})}{\zeta}\right]}{\int \mathrm{d}\boldsymbol{s} \exp\left[-\frac{\beta F(\boldsymbol{s})}{\zeta}\right]} = \frac{[P(\boldsymbol{s})]^{1/\zeta}}{\int \mathrm{d}\boldsymbol{s} [P(\boldsymbol{s})]^{1/\zeta}}, \tag{3.24}$$

where $\zeta \geq 1$ is a biasing factor that can be set freely by the user. Just as in WTMetaD, for finite ζ the barriers are just reduced, but not completely flattened. This focuses sampling on the low energy regions that are usually most relevant while enhancing fluctuations in a controlled manner to allow for transitions between the desired states on shorter time scales. In practice, this distribution requires knowledge of the a priori unknown free energy surface $F(\boldsymbol{s})$. Therefore, an iterative scheme was designed that updates the target distribution based on the current estimate of the free energy surface during the course of the simulation. Starting with some initial distribution $p^{(0)}(\boldsymbol{s})$, that is usually taken to be the uniform distribution, the update rule of the well-tempered target distribution is given by

$$p^{(k+1)}(\boldsymbol{s}) = \frac{\exp\left[-\beta\zeta^{-1} F^{(k+1)}(\boldsymbol{s})\right]}{\int \mathrm{d}\boldsymbol{s} \exp\left[-\beta\zeta^{-1} F^{(k+1)}(\boldsymbol{s})\right]}. \tag{3.25}$$

The current estimate of the free energy is obtained from eq. (3.21)

$$F^{(k+1)}(\boldsymbol{s}) = -V^{(k)}(\boldsymbol{s}) - \beta^{-1} \log p^{(k)}(\boldsymbol{s}), \tag{3.26}$$

using the bias potential $V^{(k)}$ and target distribution $p^{(k)}$ from the previous iteration.

In general, the framework of target distributions is very flexible and allows tailoring specific forms of sampling by the VES method: There are only minimal requirements and nearly every normalized probability distribution can be chosen. This can be used to prohibit sampling in undesired regions of the CV space or to localize sampling in the desired parts [17, 148]. More complicated schemes have also been proposed, such as targeting the multithermal-multibaric ensemble [161]. In practice, the well-tempered target distribution is an appropriate choice for most applications.

3.2.3 *Optimization schemes*

Up to now, many properties of the stationary point of the functional $\Omega[V]$ were described but not how to actually find the minimum. In practice, the minimization of the functional $\Omega[V]$ is performed by assuming a functional form of the bias potential $V(\boldsymbol{s}, \boldsymbol{\alpha})$ that depends on a set of variational parameters $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_M\}$. Thus the task is now the minimization of the multi-dimensional function $\Omega(\boldsymbol{\alpha})$ instead of an abstract functional. For this, the variational parameters are typically updated iteratively via some optimization scheme. Most optimization schemes require the gradient $\nabla\Omega(\boldsymbol{\alpha})$, whose elements are defined as

$$\frac{\partial \Omega(\boldsymbol{\alpha})}{\partial \alpha_i} = -\left\langle \frac{\partial V(\boldsymbol{s}, \boldsymbol{\alpha})}{\partial \alpha_i} \right\rangle_{V(\boldsymbol{\alpha})} - \left\langle \frac{\partial V(\boldsymbol{s}, \boldsymbol{\alpha})}{\partial \alpha_i} \right\rangle_{p} \tag{3.27}$$

where the expectation values are obtained either over the bias potential or over the target distribution. If the second derivatives are also required, the elements of the Hessian matrix $\mathbf{H}(\boldsymbol{\alpha})$ can be defined similarly as

$$\begin{aligned} \frac{\partial^2 \Omega(\boldsymbol{\alpha})}{\partial \alpha_i \alpha_j} = &- \left\langle \frac{\partial^2 V(\boldsymbol{s}, \boldsymbol{\alpha})}{\partial \alpha_i \partial \alpha_j} \right\rangle_{V(\boldsymbol{\alpha})} - \left\langle \frac{\partial^2 V(\boldsymbol{s}, \boldsymbol{\alpha})}{\partial \alpha_i \partial \alpha_j} \right\rangle_{p} \\ &+ \beta \operatorname{Cov}\left[\frac{\partial V(\boldsymbol{s}, \boldsymbol{\alpha})}{\partial \alpha_i}, \frac{\partial V(\boldsymbol{s}, \boldsymbol{\alpha})}{\partial \alpha_j} \right]_{V(\boldsymbol{\alpha})}, \end{aligned} \tag{3.28}$$

where $\operatorname{Cov}[\dots]$ denotes the covariance.

Averaged stochastic gradient descent

Due to statistical sampling, the estimates of the gradient and Hessian are generally noisy. Therefore, the minimization of $\Omega(\boldsymbol{\alpha})$ is performed using stochastic optimization algorithms. In particular, the averaged stochastic gradient descent algorithm from Ref. 162 has proven a convenient choice. In this algorithm, in addition to the instantaneous parameters, their averages $\bar{\boldsymbol{\alpha}}^{(n)} = \frac{1}{n+1}\sum_{i=0}^{n} \boldsymbol{\alpha}^{(i)}$ are used. The parameters are updated according to the following recursive equation

$$\boldsymbol{\alpha}^{(n+1)} = \boldsymbol{\alpha}^{(n)} - \mu \left[\nabla\Omega(\bar{\boldsymbol{\alpha}}^{(n)}) + H_{\Omega}(\bar{\boldsymbol{\alpha}}^{(n)})(\boldsymbol{\alpha}^{(n)} - \bar{\boldsymbol{\alpha}}^{(n)}) \right], \tag{3.29}$$

where μ is a constant step size and the gradient and Hessian are obtained from the averaged parameters. The bias potential is constructed from the averaged parameters only. The averaging results in small changes per parameter update, which leads to a smooth convergence of the bias potential and the estimated FES. The averaging also allows using relatively few samples per iteration, usually on the order of 1,000 is sufficient. In practice, only the diagonal part of the Hessian is used for the optimization. In Figure 3.3, the time evolution of an exemplary set of parameters $\boldsymbol{\alpha}$, that is being optimized by this algorithm, is shown. The averaged parameters evolve smoothly and do not change significantly at the end of the displayed 2,000 iterations. On the other hand, the instantaneous parameters fluctuate still significantly around their respective average value.

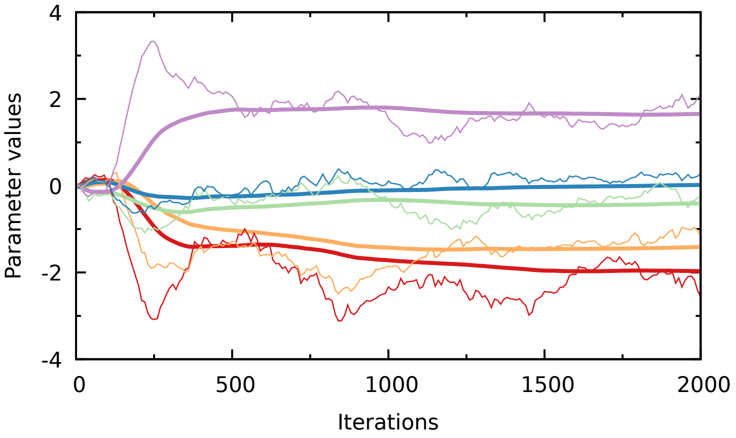


Figure 3.3: Illustration of the averaged stochastic gradient descent optimization method. Shown is the time evolution of five parameters (thin lines) and their averages (thick lines). While the instantaneous parameters still fluctuate after 2,000 iterations, the averages have already converged.

Adaptive moment estimation (Adam)

Stochastic optimization algorithms have recently seen lots of research due to the popularity of machine learning methods. Many new algorithms with promising properties have been developed, two popular ones are the adaptive gradient (AdaGrad) algorithm [163], and the adaptive moment estimation (Adam) algorithm [164]. Here, only the Adam stochastic gradient descent algorithm will be described in detail while the reader is referred to other literature (e.g., Ref. 165) for a more exhaustive overview. The Adam algorithm was previously introduced for VES in combination with neural networks [21]. It uses a running average $\boldsymbol{m}(\boldsymbol{\alpha})$ and second moment $\boldsymbol{v}(\boldsymbol{\alpha})$ of the gradient $\boldsymbol{g}(\boldsymbol{\alpha}) = \nabla\Omega(\boldsymbol{\alpha})$, where at iteration k these are updated with the current value of the gradient according to

$$\boldsymbol{m}^{(k)} = c_1 \boldsymbol{m}^{(k-1)} + (1 - c_1)\boldsymbol{g}^{(k)}, \tag{3.30a}$$

$$\boldsymbol{v}^{(k)} = c_2 \boldsymbol{v}^{(k-1)} + (1 - c_2)\boldsymbol{g}^{(k)}\boldsymbol{g}^{(k)}. \tag{3.30b}$$

Here, the explicit dependency on the parameters $\boldsymbol{\alpha}$ was omitted for a more concise notation. The meta parameters $c_1, c_2 \in (0, 1)$ determine the "memory" of the averaging process, the default values given in the literature are $c_1 = 0.9$ and $c_2 = 0.999$. The Adam optimization algorithm does not use the second derivatives in form of the Hessian.

As $\boldsymbol{m}$ and $\boldsymbol{v}$ are initialized to zero, the algorithm of eq. (3.30) would be biased towards that value in the first few steps. To overcome this, a correction was proposed directly with the algorithm

$$\hat{\boldsymbol{m}}^{(k)} = \frac{\boldsymbol{m}^{(k)}}{1 - c_1^k}, \tag{3.31a}$$

$$\hat{\boldsymbol{v}}^{(k)} = \frac{\boldsymbol{v}^{(k)}}{1 - c_2^k}, \tag{3.31b}$$

that unbiases the average and variance. Because c_1 and c_2 are smaller than one, the denominator quickly approaches 1 and the the correction is only relevant for small k, that is, the first iterations. The update rule for the coefficients is given by

$$\boldsymbol{\alpha}^{(k)} = \boldsymbol{\alpha}^{(k-1)} - \frac{\eta}{\sqrt{\hat{\boldsymbol{v}}^{(k)}} + \epsilon} \hat{\boldsymbol{m}}^{(k)} \tag{3.32}$$

where $\epsilon = 10^{-8}$ is a small parameter preventing division by zero and η is a stepsize to be set by the user. The algorithm scales the stepsize by the variance, thus coefficients with larger fluctuations get updated in smaller steps. For the stepsize, the literature proposes a default value of $\eta = 0.001$. Since the algorithm was developed with machine learning applications in mind, this proposed default set of parameters might not be well-suited for the application in VES.

Of note is in that context, that both AdaGrad and Adam have yet not been used "as-is" in the typical fashion of the VES method but only in modified ways: a novel optimization combination of AdaGrad and the averaged stochastic gradient descent was proposed and employed by Invernizzi and Parrinello [25], and the Adam optimizer was only used in combination with a neural network by Bonati et al. [21]. Further investigation of their behavior is therefore required before they can be employed in production simulations.

3.2.4 *Basis functions*

To obtain a functional form of the bias potential, a natural way is to use a linear expansion in some set of basis functions $\boldsymbol{f} = \{f_0, f_1, f_2, \ldots, f_M\}$,

$$V(\boldsymbol{s}, \boldsymbol{\alpha}) = \sum_{i=0}^{M} \alpha_i f_i(\boldsymbol{s}). \tag{3.33}$$

The expressions for the gradient $\nabla\Omega(\boldsymbol{\alpha})$ and the Hessian $H_\Omega(\boldsymbol{\alpha})$ then simplify to

$$\frac{\partial\Omega(\boldsymbol{\alpha})}{\partial\alpha_i} = -\left\langle f_i(\boldsymbol{s})\right\rangle_{V(\boldsymbol{\alpha})} + \left\langle f_i(\boldsymbol{s})\right\rangle_{p}, \tag{3.34a}$$

$$\frac{\partial^2\Omega(\boldsymbol{\alpha})}{\partial\alpha_i\alpha_j} = \beta\,\mathrm{Cov}[f_j(\boldsymbol{s}), f_i(\boldsymbol{s})]_{V(\boldsymbol{\alpha})}. \tag{3.34b}$$

If the number of CVs is larger than one, multidimensional functions would be required. In practice, the higher dimensional bias potentials are usually constructed as a tensor product of one-dimensional basis functions. For example, in the case of two CVs, the expansion can be written as

$$V(s_1, s_2, \boldsymbol{\alpha}) = \sum_{i,j} \alpha_{i,j}\, g_i(s_1)\, h_j(s_2), \tag{3.35}$$

where the set of one-dimensional basis functions $\boldsymbol{g}(s_1)$ is along the first CV and the basis functions $\boldsymbol{h}(s_2)$ along the second CV. Because higher dimensional bias potentials are easily constructed via the tensor product as in eq. (3.35), the following discussions will be limited to the description of one-dimensional basis functions.

The VES method was historically designed with basis functions from spectral methods in mind, such as the Fourier series or polynomials of the Gegenbauer family [166].

Fourier series

The Fourier series is a periodic basis set consisting of plane waves with different frequencies. Therefore, it is employed when dealing with CVs defined on a periodic domain. Its complex form is given by $V(s) = \sum_k \alpha_k \mathrm{e}^{iks}$, where i is the imaginary unit and $k \in \mathbb{Z}$ goes also over the negative integers. The coefficients α are generally complex values, but the bias potential is desired to be a real-valued function. To ensure real-valued coefficients, instead of the concise complex form an expansion in cosine and sine functions

$$V(s) = \alpha_0 + \sum_{k=1}^{M/2} \alpha_k^{\cos} \cos\left(\frac{2\pi}{p} ks\right) + \sum_{k=1}^{M/2} \alpha_k^{\sin} \sin(\frac{2\pi}{p} ks) \tag{3.36}$$

is used for the VES method, where $\alpha_k^{\cos}$ and $\alpha_k^{\sin}$ are two independent sets of $M/2$ real-valued parameters and the desired interval of the bias potential is given by $[a, b]$ with periodicity $p = b - a$.

Legendre and Chebyshev polynomials

The Gegenbauer polynomials, also called "ultraspherical" polynomials, are a family of polynomials that includes the Legendre and Chebyshev polynomials as special cases. For a fixed value of the parameter m, they are orthogonal on the interval $[-1,1]$ with respect to the weight function $w(x) = (1-x^2)^{m-1/2}$ and can be defined via a recurrence relation as

$$
\begin{aligned}
G_1^{(m)}(x) &= 1, && (3.37a)\\
G_2^{(m)}(x) &= 2mx, && (3.37b)\\
(n+1)G_{n+1}^{(m)}(x) &= 2(n+m)x\,G_n^m(x) - (n+2m-1)G_{n-1}^{(m)}(x). && (3.37c)
\end{aligned}
$$

Two special cases are the Legendre polynomials, for $m = 1/2$, and the Chebyshev polynomials of the first kind, for $m \to 0$ [167]. Their recurrence relations are given by

$$L_1(x) = 1, \tag{3.38}$$

$$L_2(x) = x, \tag{3.39}$$

$$L_{n+1}(x) = \frac{2n+1}{n+1}\,x\,L_n(x) - \frac{n}{n+1}\,L_{n-1}(x), \tag{3.40}$$

for the Legendre polynomials and

$$C_1(x) = 1, \tag{3.41}$$

$$C_2(x) = x, \tag{3.42}$$

$$C_{n+1}(x) = 2x\,C_n(x) - C_{n-1}(x), \tag{3.43}$$

for the Chebyshev ones. As the previously described polynomials are defined intrinsically on the interval $[-1,1]$, they need to be scaled and shifted when employed on different intervals. For a given interval $[a,b]$, the following function is used to transform $t \in [a,b]$ to $x \in [-1,1]$

$$x(t) = \frac{2t-(a+b)}{(b-a)}. \tag{3.44}$$

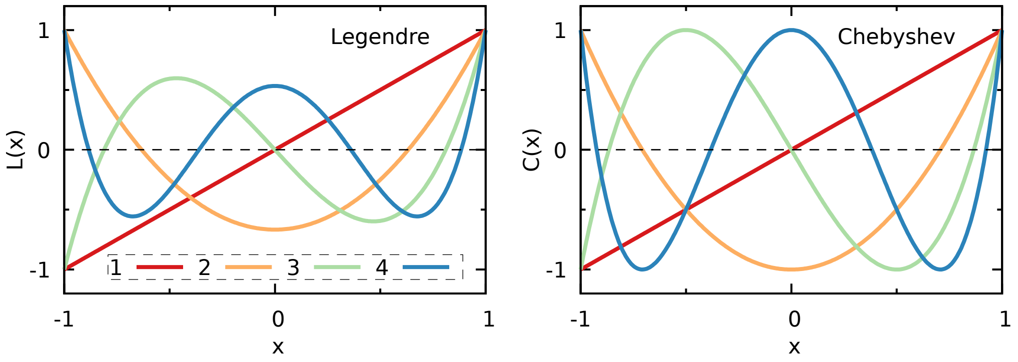


Figure 3.4: Functional form of the first four Legendre (*left panel*) and Chebyshev (*right panel*) polynomials.

Exemplary, the functional form of the first four Legendre and Chebyshev polynomials can be seen in Figure 3.4. Besides their clear similarity, the different weighting over the interval is also apparent.

Which kind of basis function should be employed depends on the problem under investigation. For example, when using a periodic CV (e.g. for a molecular angle or torsion), the Fourier series is clearly better suited than non-periodic polynomials. Nevertheless, it is often not clear a priori what basis functions will yield the best results. Until now, no extensive study regarding the choice of basis function for the VES method has been published. This is one of the topics under investigation in this work, which is considered in detail starting with Chapter 4. There, after some theoretical considerations on the choice of good basis functions are provided, the focus is on the development of new basis functions for the VES method. These new functions are “localized” in CV space, in contrast to the “global” functions that were just presented, that is, each function is non-zero on a small part of the CV space only. After their introduction, a comprehensive investigation of their performance, also in comparison to the established basis functions, is executed by means of simulating several model systems as well as a realistic chemical systems.

3.2.5 *Reweighting*

The FES can be determined directly from the bias potential through eq. (3.21) after the optimization process has reached a point close to the minimum of the functional, that is the bias has converged sufficiently and is quasi-stationary.

Alternatively, the FES can be obtained, both for the biased CVs and also for any other set of CVs, in a similar vein as eqs. (3.16, 3.17, 3.19) for metadynamics simulations. This is done by using a reweighted histogram where each configuration is weighted by the bias acting on it, $P(\vec{\boldsymbol{r}}) \propto P_V(\vec{\boldsymbol{r}})\mathrm{e}^{\beta V[\boldsymbol{s}(\vec{\boldsymbol{r}})]}$. While the VES bias potential is time-dependent, it quickly becomes quasi-stationary. Therefore, this reweighting procedure is valid after a short initial transient in the time series that is excluded from the calculation. The unbiased average of an observable can then be calculated as

$$\langle O(\vec{\boldsymbol{r}})\rangle = \frac{\langle O(\vec{\boldsymbol{r}})\mathrm{e}^{\beta V[\boldsymbol{s}(\vec{\boldsymbol{r}})]}\rangle_V}{\langle \mathrm{e}^{\beta V[\boldsymbol{s}(\vec{\boldsymbol{r}})]}\rangle_V} \tag{3.45}$$

from the expectation values obtained in the biased ensemble. The determination of the FES as a function of any desired variables can be done via eq. (3.19) just as for the metadynamics method.

Note that differently from metadynamics [77, 143], there is generally no need to account for time-dependent constants when performing reweighting with VES. Furthermore, under certain conditions, the VES method can also be used to obtain kinetic properties [150].

3.3 MULTIPLE REPLICA METHODS

The molecular and Langevin dynamics methods presented in Chapter 2 generate samples by moving a single representation of the system through configuration space. Especially for systems with high barriers and unlikely transitions between different metastable states (“rare event systems”), getting an adequate representation of the relevant parts of the system requires many independent samples and therefore long simulation times. The previous sections presented two methods to push the system towards the most relevant parts of the configuration space, metadynamics and VES.

Alternatively, or on top of these methods, parallelization can be used: multiple copies of the system, usually with different initial conditions, are simulated at the same time. This can be seen as imaginary particles ("walkers" or "replicas") following the same underlying dynamics. While using snapshots from different copies of the system can already help to get better coverage of the configuration space, that is running multiple short simulations instead of a long one, additional benefits can be realized by introducing interactions between the replicas. The individual replicas then usually do not all sample the desired distribution. Typically, either a single one of the replicas samples the correct distribution and is augmented by the others, or the correct distribution is only sampled by way of an ensemble average over multiple replicas.

Different strategies have been developed, very popular is the family of parallel tempering or replica exchange algorithms [168–170], where the replicas of the system are simulated in a ladder of temperatures. The replica at the lowest temperature is the one of actual interest, while the ones at higher temperatures explore the configuration space faster. Periodically the replicas attempt to swap their current configuration with the ones at adjacent temperatures via Monte Carlo moves, which provides more comprehensive sampling also at the desired temperature.

Another family of multiple replica methods are algorithms that use population dynamics, that is, they use mechanisms that manipulate the distribution of the replicas in the configuration space. While some methods only enhance the dynamics of the replicas with a term depending on all replicas [171, 172], the focus of this work is on "cloning" methods that involve the duplication and killing of replicas [30–35]. A new method belonging to this family will be presented in Part III.

Additionally to the usage of multiple replica methods to improve the sampling in molecular simulations, they can often be combined with other enhanced sampling methods. For example, the previously described metadynamics has been used together with parallel tempering [119], solute tempering [173], or in the form of bias exchange metadynamics [117]. Besides these more sophisticated algorithms, the basic framework of multiple walkers has been employed in combination with many different methods, including metadynamics [132], variationally enhanced sampling [24, 159], or adaptive biasing force [174].

Multiple walkers metadynamics and variationally enhanced sampling

For metadynamics and VES, multiple walkers are implemented by running multiple independent simulations in parallel that are coupled by using the same bias potential. Each of the simulations starts from different initial conditions but is biased by the same bias potential $V(\mathbf{s})$ and also contributes equally to its time evolution. For metadynamics, this means that Gaussians are added to the bias potential from the positions of all walkers, typically at regular intervals as in the case of a single simulation. For VES the implementation is even more intuitive: The ensemble averages for the gradient and hessian of eqs. (3.27, 3.28) are collected from all walkers simultaneously. The just introduced variants of multiple walkers for metadynamics and VES are used for applications in Chapter 7. Besides the increased number of samples, the better coverage of different parts of the CV space can help to improve the optimization in the VES method. However, there is an upper limit for the number of walkers that can be used: Beyond a certain number their motion often becomes correlated. The usage of cloning algorithms might help to break these correlations.

3.4 PERFORMANCE MEASURES

To evaluate and compare the performance of enhanced sampling simulations, different performance measures can be considered. While many different quantities are used in the literature [19, 133, 138, 159, 175, 176], the focus will here be on measures that assess the accuracy of the free energy surface obtained from the simulation.

For biased simulations with the metadynamics or VES methods presented in Sections 3.1 and 3.2, respectively, a FES estimate can be estimated directly from the bias via the eqs. (3.8, 3.15, 3.21). Alternatively the FES can be obtained by reweighting the samples of the trajectory by eq. (3.19). For unbiased simulations, eq. (3.2) is used directly: The required equilibrium probability distribution can be estimated from the obtained samples, for example by histogramming or a kernel density estimation. When using a suitable, normalized histogram H, the free energy can be calculated by

$$F(\boldsymbol{s}) = -\beta^{-1} \log H(\boldsymbol{s}) + C, \tag{3.46}$$

where the constant C is typically chosen such that $\min F(\boldsymbol{s}) = 0$.

To measure the quality of a FES estimate $F(\boldsymbol{s})$, it can be compared to a known reference $F_{\mathrm{ref}}(\boldsymbol{s})$. Since the FES is usually determined only up to a constant (see eq. (3.2)), it needs to be aligned. The simplest option would be to choose the constant C such that $\min F(\boldsymbol{s}) = 0$ for both the obtained FES and the reference. If the obtained FES is not accurate at its minimum, this choice negatively influences the outcome of any comparison. It is therefore better to shift the FES by the average value of the low-energy region Γ, that is

$$\tilde{F}(\boldsymbol{s}) = F(\boldsymbol{s}) - \int_{\Gamma} \mathrm{d}\boldsymbol{s}\, F(\boldsymbol{s}) + \int_{\Gamma} \mathrm{d}\boldsymbol{s}\, F_{\mathrm{ref}}(\boldsymbol{s}). \tag{3.47}$$

Typically, Γ is chosen as the area of CV space where the reference free energy is below a certain threshold ν_{shift}

$$\int_{\Gamma} \mathrm{d}\boldsymbol{s} = \int \mathrm{d}\boldsymbol{s}\, \theta(\nu_{\mathrm{shift}} - F_{\mathrm{ref}}(\boldsymbol{s})), \tag{3.48}$$

with θ being the Heaviside step function.

Root mean square error

A possible error measure is to calculate the root mean square (RMS) error of the FES with respect to the reference, as it was done in Refs. 133, 159:

$$\epsilon = \sqrt{\frac{\int \mathrm{d}\boldsymbol{s}\, \left[\tilde{F}(\boldsymbol{s}) - F_{\mathrm{ref}}(\boldsymbol{s})\right]^2 \theta(\nu_{\mathrm{error}} - F_{\mathrm{ref}}(\boldsymbol{s}))}{\int \mathrm{d}\boldsymbol{s}\, \theta(\nu_{\mathrm{error}} - F_{\mathrm{ref}}(\boldsymbol{s}))}} \tag{3.49}$$

The region of interest is defined similarly as for the shifting, that is, by the regions where the reference free energy is smaller than another parameter ν_{error}. In this measure, every point within the region of interest contributes equally. Typically, the focus of simulations is on the low energy regions and on obtaining accurate estimates of the relative differences between states.

Kullback-Leibler divergence

Another possible measure, that weights the low energy regions significantly more than the high energy regions, is the Kullback-Leibler divergence that was already introduced in eq. (3.22). To calculate it, the FES has to be transferred to a probability distribution, $P(\boldsymbol{s}) = \exp(-\beta F(\boldsymbol{s})) / \int \mathrm{d}\boldsymbol{s} \exp(-\beta F(\boldsymbol{s}))$. Then the distance between the obtained distribution $P(\boldsymbol{s})$ and the reference distribution $P_{\text{ref}}(\boldsymbol{s})$ is given by

$$D_{\text{KL}}(P_{\text{ref}}(\boldsymbol{s}) \,|\, P(\boldsymbol{s})) = \int \mathrm{d}\boldsymbol{s} \log \left(\frac{P_{\text{ref}}(\boldsymbol{s})}{P(\boldsymbol{s})} \right) P_{\text{ref}}(\boldsymbol{s}). \tag{3.50}$$

For this measure, the areas with high probabilities (corresponding to low free energies) contribute most due to the weighting of the integral with the reference probability. Because of the exponential dependency of the probability on the FES, this effect is rather large and already the regions with moderately high free energies do not significantly contribute to this measure. Therefore, the high energy regions do not need to be excluded explicitily as for the RMS error. Of note is, that the Kullback-Leibler divergence is not symmetric, that is $D_{\text{KL}}(P_{\text{ref}}(\boldsymbol{s}) \,|\, P(\boldsymbol{s})) \neq D_{\text{KL}}(P(\boldsymbol{s}) \,|\, P_{\text{ref}}(\boldsymbol{s}))$, and it is therefore not a metric in the mathematical sense, although it can be related to actual metrics via Pinsker's inequality [177].

Free energy difference ΔF

For chemical and physical applications, often not the full free energy surface but only the relative free energy difference between (meta-)stable states is of interest.

Taking two states A and B, it can be determined from the FES by integrating over the associated probabilities

$$\Delta F_{A,B} = F_A - F_B = -\frac{1}{\beta}\log\frac{P_A}{P_B} = -\frac{1}{\beta}\log\frac{\int_A \mathrm{d}\boldsymbol{s}\ \exp[-\beta F(\boldsymbol{s})]}{\int_B \mathrm{d}\boldsymbol{s}\ \exp[-\beta F(\boldsymbol{s})]}, \tag{3.51}$$

where the domains of integration are the regions in CV space associated with the states A and B, respectively. This only represents the experimentally accessible free energy difference if the CVs $\boldsymbol{s}$ clearly separates the states A and B.

Averaging over multiple simulations

The results from a single simulation might not yield a good representation of equilibrium properties [71]. It is therefore good practice to use quantities that represent the results of an ensemble of multiple independent runs, started for example with different initial conditions, to measure the performance of a method. A simple way to achieve this is to average over measures for the individual runs. For example, the average RMS error of N independent simulations is given by

$$\langle \epsilon \rangle = \frac{1}{N}\sum_i^N \epsilon_i. \tag{3.52}$$

A benefit of this approach is that it also allows to assess fluctuations between runs, by calculating the standard deviation

$$\sigma_\epsilon = \sqrt{\frac{1}{N-1}\sum_i^N (\epsilon_i - \langle \epsilon \rangle)^2}. \tag{3.53}$$

Here, the factor of $N-1$ in the denominator arises from Bessel's correction, which unbiases the estimate when plugging in the mean value calculated from the same data. From the standard deviation the standard error is obtained via

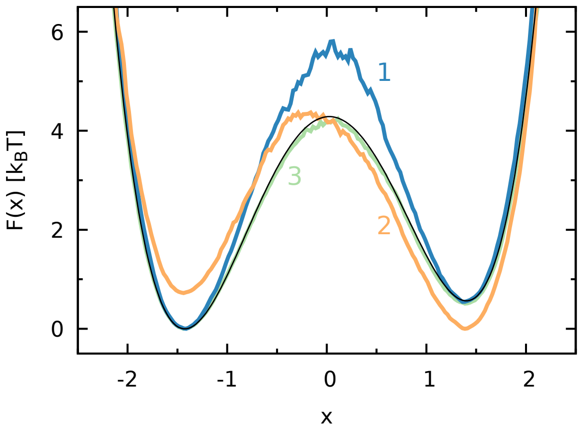
$$\hat{\sigma}_\epsilon = \frac{\sigma_\epsilon}{\sqrt{N}}. \tag{3.54}$$

Whether to use the standard deviation or the standard error depends on the property under investigation: The standard deviation measures the fluctuations of the different values and is therefore useful to assess the quality of predictions from individual runs. It is typically preferred when comparing the performance of different methods. On the other hand, if of interest is the quality of the mean value, that is how much the prediction can be trusted, the standard error is the quantity of choice.

Choice of measure

Which performance measure to use depends on the problem under investigation, if a reference FES is available, and personal preferences. An illustration of the differences between the presented measures can be seen in Figure 3.5. There, three quite different estimates of a simple one-dimensional FES, consisting of two states separated by a barrier, are shown. The first estimate, given in blue, offers a good representation of the lower-lying regions of the FES and gets the minima correctly, while drastically overestimating the barrier between the states. On the other hand, the second shown estimate, given in orange, captures the height of the barrier approximately but does not yield a good estimate of the minima. The third estimate, shown in green, is pretty close to the reference but still has small deviations over the full range, such as a slightly smaller FES estimate in the barrier region and the right minimum.

This can be seen when looking at the calculated values of the free energy difference also provided in the figure. While the first estimate matches the reference value closely ($\Delta F_{\mathrm{ref}} = 0.528$), the second one is completely off and even predicts the wrong state to be the global minimum. The third one also yields a good estimate that is off by only $0.05\,k_{\mathrm{B}}T$. As the value of the free energy difference is very sensitive to small changes of the FES around the minima, it typically fluctuates on these scales also when taking FES estimates at different times of a simulation and the accuracy is therefore limited. When comparing with experimental results, it is also important to keep the measurement uncertainty of these methods in mind. Typically the chemical accuracy, which is



	ϵ	D_{KL}	ΔF
1	0.59	4.8×10^{-4}	0.53
2	0.57	4.8×10^{-3}	−0.71
3	0.05	1.4×10^{-5}	0.47

Figure 3.5: Illustration of the different performance measures.
Left side: Three exemplary FES estimates (1 blue; 2 orange; 3 green) together with the reference (thin black).
Right side: Values of the different performance measures for the two estimates. For the RMS error, the parameters were $\nu_{\text{shift}} = 2\,k_\mathrm{B}T$, $\nu_{\text{error}} = 5\,k_\mathrm{B}T$. The states for the estimates of the free energy difference were defined by separating the FES at $x = 0$ (i.e. $A : x \in (-\infty, 0)$, $B : x \in (0, \infty)$), and the value for the reference FES is $\Delta F_{\text{ref}} = 0.528$.

the smallest uncertainty that is possible in experiments, is given as 1 kcal/mol which at 300 K corresponds to approximately $0.4\,k_\mathrm{B}T$.

Similar to this approach, the Kullback-Leibler divergence also emphasizes the low energy regions which results in an order of magnitude difference of the measure between the first two estimates, in favor of the first estimate. The third estimate has by far the lowest KL divergence value because it best resembles the correct distribution.

Differently, the RMS error weights all parts lower than $\nu_{\text{error}} = 5\,k_\mathrm{B}T$ of the reference equally, that is roughly the region $[-2, 2]$, incorporating both minima and the barrier region. Here, the third estimate is again clearly the best, while the first two estimates yield nearly the same values. This measure therefore captures the suboptimal result of the first estimate in the barrier region, in contrast to the other measures. As it is more directly related to the visible similarity of the FES estimates to the reference, it also matches the expected result from visual inspection of the figure: Both estimates deviate drastically from the reference in roughly equally sized parts of the investigated area.

It has become clear from the example, that the choice of measure depends on the properties of the FES that are of interest for the problem at hand. Generally speaking, the Kullback-Leibler divergence measures the distance between the probability distributions and therefore shows if the sampling of the simulation is correct. This might be useful if the sampling in general is the focus of the study, as it is for the birth-death augmented sampling in part III of this thesis. On the other hand, the RMS error is more flexible and allows for the investigation of more specific behavior. For example, when looking at the performance of an enhanced sampling method, the interesting parts are typically not only the regions of lowest energy, as these would be sufficiently sampled even without the method. Using the RMS error with a threshold energy on the order of the barrier energy will put more emphasis on the typically rarely sampled parts that the method gives access to. It also yields results that are more in line with visual comparisons of the FES estimates and the reference. Calculating the free energy difference between states is useful to assess how well chemical properties are matched, although the fluctuation of this measure even with small changes should be kept in mind.

Part II

DAUBECHIES WAVELETS AS BASIS FUNCTIONS FOR VARIATIONALLY ENHANCED SAMPLING

> *"Mathematicians have various ways of judging the merits of new theorems and constructions. One very important criterion is esthetic — some developments just 'feel' right, fitting, and beautiful. [...] Another important criterion for according merit to some particular piece of mathematics is the extent to which it can be useful in applications; this is the criterion almost exclusively used by nonmathematicians."*
>
> — Ingrid Daubechies [178]

4

PROPERTIES OF GOOD BASIS FUNCTIONS

4.1 INTRODUCTION

For the successful application of the VES method, the choice of a good set of basis functions plays a crucial role. These are used to represent the bias potential and can thus severely limit not only the kind of enhanced sampling that can be achieved but also negatively impact the accuracy of quantities obtained from the bias, such as the FES via eq. (3.21).

It is therefore surprising that after the fundamental principles of the method had been developed in Refs. 17, 159, the choice of basis functions played only a marginal role in the literature of the VES method. The supplemental material of the first VES paper [17] included results for different numbers of basis functions of the fourier set. For the simulation of Alanine dipeptide in vacuum and water, two dihedral angles were used as biased CVs, where the number of fourier basis functions per CV was modified in the range from 7 to 13. The authors found that all main features of the FES are observed even with the minimal set (7 basis functions per CV), although due to the limited flexibility of the small set some finer features were not ideally represented in the solvated simulations. It was noted that a better representation even with few basis functions can be obtained by reweighting the trajectories.

In Ref. 159 another discussion on the choice of the number of basis functions was given. There, alanine tetrapetide was simulated in water, where the focus was on the different behavior of the method when using a uniform or a well-tempered target distribution. In both cases, the number of basis functions per biased CV (used were three different dihedral angles) was varied in the range between 7 and 21. The authors concluded that when using a uniform target distribution, a smaller number of functions resulted in better convergence behavior, which shows that a larger set is not always better. The performance differences were found to be neglectable when using a well-tempered target distribution.

A special representation of the bias potential was introduced and discussed by Bonati et al. [21]: Instead of using a fixed set of basis functions, a deep neural network was employed to construct the bias potential by nonlinear combination of the CV values over several layers. For this variant, assuming perfect convergence, the final form of the bias potential depends on the functional form of the individual neurons and the architecture, that is the number of neurons and network layers. The focus of the paper was on the general introduction of the method and not on the study of these parameters though, such that only a short side note on the choice of the network architecture was given in the supplemental material. Arguably, this adds to the discussion about good basis function a rather orthogonal answer, by introducing a neural network representation instead of a linear expansion in fixed basis functions. Neural networks shine most when they are employed to yield a good representation starting from many input features that cannot be easily obtained with more direct methods. In the case of VES, the number of CVs used to build the bias is small such that the full power of a neural network is typically not required. Of note is also that the approach was not designed to get a perfect representation of the bias (i.e., the minimum of the functional) but rather to get a suitable approximation in a few iterations to then use as a static bias for further simulations, which it accomplishes successfully for the tested systems.

Besides these few examples, which also only touch on certain aspects of the topic, contributions that compare the performance of fundamentally different basis sets are still missing in the community. Additionally, the development of new, and possibly better, basis functions has been minimal since the introduction of the method in 2014, besides the already mentioned neural network representation [21] and free energy models that try to tailor a custom form of the potential from physical principles [22–25]. In the following, an attempt to address this shortage is made.

So far, the basis functions employed in VES have been the ones introduced in Section 3.2.4, which were plane waves (i.e., Fourier series) [17], Chebyshev polynomials [159], or Legendre polynomials. The usage of such "global" functions is closely related to the idea of using spectral methods for function approximation [166]. Global functions are delocalized in the CV space, or in other words, they are non-zero over the full range of the CV domain to be biased except on isolated points.

Using global, or delocalized, basis functions means that during the optimization process the bias potential will change even in parts of the CV space that the simulation is currently not exploring. Even though the simulation does not provide any samples from these areas for the estimates of the gradient and Hessian of eq. (3.34), the bias potential is still optimized in the full range. While this has not proven to be a significant issue, it is clear that delocalized basis functions might not be the optimal choice, especially for the initial part of the simulation where sampling is typically restricted to the low energy regions.

For that reason, new basis functions that are localized in CV space, which means that each function is non-zero only in a small region of the CV space, are investigated in this work. Examples of such localized basis functions that come to mind would be Gaussians or splines. In fact, in Refs. 19, 20, the authors employed VES with Gaussian basis functions. The results obtained with this VES setup were found to be inferior to some of the results obtained with other enhanced sampling methods also under investigation (such as umbrella sampling [89]), but as no other basis functions were used with VES, it is hard to judge the performance of the Gaussian basis from their results. Although Gaussians are a simple choice and their usage is common in for example kernel density estimation [179, 180] or as basis functions in density functional theory [181–184], they are rarely employed as fixed basis set for general function approximation. In this area, the family of splines is employed frequently as an alternative to polynomials, for example, the multidimensional cubic splines introduced by Habermann and Kindermann [185]. These interpolating splines were designed with simplicity of their implementation in computer code in mind and also allow for a simple multidimensional representation by the tensor product.

Another concept related to finding smoothed approximations of general functions is wavelets. These are the main focus here, in particular the family of orthonormal wavelets introduced by Daubechies [26, 186]. Furthermore, they have an intrinsic multiresolution property that makes it possible to iteratively add more basis functions to refine the previously obtained representation. They have also already been employed for other problems related to molecular or quantum mechanical calculations, such as in the BigDFT variant of density functional theory [27, 28], or to obtain coarse-grained potentials [29].

These three families of possible basis functions, namely wavelets, Gaussians, and splines, are all localized in CV space, that is each basis function contributes to the bias potential only in a specific, predefined range. Therefore, they should not suffer from the issue of the bias potential changing in parts of CV space that the simulation is not currently exploring.

In the following, the term “localized” always refers to the localization in the space of CVs. This is in contrast to the “global” or “delocalized” basis functions, for example the Chebyshev polynomials, that represent only isolated frequencies and are therefore localized in the associated Fourier transformed space.

A discussion about the performance of these new basis functions is laid out in the following chapters of the thesis. As a starting point, some general considerations on the choice of basis functions for VES are given in Section 4.2 In Chapter 5 some fundamental aspects of wavelet theory are presented, with a focus on the family of Daubechies wavelets and their application as basis functions for VES. Subsequently, Chapter 6 deals with their implementation into the VES code, while also other localized basis functions, Gaussians and cubic B-splines, are introduced. Finally, Chapter 7 puts the different basis functions to the test, comparing their performance using several model and atomistic systems.

4.2 THEORETICAL CONSIDERATIONS

In the mathematical sense, a basis of a space of functions is defined as a set of functions with some special properties [187, 188]. Generally, a Lebesgue space L^p is defined as the space of all measurable functions with

$$\int \mathrm{d}x \, |f(x)|^p < \infty, \tag{4.1}$$

and the associated norm $|| \cdot ||_p$. For the representation of the VES bias, which is finite and bounded, the space to be considered is the one of all square-integrable functions L^2, which has the euclidean norm and the well-known inner product $\langle f, g \rangle_2 = \int \mathrm{d}x \, f(x) g(x)$.

A basis of the space has to be complete, that is, any L^2 function can be approximated up to arbitrarily small precision by a finite linear combination of the basis functions. On top of this, the representation has to be unique, such that no two combinations represent the same function. In mathematical terms this means that if any combination of the basis functions yields the zero element, $\sum_i \alpha_i f_i(x) = 0$, then all coefficients must be zero: $\alpha_0 = \alpha_1 = \cdots = \alpha_N = 0$. This property is also called linear independence.

Another related concept is that of orthonormality. An orthonormal basis is given if the elements also satisfy orthonormality

$$\int \mathrm{d}x \, f_i(x) f_j(x) = \delta_{i,j}. \tag{4.2}$$

on top of the previously mentioned condition of completeness (linear independence is automatically satisfied for orthonormal basis functions).

For any orthonormal basis in, for example, L^2, every function g of that space can be represented exactly by the basis functions f_i

$$f = \sum_{i=0}^{\infty} \langle g, f_i \rangle \, f_i = \sum_{i=0}^{\infty} \alpha_i f_i \tag{4.3}$$

with a unique set of expansion coefficients $\boldsymbol{\alpha}$.

Specific to VES

After this theoretical introduction into functional analysis, the question remains what a "good" basis is. For approximation problems via spectral methods, Ref. 166 gives a number of desirable properties for basis functions: (1) completeness, (2) rapid convergence, and (3) easy to compute. To use them as VES bias an additional property is essential: (4) existence of at least one continuous derivative.

Completeness is not essential for the VES method. In practice, it is sufficient to get an adequate representation of the bias. Nevertheless, an incomplete set of basis functions might not have enough variational flexibility. This makes convergence of the bias potential difficult and severely limits the accuracy of the FES obtained directly from the bias via eq. (3.21). However, this can often be corrected by using reweighting according to eq. (3.19).

The second property, rapid convergence, means that a good approximation should be achieved with the fewest possible coefficients. This can be understood better by looking at eq. (4.3): When using a basis with theoretically infinitely many basis functions, such as a Fourier series or Chebyshev polynomials, it is beneficial to have a small error when omitting the remaining higher order basis functions already at low orders of the expansion. For the optimization process in VES this is crucial, as a lower number of basis functions results also in fewer degrees of freedom to optimize. It is clear that the number of required basis functions depends not only on the chosen basis set but also on the problem at hand. On top of this, this number is typically not determinable a priory. In practice, it is common to perform the first simulation with a rather small set to get rapid convergence and add more basis functions if the obtained FES does not capture all features satisfactorily.

The property of easy computability comes from the implementation side: any basis set to be used should not only be implementable in code but desirably also have minimal computational cost for evaluation. To calculate the averages over the bias potential required for the gradient and Hessian of the functional in eq. (3.34), each basis function has to be evaluated at each step of the simulation. Additionally, also the first derivatives have to be evaluated to obtain the force acting on the atoms from the bias potential. This makes the fourth requirement clear: While numerical calculation of the derivatives is possible, these must at least exist in a continuous way. Ideally, the first derivative also has a functional form that is easy to evaluate. Although the evaluation of the $2N$ values (basis functions and derivatives) is typically only a small overhead compared to the evaluation of the atomistic forces, it is nevertheless important to keep it in mind when designing basis functions for the usage in VES.

Established basis functions for VES

For the basis functions that were already described in Section 3.2.4, some insight on the choice can be found in the literature. When using them as basis functions for spectral methods, Boyd [166] says it rather explicit:

"(i) When in doubt, use Chebyshev polynomials unless the solution is spatially periodic, in which case an ordinary Fourier series is better. (ii) Unless you're sure another set of basis functions is better, use Chebyshev polynomials. (iii) Unless you're really, really sure that another set of basis functions is better, use Chebyshev polynomials." [166, p. 10]

Both Legendre and Chebyshev polynomials are part of the same family of orthogonal polynomials, the difference lies in the weighting on their intrinsic interval $[-1, 1]$. The Chebyshev weighting, $w(x) = (1 - x^2)^{-1/2}$, is heavily biased towards the endpoints and the best choice for expanding general functions. On the contrary, the Legendre polynomials have no weighting at all. Their error is thus smaller than for Chebyshev polynomials on most of the interval but much larger near the endpoints. In Ref. 166 the difference in error when truncating the expansion after N basis functions is given as roughly $\mathcal{O}(N^{1/2})$ larger for Legendre polynomials when considering the whole range.

For the expansion of the bias potential in the VES method, the boundary areas are often not of huge importance, as they are typically placed at regions of high energy away from the states and processes under investigation. Depending on the problem at hand, Legendre polynomials therefore often fare better in comparison to Chebyshev polynomials than expected from the considerations for spectral methods.

Also, the situation for the VES method is not as clear as in spectral methods or approximation theory in general: The function to be approximated by the set of basis functions is the bias potential that is not static but changes during the course of the optimization. To further complicate the matter, the input for the optimization is also not ideal or static data but obtained from sampling with the current version of the bias potential. This results in a feedback loop: During the optimization process of the bias potential, the basis functions approximate a temporary estimate of the bias potential. This estimate is then used during the next iteration to obtain sampling closer to the desired target distribution and then again update the estimate of the bias potential. The optimization process therefore relies on a good bias representation also during the intermediate steps.

This limits what can be theoretically deduced: Assuming a system for which the minimum of the functional and the associated optimal bias potential can be calculated (e.g., for a model system where the FES is known), it would be possible to determine the set of basis functions that offers the best representation. In practice, however, the basis functions are not only required to yield a good approximation of the converged potential, but also a good representation at the intermediate steps and a smooth evolution with changing coefficients. These intermediates cannot be easily calculated, due to the self-learning nature of the optimization and its dependency on sampling. Additionally, while a higher number of basis functions, for example of the Fourier series, is always better at representing a specific functional form of the bias potential, the increased number of optimization parameters might lead to worse convergence behavior, as observed in Ref. 159. It is also not clear how much different optimization algorithms and the associated parameters play a role. For example, when using the averaged stochastic gradient descent algorithm, the coefficients for all basis functions are updated with the same stepsize. This might not be optimal if the chosen basis set requires vastly different values of the coefficients for convergence.

In summary, it is difficult to tell a priori which basis set works best in general or even a specific problem at hand. Therefore for VES typically a few short simulations with different sets are run initially to see which shows the best performance, before starting the productive simulation runs.

Strang and Nguyen present a very strong motivation for using wavelets as basis functions: “For [approximating] piecewise smooth functions, a wavelet basis is better” [189, p. 229]. The FES to be approximated by the bias potential in the VES method are typically rather smooth. The aforementioned motivation makes a wavelet basis a good choice if the FES consists of features with different shapes and sizes.

5

THEORY OF DAUBECHIES WAVELETS

The view on Wavelets has evolved a lot since the 1980s when their theoretic foundation was first studied in a comprehensive way and previously existing ideas were collected and refined to form the field that is now called wavelet analysis. Many books and articles with different approaches to the theory exist. The book by Daubechies [186] is considered to be a comprehensive classic on the topic of wavelet analysis. There, she collected her work on orthonormal wavelets coming from the theory of "frames". Another approach is the concept of "filters", used for example by Kahane and Lemarié-Rieusset [190] or Strang and Nguyen [189]. In the following, a short introduction to wavelets will be given, following ideas of these theoretical approaches. Concepts from Daubechies' book will provide most of the fundamental aspects of wavelet analysis, while the more practical aspects such as algorithms for the numeric construction will come from the book of Strang and Nguyen.

Besides the already mentioned works, interested readers are referred to Refs. 191–193 for further studies.

5.1 WINDOWED FOURIER TRANSFORM AND WAVELET TRANSFORM

Many of the initial ideas of wavelet analysis come from the field of time-frequency analysis for signal processing. There, typically the frequencies of a signal $f(x)$ are analyzed via the Fourier transform

$$(\mathcal{F}f)(\omega) = \hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \mathrm{d}x\, \mathrm{e}^{-ix\omega} f(x). \tag{5.1}$$

Here the normalization was chosen such that the inverse transform also has the prefactor $(2\pi)^{-1/2}$:

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \mathrm{d}\omega \, \mathrm{e}^{ix\omega} \hat{f}(\omega). \tag{5.2}$$

The problem with this approach comes from the fact that most signals are evolving with time, where the concept of fixed frequencies applies only to specific points in time. Thus, the full signal can no longer be represented by a superposition of plane waves. Instead, the idea of wavelets introduces a locality in time: The signal is described by a superposition of short wave-like functions, the wavelets.

One critical aspect is the so-called "uncertainty principle" of signal processing: A wavelet cannot be both very short in time and represent only a small range of frequencies. In other words, a compromise between resolution in the frequency and time domain has to be made. The following will continue to speak of the "time" domain of the signal for an intuitive and short description, while this could also be replaced by, for example, a spatial domain or the CV space.

Before wavelet analysis, similar ideas have existed. Already in the 1940s, the windowed Fourier transform or Gabor transform was developed, which employs a time-shifted window to analyze the signal:

$$(\mathcal{G}f)(t,\omega) = \int_{-\infty}^{+\infty} \mathrm{d}x \, g(x-t) \, \mathrm{e}^{-ix\omega} f(x) \tag{5.3}$$

Often, a Gaussian function is used as the window $g(x-t)$ that is shifted over the signal to allow for a local inspection. To analyze signals numerically, the time shift and frequency modulation can be discretized, $t = nt_0$, $\omega = m\omega_0$ with fixed t_0, ω_0 and $n, m \in \mathbb{Z}$. Then eq. (5.3) becomes

$$\mathcal{G}_n^m(f) = \int_{-\infty}^{+\infty} \mathrm{d}x \, g(x - nt_0) \, \mathrm{e}^{-ixm\omega_0} f(x). \tag{5.4}$$

Given a signal, the coefficients for n, m can be calculated. For a fixed n the signal is localized around nt_0 and the local Fourier coefficients correspond to the frequencies $m\omega_0$ inside the windowed region.

Coming from the analysis of acoustic waves for sedimental analysis, in the early 1980s, Morlet proposed a new transform that uses dilation (i.e. scaling) instead of modulation (i.e. shifting in frequency) of the signal:

$$(\mathcal{T}f)(a,b) = |a|^{-1/2} \int_{-\infty}^{+\infty} \mathrm{d}x\, \psi\left(\frac{x-b}{a}\right) f(x) \tag{5.5}$$

This is the wavelet transform, where the signal is convoluted with wavelets

$$\psi_b^a(x) = |a|^{-1/2}\, \psi\left(\frac{x-b}{a}\right), \tag{5.6}$$

which are shifted and dilated versions of a function ψ, that is sometimes called "mother wavelet". The discrete variant is again obtained by restricting the parameters a, b to discrete values. With the choice $a = a_0^j$, $b = kb_0 a_0^j$, one obtains

$$\mathcal{T}_k^j(f) = |a_0|^{-j/2} \int_{-\infty}^{+\infty} \mathrm{d}x\, \psi(a_0^{-j} x - kb_0) f(x), \tag{5.7}$$

where a_0, b_0 are fixed and $j, k \in \mathbb{Z}$. Similarly to the windowed Fourier transform of eqs. (5.3, 5.4), the index k shifts a window of constant shape over the signal. Differently, the wavelet transform uses windows (in the form of wavelets) of different sizes. When the value of the scale parameter j is changed, the size of the wavelet and the corresponding frequencies also changes: large values of j correspond to large wavelets representing small frequencies, while small values of j correspond to small wavelets and large frequencies. The shift parameter k determines the localization centers for the wavlets at each scale. An individual wavelet is centered around $kb_0 a_0^j$.

The main difference between the windowed Fourier and wavelet transform is the shape of the analyzing functions $g_t^\omega = g(x-t)\mathrm{e}^{-ix\omega}$ and ψ_b^a. All functions g_t^ω consist of the same window function $g(x-t)$ shifted to a location t, where the frequencies are evaluated. In contrast, the wavelet functions ψ_b^a adapt their widths to the frequencies, that is low-frequency wavelets are wider than the narrow wavelets representing high frequencies. This means that the wavelet transform is better suited for signals with features of drastically different sizes or frequencies.

5.2 MULTIRESOLUTION ANALYSIS

Of special interest for this work are the wavelets proposed by Daubechies. She found that for some values of a_0 and b_0, there are wavelets that constitute orthonormal bases of $L^2(\mathbb{R})$ [26]. In the following, $\psi \in L^2(\mathbb{R})$ is supposed and only real-valued functions are assumed such that the argument of L^2 can be omitted. Orthonormal bases exist for at least all rational $a_0 > 1$, but most common and simplest is the choice $a_0 = 2$ and $b_0 = 1$ leading to

$$\psi_k^j(x) = 2^{-j/2}\psi\left(2^{-j}x - k\right). \tag{5.8}$$

Several conditions have to be satisfied to obtain an orthonormal basis. In general, for the wavelet transform to be invertible, the admissibility condition

$$\int_{-\infty}^{+\infty} \frac{\left|\hat{\psi}(\omega)\right|^2}{|\omega|}\,\mathrm{d}\omega < \infty \tag{5.9}$$

has to hold, where $\hat{\psi}$ is the Fourier transform of the wavelet [194]. It can be shown that condition (5.9) is equivalent to

$$\int_{-\infty}^{+\infty} \psi(x)\,\mathrm{d}x = 0 \tag{5.10}$$

for all practical purposes. To form orthonormal bases the wavelets have to be orthonormal across all scales and shifts

$$\int_{-\infty}^{+\infty} \psi_k^j(x)\,\psi_{k'}^{j'}(x)\;\mathrm{d}x = \delta_{jj'}\delta_{kk'}. \tag{5.11}$$

If the basis is also complete, any function $f(x) \in L^2$ can be represented.

The approximation procedure of a function $f(x)$ can be explained with a “multiresolution” approach: Starting at some scale j the function is approximated with a linear combination of the shifted ψ_k^j using some coefficients α_k^j:

$$f^j(x) = \sum_k \alpha_k^j \psi_k^j(x) \tag{5.12}$$

Going to the next "fine-grained" scale $j-1$, the approximation can be improved by adding more details at the finer scale

$$f^{j-1}(x) = f^{j}(x) + \sum_{k} \alpha_k^{j-1} \psi_k^{j-1}(x). \tag{5.13}$$

This procedure can then be repeated as often as desired to obtain approximations at progressively smaller scales.

The multiresolution property can be demonstrated further when looking at the spaces associated with the approximation levels. Introducing $\mathcal{V}_j$ as the space of the approximation f^j, it is clear that the space of the next smaller scale $\mathcal{V}_{j_1}$ will contain the previous one, which leads to a ladder of spaces:

$$\ldots \subset \mathcal{V}_{j+2} \subset \mathcal{V}_{j+1} \subset \mathcal{V}_{j} \subset \mathcal{V}_{j-1} \subset \mathcal{V}_{j-2} \ldots \tag{5.14}$$

These spaces have analog properties to the wavelet basis:

$$\bigcap_{j \in \mathbb{Z}} \mathcal{V}_j = \{0\} \tag{5.15a}$$

$$\overline{\bigcup_{j \in \mathbb{Z}} \mathcal{V}_j} = L^2 \tag{5.15b}$$

The first condition, eq. (5.15a), is the linear independence or separation property: The intersection of all spaces is only the zero set. Secondly, eq. (5.15b) requires that the closure of the spaces is the full function space L^2. In other words, the union of all spaces is dense in L^2, or even less mathematical, all L^2 functions are contained. This is the analogon to the completeness of the wavelet basis.

Having introduced the spaces $\mathcal{V}_j$, the multiresolution aspect comes from the fact that all these spaces are in fact scaled versions of a single space

$$f(x) \in \mathcal{V}_j \quad \longleftrightarrow \quad f(2^j x) \in \mathcal{V}_0. \tag{5.16}$$

This means, if a function is contained in the space $\mathcal{V}_0$, its scaled variant has to be contained in all other spaces $\mathcal{V}_j$. Additionally, all integer translates of contained functions have to also be contained, that is

$$f(x) \in \mathcal{V}_0 \quad \longrightarrow \quad f(x-k) \in \mathcal{V}_0 \quad \text{for all } k \in \mathbb{Z}. \tag{5.17}$$

Although a ladder of spaces with the correct properties to obtain orthonormal bases has now been constructed, for practical purposes a problem remains: The ladder of spaces is infinite in both directions. While omitting levels at the finer scales is excusable as it results only in deviations on the missing small scales, the large scales cannot be omitted without fundamentally breaking the approximation. This problem can be solved with the introduction of a scaling function or "father wavelet" ϕ (in contrast to the "mother wavelet" ψ), defined as the generating function of a set of functions constituting an orthonormal basis for $\mathcal{V}_j$:

$$\phi_k^j(x) = 2^{-j/2}\phi\left(2^{-j}x - k\right). \tag{5.18}$$

The single level of scaling functions now serves as the "most coarse-grained" approximation. This can be made clear by defining the space of the wavelet $\mathcal{W}_j$ as the orthogonal complement of the space $\mathcal{V}_j$

$$\mathcal{V}_j \cap \mathcal{W}_j = \{0\}. \tag{5.19}$$

Now the direct sum connects spaces at different scales

$$\mathcal{V}_j \oplus \mathcal{W}_j = \mathcal{V}_{j-1}, \tag{5.20}$$

which means that $\mathcal{V}_j$ contains all the higher level wavelet spaces, for example when starting at some scale $j = 0$

$$\mathcal{V}_0 = \mathcal{W}_1 \oplus \mathcal{V}_1 = \mathcal{W}_1 \oplus \mathcal{W}_2 \oplus \mathcal{V}_2 = \mathcal{W}_1 \oplus \mathcal{W}_2 \oplus \mathcal{W}_3 \oplus \mathcal{V}_3 = \dots . \tag{5.21}$$

The full multiresolution analysis is recovered by adding the smaller wavelet spaces

$$L^2 = \mathcal{V}_0 \oplus \mathcal{W}_0 \oplus \mathcal{W}_{-1} \oplus \mathcal{W}_{-2} \oplus \mathcal{W}_{-3} \oplus \dots \tag{5.22}$$

and any function $f \in L^2$ can be approximated by

$$f(x) = \sum_k \alpha_k \phi_k^j(x) + \sum_{l \geq j} \sum_k \alpha_{l,k} \psi_k^l(x). \tag{5.23}$$

Left to choose is the specific wavelet and scaling functions to be used.

5.3 FILTER COEFFICIENTS

Although different approaches exist, the wavelet and scaling functions are typically defined by filter coefficients, that is via the refinement relations

$$\phi(x) = \sqrt{2} \sum_{k=0}^{L} h_k \, \phi(2x - k), \tag{5.24a}$$

$$\psi(x) = \sqrt{2} \sum_{k=0}^{L} g_k \, \phi(2x - k). \tag{5.24b}$$

The length of the filter L has to be finite for compactly supported wavelets. To obtain wavelets that satisfy orthogonality relations [191]

$$\int_{-\infty}^{+\infty} \phi_k^j(x) \phi_{k'}^j(x) \, \mathrm{d}x = \delta_{kk'}, \tag{5.25a}$$

$$\int_{-\infty}^{+\infty} \phi_k^j(x) \psi_{k'}^{j'}(x) \, \mathrm{d}x = 0 \qquad \text{for } j \leq j', \tag{5.25b}$$

$$\int_{-\infty}^{+\infty} \psi_k^j(x) \psi_{k'}^{j'}(x) \, \mathrm{d}x = \delta_{jj'} \delta_{kk'}, \tag{5.25c}$$

a necessary condition for the coefficients g_k and h_k is their relation by

$$h_k = (-1)^k g_{L-k}. \tag{5.26}$$

This illustrates that only one of the two sets of filter coefficients has to be found and fully determines the wavelet. Here the coefficients of the scaling function h_k will be constructed.

Following the ideas of Daubechies, the required conditions for the coefficients will be given in the Fourier transformed space. The Fourier transform of eq. (5.24a) is

$$\hat{\phi}(\omega) = \frac{1}{\sqrt{2}} \sum_k h_k \, e^{-ik\omega/2} \, \hat{\phi}(\omega/2) \tag{5.27a}$$

$$= m_0(\omega/2) \, \hat{\phi}(\omega/2), \tag{5.27b}$$

where

$$m_0(\omega) = \frac{1}{\sqrt{2}} \sum_k h_k \, e^{-ik\omega}. \tag{5.28}$$

To construct orthonormal wavelets the following conditions have to hold [186, 195]:

$$m_0(0) = 1 \tag{5.29a}$$

$$|m_0(\omega)|^2 + |m_0(\omega + \pi)|^2 = 1 \tag{5.29b}$$

$$|m_0(\omega)|^2 \text{ has no zeros in the interval } \left[-\frac{\pi}{3}, \frac{\pi}{3}\right] \tag{5.29c}$$

Completeness of the multiresolution analysis requires condition (5.29a), which is equivalent to $\hat{\phi}(0) \neq 0$. The second condition, eq. (5.29b), is necessary for the orthonormality of the wavelets. Equation (5.29c) is sometimes also named "Cohen condition" [195]. It ensures L^2 convergence of the approximation, that is, completeness of the basis.

On top of these conditions for orthonormal wavelets, Daubechies suggested conditions for a family of compactly supported wavelets with the highest possible number of vanishing moments for their support width. A wavelet is said to have N vanishing moments if

$$\int \mathrm{d}x \, x^p \psi(x) = 0 \tag{5.30}$$

holds for all $p = 0, 1, \ldots, N-1$. All polynomials up to order $N-1$ are then orthogonal to the wavelet function and therefore not contained in the spaces $\mathcal{W}_j$. Because completeness of the multiresolution analysis is required, from eq. (5.22) follows that these are then in the space of the scaling function $\mathcal{V}_0$. This is crucial when representing functions by a wavelet multiresolution analysis via eq. (5.23): All functions consisting of polynomials of order up $N-1$ can be exactly represented by the first sum, that is a single level of scaling functions.

For the construction of the filter coefficients, two additional conditions on m_0 are imposed: First, to obtain compactly supported wavelets, the function m_0 has to be a trigonometric polynomial. Second, having N vanishing moments requires that m_0 has a zero of order N at $\omega = \pi$. This motivates an ansatz of the form

$$\left|m_0^{(N)}(\omega)\right|^2 = \left(\cos^2(\omega/2)\right)^N P_N\left(\sin^2(\omega/2)\right), \tag{5.31}$$

where the first term ensures the zero of order N at $\omega = \pi$ and the P_N is a polynomial in $\sin^2(\omega/2) =: y$. Condition (5.29b) can be rewritten for this polynomial as

$$(1-y)^N P_N(y) + y^N P_N(1-y) = 1. \tag{5.32}$$

Daubechies showed the general solution to this equation. The special solution that yields the wavelets with minimal support, that were named after her, is given by

$$P_N(y) = \sum_{k=0}^{N-1} \binom{N-1+k}{k} y^k. \tag{5.33}$$

This is an ordinary polynomial of order $N-1$ with the coefficient of y^k given by a binomial.

While this already uniquely defines the wavelet for any positive integer N, for practical applications the problem of obtaining the filter coefficients h_k from $|m_0|^2$ remains. The square root of $|m_0|^2$ can be extracted by spectral factorization.

Spectral factorization

Strang and Nguyen [189] presented a simple and efficient algorithm for the construction of the filter coefficients from the trigonometric polynomial. It relies on finding the zeros of a polynomial for the spectral factorization. A trigonometric polynomial $S(\omega)$ can be rewritten as

$$|S(\omega)|^2 = \left| \sum_{k=0}^{N} c_N \, \mathrm{e}^{-ik\omega} \right|^2 = \sum_{k=-N}^{N} d_N \, \mathrm{e}^{-ik\omega} = R(\omega), \tag{5.34}$$

where the coefficients $d_N = d_{-N}$ are symmetric. Introducing a complex variable $z = e^{i\omega}$ corresponding to the z-transform of x, this means that $R(z) = R(1/z)$. For any root z_i there is also a root $1/z_i$ of the polynomial. If the root z_i is inside the unit circle, $1/z_i$ is outside and reverse. Any roots exactly on the unit circle must have even multiplicity. Using these considerations, the polynomial $z^N R(z)$ with leading coefficient $d_N \neq 0$ and $N - M$ roots on the unit circle must have $2N$ factors of the form

$$z^N R(z) = d_N \prod_{i=1}^{M} (z - z_i) \left(z - \frac{1}{z_i} \right) \prod_{j=1}^{N-M} (z - z_j)^2 \tag{5.35}$$

For the problem at hand, the coefficients c_n correspond to h_k which are required to be real coefficients. Therefore, the roots not on the unit circle do not only come in pairs but actually four at a time: The complex conjugate $\bar{z}_i$ is also a root if z_i is a root, which leads to a quadruple of z_i, $\bar{z}_i$, $1/z_i$, and $1/\bar{z}_i$. All roots on the unit circle come in pairs, which satisfies the even multiplicity. To construct the polynomial $S(z)$, one now has to choose either the two roots within or outside the unit circle for every quadruple z_i, as well as one out of every pair of roots on the unit circle z_j. Assuming the choice z_i (and the corresponding $\bar{z}_i$) for all quadruples, this leads to

$$z^N S(z) = |d_N|^{1/2} \prod_{i=1}^{M} (z - z_i) \prod_{j=1}^{N-M} (z - z_j). \tag{5.36}$$

Although the roots z_i are complex, the coefficients c_N of the polynomial are still real because every root comes with its complex conjugate. The problem of obtaining $S(\omega)$ from $R(\omega) = |S(\omega)|^2$ has thus been reduced to finding the roots of $R(\omega)$. Due to the possibility of choosing the roots z_i or $1/z_i$ in the construction, it is clear that the resulting coefficients of $S(z)$ are not unique, but that there are in fact 2^M different possibilities. Additionally, $S(\omega)$ can always be multiplied with a factor $e^{in\omega}$ for any integer n.

Algorithm for the filter coefficients of Daubechies wavelets

Returning to the problem of extracting the root of $|m_0|^2$ and then obtaining the filter coefficients h_k via the fourier transform (see eq. (5.28)), an explicit algorithm can now be derived [189]. From $\sin^2(\omega/2) = y$ follows $\cos(\omega) = 1 - 2y$ and a change in the variable of $|m_0|^2$ leads to

$$\left|m_0^{(N)}(y)\right|^2 = (1-y)^N P_N(y), \tag{5.37}$$

with P_N defined as in eq. (5.33). To perform the spectral factorization, another transform into the z-domain is required. With

$$1 - y = \left(\frac{1+z}{2}\right)\left(\frac{1+z^{-1}}{2}\right), \tag{5.38a}$$

$$y = \left(\frac{1-z}{2}\right)\left(\frac{1-z^{-1}}{2}\right), \tag{5.38b}$$

the trigonometric polynomial for the Daubechies wavlets becomes

$$\left|m_0^{(N)}(z)\right|^2 = \left(\frac{1+z}{2}\right)^N \left(\frac{1+z^{-1}}{2}\right)^N \sum_{k=0}^{N-1} \binom{N-1+k}{k} \left(\frac{1-z}{2}\right)^k \left(\frac{1-z^{-1}}{2}\right)^k . \tag{5.39}$$

This polynomial has $2N$ zeros on the unit circle (at $z = -1$) from the first two terms. Half of these have to go into m_0 for the spectral factorization. The $2N - 2$ zeros of the sum come in pairs inside and outside the unit circle, just as described above. The problem is now analog to obtaining the coefficients of $S(z)$ from $R(z)$:

1. Determine the $N - 1$ complex roots of the polynomial $P_N(y)$
2. Calculate the $2N - 2$ corresponding roots in z via the quadratic formula $z^2 + z(4y - 2) + 1 = 0$

3. Choose one of each pair of roots, resulting in $N-1$ roots z_i
4. Add N roots at $z=-1$
5. Construct a polynomial from the $N+N-1$ roots: $\prod_i^{N-1}(z-z_i)\prod_j^N(z+1)$
6. Obtain the $2N$ filter coefficients h_k by expanding the polynomial: The filter coefficients are the coefficients of the polynomial

In the last step, the inverse z-transform from $m_0(z)$ to the h_k was done implicitly, but without taking the normalization into account. In the literature, different normalizations for the filter coefficients can be found, the one by Daubechies that was also given in eq. (5.28) is $\sum_k h_k = \sqrt{2}$.

Step 3 of the construction algorithm offers flexibility in the choice of the coefficients. In total $2^{\lfloor N/2 \rfloor}$ different sets of filter coefficients are possible. The one leading to the "Daubechies" wavelets is the one with extremal phase, which corresponds to selecting all roots inside the unit circle (or equivalently all outside). This results in wavelets with good regularity but pronounced asymmetry, with the highest absolute function values all in the first part. Two examples, the Db4 and Db8 wavelets, can be seen in the top row of Figure 5.1. Here the short form denotes the type of wavelet (Db for Daubechies) and the number of vanishing moments N.

Another popular goal is to obtain more symmetric wavelets, or in other words, wavelets that have a filter with (nearly) linear phase. In the field of signal analysis, linear phase filters are often desired because they preserve the "waveform" of the signal. For compactly supported and orthonormal wavelets, perfect symmetry or asymmetry can only be achieved for $N=1$. Nevertheless, better symmetry than for the Db wavelets is possible by choosing different roots in step 3. While a method for selecting the roots to obtain a filter with a near-linear phase with error up to $\mathcal{O}(1)$ is given in Ref. 190, this does not always result in the most linear choice. To obtain the "least asymmetric" wavelets, no a-priori criterium is known and one has to actually calculate the phase for all permutations and then select the one with the phase closest to linear. The resulting wavelet family is also called Symlets (denoted Sym$N$, with the number of vanishing moments N), although it is of note that these are not

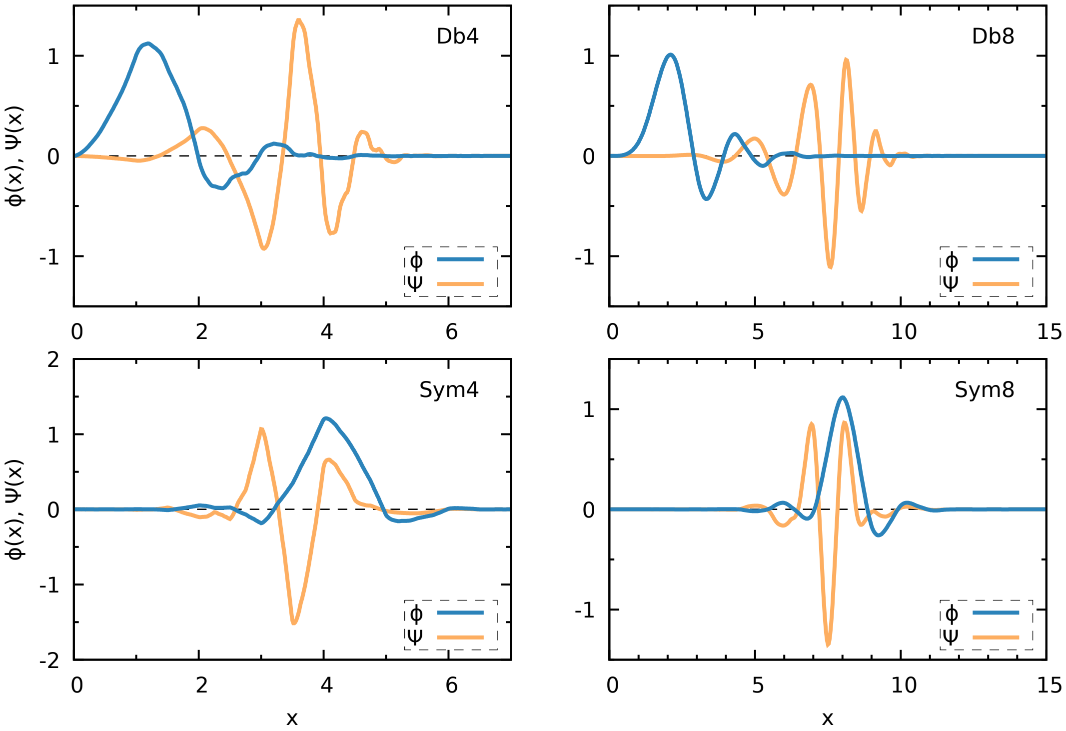


Figure 5.1: Functional form of the scaling function (blue) and mother wavelet (orange) for four exemplary choices of Daubechies filter coefficients. Top row: extremal phase with 4 (left) and 8 (right) vanishing moments. Bottom row: near-linear phase with the same vanishing moments. The plot labels give the short notation of the specific wavelet type. Depicted is exactly the supported range, the functions are zero by definition outside.

actually symmetric. Exemplary scaling and wavelet functions of the Sym4 and Sym8 filters are shown in the bottom row of Figure 5.1.

5.4 VECTOR CASCADE ALGORITHM

Although the filter coefficients h_k define the scaling and (mother) wavelet function uniquely, the task of obtaining their functional value at arbitrary points remains. No closed analytic form exists for these wavelets, so typically an iterative scheme is used for the generation on a grid. The most common algorithm is the cascade algorithm, in the following a vectorized form that is discussed in Ref. 189 is described. This vector cascade algorithm starts by introducing two special matrices $\boldsymbol{M}_0$ and $\boldsymbol{M}_1$ constructed from the filter coefficients. The values at the integers are given by an eigenvector of $\boldsymbol{M}_0$. An iterative scheme that involves multiplication with the matrices $\boldsymbol{M}_0$ and $\boldsymbol{M}_1$ is then used to generate values at intermediate points.

Starting point is the dilation or recursion relation for the scaling function that was previously given in eq. (5.24a):

$$\phi(x) = \sqrt{2} \sum_{k=0}^{L} h_k \, \phi(2x - k) \tag{5.40}$$

In the previous section, the number of coefficients for Daubechies wavelets has been determined to be $2N$ which are assumed to be given by $h_0, h_1, \ldots, h_{2N-1}$. Because there is only a finite number of non-zero coefficients, $\phi(x)$ can only be non-zero inside $[0, 2N-1)$.

Analyzing eq. (5.40), it is clear that the value at a specific point depends only on points at double the position and then shifted by an integer. The values at the integers therefore only depend on each other. Inserting these into the equation, that is $x = 0$, $x = 1$, …, $x = 2N-2$, leads to one equation per integer. Choosing exemplarily $N = 3$, this set of equations can be written explicitly in a vector-matrix form

$$\begin{bmatrix} \phi(0) \\ \phi(1) \\ \phi(2) \\ \phi(3) \\ \phi(4) \end{bmatrix} = \sqrt{2} \begin{bmatrix} h_0 & & & & \\ h_2 & h_1 & h_0 & & \\ h_4 & h_3 & h_2 & h_1 & h_0 \\ & h_5 & h_4 & h_3 & h_2 \\ & & & h_5 & h_4 \end{bmatrix} \begin{bmatrix} \phi(0) \\ \phi(1) \\ \phi(2) \\ \phi(3) \\ \phi(4) \end{bmatrix} = \boldsymbol{M}_0 \boldsymbol{\Phi}(0), \tag{5.41}$$

where the matrix $\boldsymbol{M}_0$ and the vector $\boldsymbol{\Phi}(t) = \{\phi(t), \phi(t+1), \ldots, \phi(t+2N-2)\}$ were introduced for a shorter notation. Also, the zeros at the empty positions of the matrix $\boldsymbol{M}_0$ where omitted. Looking closely at the dilation equation (5.40), it becomes clear that the matrix can be constructed for any N by

$$M_{0;i,j} = \begin{cases} \sqrt{2}\,h_{2i-j} & \text{for } 0 \leq 2i-j \leq 2N-1, \\ 0 & \text{otherwise,} \end{cases} \tag{5.42}$$

where the indices for the rows i and columns j of the matrix go from 0 to $2N-2$ for a simpler notation. The "dilation equation for the integers" (5.41) is an eigenvalue problem for the matrix $\boldsymbol{M}_0$ with the eigenvector $\boldsymbol{\Phi}(0)$. For a nontrivial solution, the eigenvalue $\lambda = 1$ must exist. The associated eigenvector then contains the values of $\phi(x)$ at the integers.

Assuming that these values were obtained, the construction can be continued with the intermediate points. Because the argument of ϕ only appears multiplied by two on the right side of eq. (5.40), the values of $\phi(x)$ at the half-integers also only depend on the values at the integers. Explicitly, when inserting the half-integer values $x = 1/2$, $x = 3/2$, …, $x = (N-1)/2$ into the dilation equation (5.40), a similar set of equation as for the integers is obtained. Writing the equations again explicitly in vector-matrix form for the case $N = 3$ yields

$$\begin{bmatrix} \phi(1/2) \\ \phi(3/2) \\ \phi(5/2) \\ \phi(7/2) \\ \phi(9/2) \end{bmatrix} = \sqrt{2} \begin{bmatrix} h_1 & h_0 & & & \\ h_3 & h_2 & h_1 & h_0 & \\ h_5 & h_4 & h_3 & h_2 & h_1 \\ & & h_5 & h_4 & h_3 \\ & & & & h_5 \end{bmatrix} \begin{bmatrix} \phi(0) \\ \phi(1) \\ \phi(2) \\ \phi(3) \\ \phi(4) \end{bmatrix} = \boldsymbol{M}_1 \boldsymbol{\Phi}(0). \tag{5.43}$$

A construction rule for the matrix $\boldsymbol{M}_1$ is given by

$$M_{1;i,j} = \begin{cases} \sqrt{2}\,h_{2i-j+1} & \text{for } 0 \leq 2i-j+1 \leq 2N-1, \\ 0 & \text{otherwise,} \end{cases} \tag{5.44}$$

where again the indices for the rows i and columns j of the matrix start at 0. The values at the half-integers are therefore obtained by a simple matrix multiplication:

$$\boldsymbol{\Phi}(1/2) = \boldsymbol{M}_1 \boldsymbol{\Phi}(0) \tag{5.45}$$

The matrix M_1 can be obtained from the matrix M_0 by shfting the columns to the right by 1 and then filling the first column with the odd coefficients. The following will show that only these two matrices are required to obtain all values.

Continuing in the same vein with the next intermediate values, for the values at the quarter integers the scheme is already clear: Only the values at the half-integers appear in the right side of the dilation equation (5.40). The coefficient matrix required for this construction is also already known: it is the matrix M_0. Similarly, the values at the three-quarter-integers also only depend on the values at half-integers but shifted by one compared to the values at the quarter-integers. The corresponding matrix multiplications are therefore given by

$$\mathbf{\Phi}(1/4) = M_0\mathbf{\Phi}(1/2) = M_0 M_1 \mathbf{\Phi}(0), \tag{5.46}$$

$$\mathbf{\Phi}(3/4) = M_1\mathbf{\Phi}(1/2) = M_1 M_1 \mathbf{\Phi}(0). \tag{5.47}$$

Continuing with this scheme, the values at all dyadic points, $\mathbf{\Phi}(n/2^j)$ with n going over all odd integers smaller 2^j, are obtained iteratively. The remaining question is which of the two matrices M has to be multiplied from the left to obtain the values at the next intermediate points: Each iteration (after the first) yields two new dyadic points, $t/2$ and $3t/2$, smaller and greater than the previous point t. A handwaving explanation is, that for the calculation of the values of the smaller points, the matrix M_0 has to be used, while for the values at the greater points multiplication with M_1 has to be done. Mathematically, the previous considerations yield the vector form of the dilation equation

$$\mathbf{\Phi}(t) = M_0\mathbf{\Phi}(2t) + M_1\mathbf{\Phi}(2t-1), \tag{5.48}$$

where $\mathbf{\Phi}(t)$ is zero outside of $0 \leq t < 1$. For $t = 0$, the eigenvalue problem of eq. (5.41) follows directly. Clearly for all points $0 < t \leq 0.5$ only the first summand is non-zero, while for $0.5 < t < 1$ only the second term contributes. This formally explains the point about the order of multiplications to obtain the value at a specific point. A better visualization of the order of matrix multiplications can be obtained by looking at the binary representation of t, that is the argument of the vector $\boldsymbol{\Phi}$ containing the desired point. Using the examples of eq. (5.46), the binary representations are $(1/4)_{10} = (0.01)_2$ and $(3/4)_{10} = (0.11)_2$. A binary 0 means the choice of the smaller dyadic point from the previous iteration, while a binary 1 is the larger dyadic point. Therefore, the digits can be associated with the required matrices: Each digit after the dot represents now a matrix multiplication, where 0 stands for $\boldsymbol{M}_0$ and 1 for $\boldsymbol{M}_1$. The vector recursion rule is therefore

$$\boldsymbol{\Phi}(0.t_1 t_2 t_3 \ldots)_2 = \boldsymbol{M}_{t_1} \boldsymbol{\Phi}(0.t_2 t_3 \ldots)_2, \tag{5.49}$$

where the t_i denote the digits of t in binary representation.

This concludes the introduction of the vector cascade algorithm to calculate the values of the scaling function. The values of the wavelet function can be obtained in a similar fashion. Nevertheless, a few details are still to be discussed: For the construction of the values at the integers, the eigenvector of $\boldsymbol{M}_0$ with the eigenvalue $\lambda = 1$ was assumed to exist. Without going into further details here, if condition (5.29b) holds for the filter coefficients, as it does for Daubechies wavelets, then $\boldsymbol{M}_0$ has a singular eigenvalue $\lambda = 1$. Furthermore, the corresponding eigenvector is not unique. Multiplication by a scalar factor still gives a solution to the eigenvalue problem. It can be shown that normalizing the integer values to

$$\sum_k \phi(k) = 1 \tag{5.50}$$

yields a scaling function with the normalization

$$\int \mathrm{d}x \, \phi(x) = 1 \tag{5.51}$$

that satisifies the orthonormality condition (5.25a).

Existence and construction of derivatives

For the usage in VES, at least the first derivative of the basis functions is required. While these could be obtained numerically, for example using Simpson's rule, this is rather slow from a computational standpoint because it requires evaluation of the function at several points. Instead, the derivatives of the scaling and mother wavelet functions can also be created iteratively on a grid. Starting point is once again the dilation equation (5.40). Taking the derivative of both sides of the equation with respect to x yields

$$\phi'(x) = 2\sqrt{2}\sum_{k}^{L} h_k\,\phi'(2x-k). \tag{5.52}$$

The chain rule gives a factor of 2 on the right side. Otherwise, only $\phi(x)$ was replaced by its derivative $\phi'(x)$. Equation (5.52) is therefore another dilation equation, just with the values of the coefficients doubled. To construct the derivative, the vector cascade algorithm can be used again. The vector dilation equation now reads

$$\mathbf{\Phi}'(t) = 2\boldsymbol{M}_0\mathbf{\Phi}'(2t) + 2\boldsymbol{M}_1\mathbf{\Phi}'(2t-1), \tag{5.53}$$

which shows that the same matrices $\boldsymbol{M}_0$ and $\boldsymbol{M}_1$ can be used for the cascade after all entries have been doubled. Only the starting point of the cascade is different: to obtain the values at the integers ($t = 0$) another eigenvalue problem

$$\mathbf{\Phi}'(0) = 2\boldsymbol{M}_0\mathbf{\Phi}'(0) \tag{5.54}$$

has to be solved. The eigenvector yielding the values at the integers $\mathbf{\Phi}'(0)$ is now the one corresponding to the eigenvalue $\lambda = 1/2$ of $\boldsymbol{M}_0$.

Although the derivatives of higher order are typically not required for the VES method, the general construction rule is now clear. The vector dilation equation for the n-th derivative is given by

$$\frac{\mathrm{d}^n\mathbf{\Phi}(t)}{\mathrm{d}x^n} = 2^n\boldsymbol{M}_0\frac{\mathrm{d}^n\mathbf{\Phi}(2t)}{\mathrm{d}x^n} + 2^n\boldsymbol{M}_1\frac{\mathrm{d}^n\mathbf{\Phi}(2t-1)}{\mathrm{d}x^n}, \tag{5.55}$$

with the values at the integers given by the eigenvalue problem

$$\frac{\mathrm{d}^n\mathbf{\Phi}(0)}{\mathrm{d}x^n} = 2^n\boldsymbol{M}_0\frac{\mathrm{d}^n\mathbf{\Phi}(0)}{\mathrm{d}x^n}. \tag{5.56}$$

The derivative can only be obtained if $\lambda = (1/2)^n$ is indeed an eigenvalue of $\boldsymbol{M}_0$. In Ref. 189 the required condition for the existence of the dyadic eigenvalues was formulated in the frequency domain of the filters: If $m_0(\omega)$ has a zero of order p at $\omega = \pi$, then the eigenvalues $\lambda = (1/2)^n$ with $n < p$ of $\boldsymbol{M}_0$ exist. For the Daubechies wavelets, the order of the zero at $\omega = \pi$ is equal to the number of vanishing moments N of the wavelet. Therefore, for the Daubechies wavelet with N vanishing moments, the first $N - 1$ derivatives can be obtained by the presented method.

Of note is that these derivatives are not necessarily continuous. The construction via the eigenvalue problem yields only the derivative from the right at each point, which does not necessarily coincide with the derivative from the left. Conditions for the existence of continuous derivatives are given in the next section, 5.5.

The vector cascade algorithm was implemented in a standalone python module that also generates Daubechies filter coefficients. Its source code can be found in Appendix A.2

5.5 PROPERTIES OF DAUBECHIES WAVELETS

Support length

The domain of support or "length" of the wavelet and scaling functions depends on the number of non-zero coefficients g_k and h_k, respectively. For Daubechies wavelets, the number of coefficients for both is given by $2N$, with the number of vanishing moments N. The functions are then supported on $[0, 2N - 1)$ and zero outside. In Figure 5.1, wavelet and scaling functions for two different values of N were shown, where the larger domain of support for larger N is clearly visible.

Because the basis consists of integer-shifted functions, this means that a higher value of N results in more overlapping basis functions at each point. To cover a specific region at a fixed scale, a larger N therefore also requires more basis functions. For the representation of the bias in VES, larger N allows for better representation of features at the cost of more variational parameters to optimize. Therefore, the choice of N has to be weighed carefully for the application under investigation.

Smoothness

The smoothness or regularity of a function is measured by differentiability classes. A function is said to be in C^n if it has n continuous derivatives. For wavelets, the theoretic maximum that can be achieved is equivalent to the number of vanishing moments $N - 1$. The Daubechies wavelets are significantly less smooth, an asymptotic formula for large N is given by $\phi^{(N)}, \psi^{(N)} \in C^{cN}$ with $c \approx 0.2$. For small N, the asymptotic formula does not hold, but explicit values have been derived. The first derivative is continuous for $N \geq 2$, and the second for $N \geq 5$.

To employ Daubechies wavelets as basis functions in VES, $N \geq 2$ is therefore required. In practice, only wavelets with significantly more vanishing moments are used, so this limit is mostly of theoretical nature.

6

LOCALIZED BASIS FUNCTIONS FOR VES

In the last chapter, the fundamental theory of Daubechies wavelets was presented and a method for their construction on a grid was introduced. Here, their usage as basis functions for the VES method will be described. Besides the Daubechies wavelets, two additional sets of localized basis functions are introduced in Sections 6.2 and 6.3, Gaussians and cubic B-splines, respectively. Afterward, Section 6.4 provides technical aspects of the implementation of the basis functions into the VES code. In Section 6.5, first small studies on the optimal choice of the basis function parameters are be presented, before more comprehensive simulations that benchmark the different basis sets are given in the next chapter.

6.1 DAUBECHIES WAVELETS

For the usage of Daubechies wavelets as basis functions in the VES method, several choices have to be made. When looking at eq. (5.23), three choices become apparent directly: (1) the number of approximation levels l, (2) the exact type of the wavelets $\phi(x), \psi(x)$, and (3) the scale of the father wavelet j.

Here, the focus will be on the coarsest approximation only, which corresponds to representing the bias with a single level of father wavelets at some scale j

$$V(\boldsymbol{s}, \boldsymbol{\alpha}) = \sum_k \alpha_k \phi_k^j(\boldsymbol{s}), \tag{6.1}$$

with the shift parameter $k \in \mathbb{Z}$. Therefore, when the term wavelet is used in the following, it always refers to the father wavelets.

The wavelet type is determined by the set of filter coefficients h_k, as introduced in eq. (5.24). Desirable properties for the employment as VES basis functions are small support of the individual functions, at least C^1 regularity (one continuous derivative), and the reproduction of polynomials up to a desired order.

The wavelets developed by Daubechies and presented in Section 5.3 satisfy these properties and in fact result in the minimally supported functions for a given polynomial order. Here, the "least asymmetric" variant of these wavelets or so-called symlets [26] will be considered. In practice, the symlets were found to perform better than the "maximum phase" Daubechies wavelet type. Their smaller asymmetry comes at the cost of slightly reduced regularity compared to the "maximum phase" Daubechies wavelets. As only one continuous derivative is required for VES basis functions, that does not cause problems. The specific symlet type will be denoted by Sym$N$, where N is equal to the number of vanishing moments, introduced in eq. (5.30). Recalling from Section 5.5, having N vanishing moments means that all polynomial functions up to order $N-1$ are orthogonal to the mother wavelet. Consequently, any polynomial of order up to $N-1$ can be represented exactly by a single level of the father wavelet ϕ (i.e., the scaling function). Employing a wavelet basis with a larger N can thus help to construct a bias potential with less regularity and steeper slopes. On the other hand, the range over which the wavelet functions are non-zero is proportional to $2N-1$. A larger support results in more overlap between functions since the basis consists of functions shifted by a fixed amount depending on the scale. Therefore, larger N requires more basis functions to cover a specific region at a fixed scale j. A study of the optimal number of vanishing moments for an exemplary test system will be given in Section 6.5.2.

The scale j of the wavelet basis can be chosen freely. Instead of determining the scale directly, in the following the desired number of basis functions N_{bf} will typically be selected. In principle, the sum of eq. (6.1) goes over all integers and there is an infinite number of shifted wavelet functions in the basis. However, only a few of them are supported inside the range $[a, b]$ on which the bias potential is expanded.

To determine the scale j of the wavelet, the number of integer shifted wavelets that is at least partially supported inside he bias range has to be calculated as a function of j. Setting $n := 2^{-j}$ in eq. (5.18) yields

$$\phi_k^n(x) = \sqrt{2}n\,\phi(nx - k). \tag{6.2}$$

For a given n, the smallest possible number of wavelet basis functions N_{bf} contained within the bias range (by setting the start of the first function at the left edge and the endpoint of the last function at the right edge) is then determined to be

$$N_{\mathrm{bf}} = l + n(b - a) - 2, \tag{6.3}$$

where l is the intrinsic support length of the wavelets (here typically $l = 2N - 1$, with N being the number of vanishing moments). The first term accounts for all shifts due to the length of the functions, the second term adds the functions to be considered due to the length of the bias interval (in the scale of the wavelets), and 2 functions can be subtracted at the edges. Setting a fixed number of basis functions N_{bf}, the smallest scale that has only this many wavelet functions in the bias range is obtained by solving eq. (6.3) for n and then taking the logarithm, which yields

$$j = -\log_2(n) = -\log_2\left(\frac{N_{\mathrm{bf}} - l + 2}{b - a}\right). \tag{6.4}$$

As an example, consider expanding the bias in the CV range $[-3, 3]$ with 21 wavelet functions of the Sym8 type. Then $n = 4/3$ and $j = -\log_2(4/3) \approx -0.42$. The chosen approach does not result in integer values for j as introduced for the wavelet transform in eq. (5.7), but this is not a requirement for the application in the presented case.

Each wavelet function has a support length of l/n in CV space. In the given example, this would result in each function spanning 11.25 of the CV space, which is larger than the length of the expanded bias. Because the wavelet functions have values significantly different from zero only on a small part of their domain, as it can be seen in Figure 5.1, the scale is typically chosen smaller.

Only the ϕ_k^j that are at least partially supported and have major contributions inside the bias range are selected for the linear expansion of the VES bias. By omitting the ones at the edges that have only small values within the bias range, the number of basis functions can be reduced while only introducing minor deviation from the full basis. The amount of contribution of a basis function is here measured relative to the maximum value: Only the ϕ_k^j with any (absolute) function value inside the bias range that is at least 1 % of the maximum value are included for the expansion of the bias. To exclude these "tail wavelets" from the basis set, the required scaling j is calculated by replacing l with the "relevant" length of the wavelet in eq. (6.4) to arrive at the desired number of basis functions. In the simulations, there were no observed disadvantages from excluding wavelets with minor contributions, while it allows reducing the number of expansion coefficients to optimize. In the given example with 21 Sym8 basis functions in the range $[-3, 3]$, this changes the scale to $j \approx -1.5$ and each basis function now spans 5.32 in CV space, which is less than half compared to before. In Section 6.5.2, the influence of the number of basis functions on the resulting bias approximation is investigated for a small test system. The top left panel of Figure 6.1 exemplarily shows two adjacent basis functions for a Sym8 basis.

It is also possible to use the wavelet basis to approximate a bias potential on a periodic CV. The scaling is then chosen such that there is exactly one basis function that would start at each the left and right border of the CV space. These two then coincide for a periodic CV and only one of them is included in the basis set.

Implementation in the VES code

The function values, as well as their derivatives, are constructed on a grid via the vector cascade algorithm of Section 5.4. The size of the grid can be specified with the input, for the present purpose a rather large grid of at least 1,000 points is used. Because the algorithm iteratively creates grid points at the dyadic points between the integers, the actual size of the grid depends on the support of the chosen wavelet. For example, when using the Sym6 wavelets that

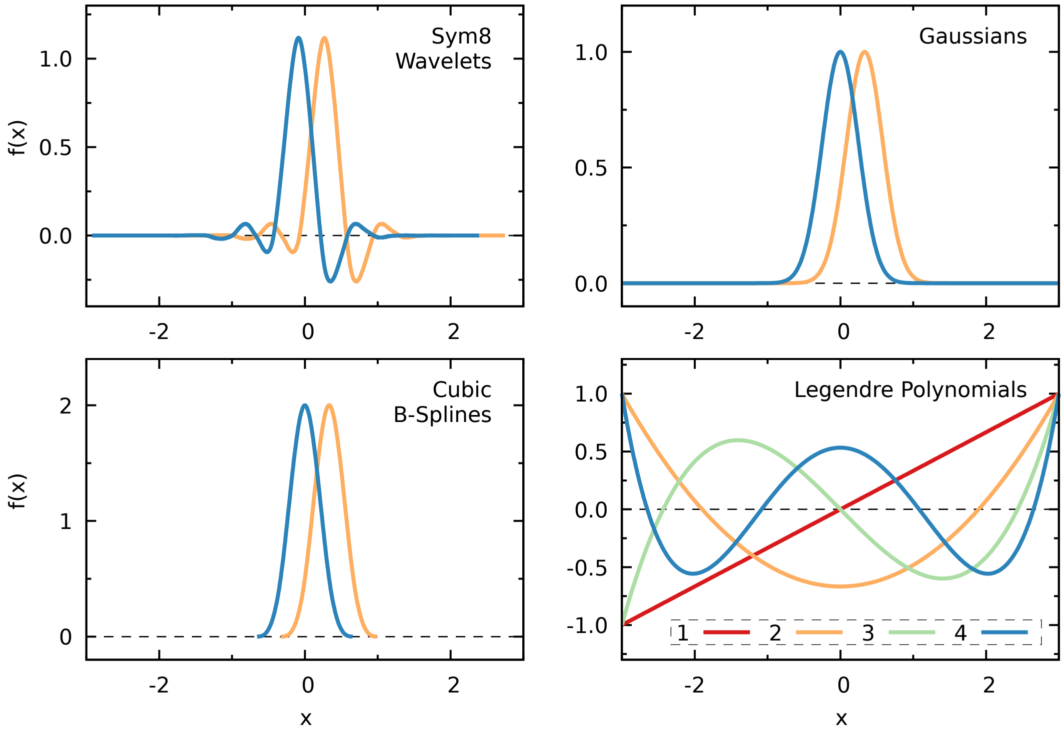


Figure 6.1: Visualization of different basis functions to expand a VES bias potential in the range $x \in [-3, 3]$. The Sym8 wavelets, Gaussians, and cubic B-splines are localized basis functions. For these, only two adjacent functions are shown exemplarily, while the full basis set includes all shifted functions in the interval with the same spacing. Each shown localized basis consists of 22 functions and the width of the Gaussians is set to $\sigma = 3d/4$. For comparison, the bottom right panel shows Legendre polynomials as an example for delocalized basis functions. The Legendre basis set consists of all polynomials up to a certain order, the figure shows the functions up to the quartic polynomial.

are supported on $[0, 11)$ this approach results in finding the smallest integer n for which $11 * 2^n \geq 1000$. The smallest $n \in \mathbb{Z}$ for which this inequality holds is 7 and results in a grid with 1408 points. Between these grid values, linear interpolation is used to obtain values at all desired points.

The implementation offers two options to specify the scaling of the basis functions. Beside the already mentioned specification of the number of basis functions, there is also the possibility to give the support length of the basis functions. The scaling of the functions is then directly defined and the code includes all basis functions with significant contributions inside the bias range, just as in the other case (see eq. (6.3) for the calculation of the resulting number of basis functions). Additionally, in both cases a constant basis function is included in the basis set.

6.2 GAUSSIANS

Gaussian basis functions are given by the mathematical expression

$$f_i(x) = \exp\left(-\frac{(x-\mu_i)^2}{2\sigma^2}\right), \tag{6.5}$$

where μ_i is the center of the individual Gaussian and σ is a width parameter. In the present work, the width parameter is kept constant for all basis functions. The full basis set is then given by Gaussians functions with centers distributed evenly on the interval $[a, b]$. The first center is added at $\mu_0 = a$ and the shift between centers is defined as $d = \mu_i - \mu_{i-1} = (b-a)/N$, where N is a user-specified integer fixing the number basis functions.

To mitigate systematic errors at the boundaries, one additional function with center outside the range is included on each side, resulting in a total of $N + 4$ basis functions including a constant basis function. As the force from the VES bias is zero outside the chosen interval by design, these additional functions will only contribute inside the bias range, similarly to the boundary correction approach for Metadynamics that was proposed by Baftizadeh et al. [118]. Although more complicated boundary correction algorithms have been developed [196, 197], this simple approach has been found to work well for the VES method.

To employ Gaussians for a periodic CV, the basis functions centered on the right and left border have to coincide. The functions centered outside the CV range, as well as one function centered on the border, are then excluded from the set, resulting in only $N + 1$ basis functions.

Besides the number of basis functions, the width σ of the Gaussians is also a parameter that can be chosen freely. Keeping the distances between the

Gaussians fixed, a larger value of σ results in more overlap between the basis functions. In Section 6.5.3 the influence of this parameter is investigated for a test system. The top right panel of Figure 6.1 visualizes the Gaussian basis set by showing two adjacent functions. For this example, the width parameter is set to $\sigma = 3d/4$.

6.3 CUBIC B-SPLINES

As the last localized basis function to be considered here, the cubic B-spline basis functions from Ref. 185 are chosen. They are given by the mathematical expression

$$f_i(x) = h\left(\frac{x - \mu_i}{\sigma}\right), \tag{6.6}$$

where

$$h(t) = \begin{cases} (2 - |t|)^3, & 1 \leq |t| \leq 2, \\ 4 - 6|t|^2 + 3|t|^3, & |t| \leq 1, \\ 0, & \text{elsewhere.} \end{cases} \tag{6.7}$$

The parameter μ_i is the center of the cubic B-spline basis function and σ sets the width of the splines (for $\mu_i = 0$ they are non-zero in the range $x \in [-2\sigma, 2\sigma]$). The full basis set is then given by spline functions with centers distributed evenly on the interval $[a, b]$. The first center is set on the left boundary $\mu_0 = a$ and the shift between between the centers is defined as $d = \mu_i - \mu_{i-1} = (b - a)/N$, where N is a user-specified integer fixing the number basis functions. Similar to the Gaussian basis functions, one function on each side outside the range is added to avoid boundary effects. This results in a total of $N + 4$ basis functions including a constant function.

For a periodic bias, the same strategy as for the Gaussian basis is chosen: The basis functions centered on the right and left border coincide and the ones outside the CV range are excluded. This results in $N + 1$ basis functions in the periodic range.

Differently from the Gaussian basis, the width parameter σ of the spline basis function is fixed and has to be equal to the distance between centers, $\sigma = d$. Therefore, the overlap between adjacent splines is fixed and the only parameter to be chosen is the number of splines to represent the bias. The bottom left panel of Figure 6.1 shows two adjacent splines of an exemplary basis with 22 functions in the range $[-3, 3]$.

6.4 IMPLEMENTATION INTO THE VES CODE

The new basis functions were implemented into the VES code. The VES code is a module of the PLUMED2 software [38, 39], which is written in C++ and publicly available under the LGPL license. The contributions discussed in this chapter have been made available for general use. They can be found in release version 2.8 and newer of the code, which is available in the GitHub repository (`https://github.com/plumed/plumed2`).

While it is straightforward to implement Gaussians and splines, wavelets pose the problem of not having an analytic mathematical expression. Instead, at the beginning of the simulation, the values and derivatives of the wavelet are generated on a grid through the vector cascade algorithm that was presented in Section 5.4. The grid is then used as a lookup table during the simulation. This means that the computational overhead for using a wavelet basis is minimal.

Modification of optimization scheme

As localized functions are non-zero only in a small region of the total CV space, the existing optimization schemes might behave suboptimally under some circumstances. The reason for this can be understood when looking at the gradient of the functional that was given in eq. (3.34a). It has two terms, of which the first is obtained from sampling and the second via averaging over the target distribution. If no sampling has occurred in the region of a basis function, the first term will be zero. Still, the gradient element for the basis function might be non-zero due to the average over the target distribution in the second term. This can actually be desired: the resulting positive gradient transports the message that the observed sampling (weighted with the individual basis function) is less than what is wanted by the target distribution. The optimization scheme will then try to increase the sampling in the region represented by the basis function. For localized basis functions, this means that the bias will become lower or even negative in the regions that are not visited between iterations of the bias potential. This might lead to a situation similar to the one in the non-well-tempered metadynamics method: the bias is never becoming static but continues to gradually change by a constant over the whole range. While this is not critical for running a biased simulation, it limits the predictions that can be drawn directly from the bias.

This can be circumvented by increasing the time between bias iterations such that all regions are sufficiently visited, at the cost of longer times until the bias has converged. Also, the time required for sampling all regions is changing during the simulation: In the initial stages the bias is still building up and transitions between the regions will likely be slower than in the later parts of the simulation.

Therefore an alternative approach is taken here that modifies the optimization scheme slightly: If there is no sampling in the region of a basis function during one iteration of the bias potential, the elements of gradient and Hessian corresponding to that basis function are set to zero before updating the variational parameters. Setting them to zero prevents erroneous updates of variational parameters if no sampling occurred. Note that this procedure is done only for individual elements so the total gradient vector and Hessian matrix still include non-zero elements.

In practice, it is not always clear what the "region" of a basis function is. For example, the Gaussians are theoretically non-zero everywhere, although the values far from the midpoint are vanishing small. Therefore, the criterion for turning off the optimization of a coefficient is implemented by setting a threshold value for the sampling: The elements of the gradient and Hessian are set to zero if the average from sampling (the first term in eq. (3.34a)) is smaller than a user-specified value. For all simulations presented in the following, this threshold is set to 10^{-6}.

6.5 DETERMINATION OF OPTIMAL PARAMETERS

Before benchmarking studies between the different basis sets are performed, the influence of the parameters of the individual basis set shall be investigated separately. For all three localized basis sets, this includes the number of basis functions. The wavelet basis can also be used with wavelets of different types. Since initial tests showed good results for the Symlet family, the following will test only the different members of this family, distinguished by their number of vanishing moments N. The Gaussian basis additionally has a width parameter σ that can be varied.

6.5.1 *Test system: Langevin dynamics in* 1D *double-well potential*

To be able to run many simulations with different parameters, a small toy system is employed. The VES code includes the `ves_md_linearexpansion` command line tool, which is an implementation of the Bussi-Parinello integrator for Langevin dynamics [48]. The integrator was presented in Section 2.2 and the exact steps of the algorithm are given by eq. (2.20). The `ves_md_linearexpansion` allows specifying an artificial potential in which the movement of a single particle is simulated.

Here, a simple one-dimensional potential of the form

$$U(x) = x^4 - 4x^2 + 0.7x \tag{6.8}$$

is considered. It has two states separated by a barrier from left to right of around 5 energy units. The form of this potential can be seen in Figure 6.2. The x-coordinate is taken as the CV such that the reference FES will be given by the potential above, $F(x) = U(x)$ (up to an additive constant). Natural units are used for all values and parameters. For better comparison, the parameters of the Langevin dynamics are set to the same values for all following simulations: The temperature is set to $T = 0.5/k_\mathrm{B}$ such that the barrier height from left to right is about 10 $k_\mathrm{B}T$. Also, a time step of $\delta t = 0.005$ and a friction coefficient of $\gamma = 10$ is chosen. Each simulation is run for 5×10^6 steps. To minimize the

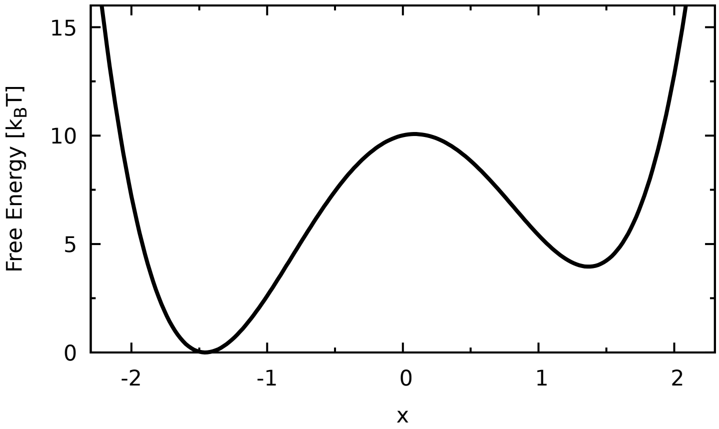


Figure 6.2: The free energy surface corresponding to the double-well potential given by eq. (6.8) at temperature $T = 0.5/k_\mathrm{B}$.

influence of the initial conditions, 20 independent simulations are run for each basis set. The simulations are started in the global minimum of the potential (at $x = -1.41$) and are given different random seeds for the initial velocities and random forces.

Left to set are the parameters of the VES method. The bias is expanded in the range $x \in [-3, 3]$ with the coefficients α of the bias potential being updated every 500 time steps. A uniform target distribution is employed. For the optimization, the averaged stochastic gradient descent optimization algorithm of eq. (3.29) is used. Its stepsize parameter is set to $\mu = 0.5$.

To analyze the performance of the basis set, the RMS error of eq. (3.49) is chosen. For each simulation, the FES is determined directly from the bias every 5×10^4 steps via eq. 3.21. For each FES estimate, the RMS error ϵ is calculated, where the reference FES is obtained directly from the potential. The time evolution of the average value $\langle \epsilon \rangle$ and the standard error $\hat{\sigma}_\epsilon$ over all 20 simulations of the basis set are then used for comparison with the other sets.

While some insight can already be gained by plotting the time evolution of the average RMS error, two points will be investigated explicitly: (1) The time required for the average error to become lower than a certain threshold value, and (2) the value of the average error at the end of the simulation. The first gives an insight into how fast the FES converges to a good approximation of the reference. The threshold value will be set individually to match the data at hand. The idea is to set it that the bias has mostly converged after reaching the threshold and only minor changes are observed afterward. The second point, in form of the final value of the average error, assesses how good the correct FES can be represented by the chosen set. To flatten possible fluctuations even at long times, an average over the last 10 available times is chosen.

6.5.2 *Daubechies wavelets basis functions: vanishing moments and number of basis functions*

As discussed in Section 6.1, two parameters can currently be adjusted when using a wavelet basis in VES. The first parameter of the basis is the type of wavelet functions to be used. Focusing on the Symlet family, the parameter to tune is the number of vanishing moments N. Father Wavelets with N vanishing moments can locally represent polynomials up to order $N-1$ well, so steeper slopes require wavelets with more vanishing moments. With increasing vanishing moments also the regularity of the wavelet increases, at the cost of a larger range at which they are non-zero if used at the same scale. When instead keeping the number of basis functions fixed, a larger N results in a larger scale. These properties can be seen in Figure 6.3, where bases constructed by different Symlets are given.

The performance of different Symlet types (Sym4 to Sym12) was tested for the one-dimensional double-well potential using the setup explained in Section 6.5.1. As threshold for the error, $\epsilon = 0.3\ k_\mathrm{B}T$ is chosen. The results are shown in Figure 6.4.

With an increasing number of vanishing moments, the individual basis functions overlap more and are thus able to represent features of higher polynomial order. While Sym4 wavelets result in only a very coarse approximation that fails to reach the area of $\epsilon < 0.3\ k_\mathrm{B}T$, starting at Sym6 there are only minimal differences to be seen. The standard deviation of the error also does not show substantial differences between the different basis functions. Concludingly, there is a minimum number of vanishing moments N to properly represent the un-

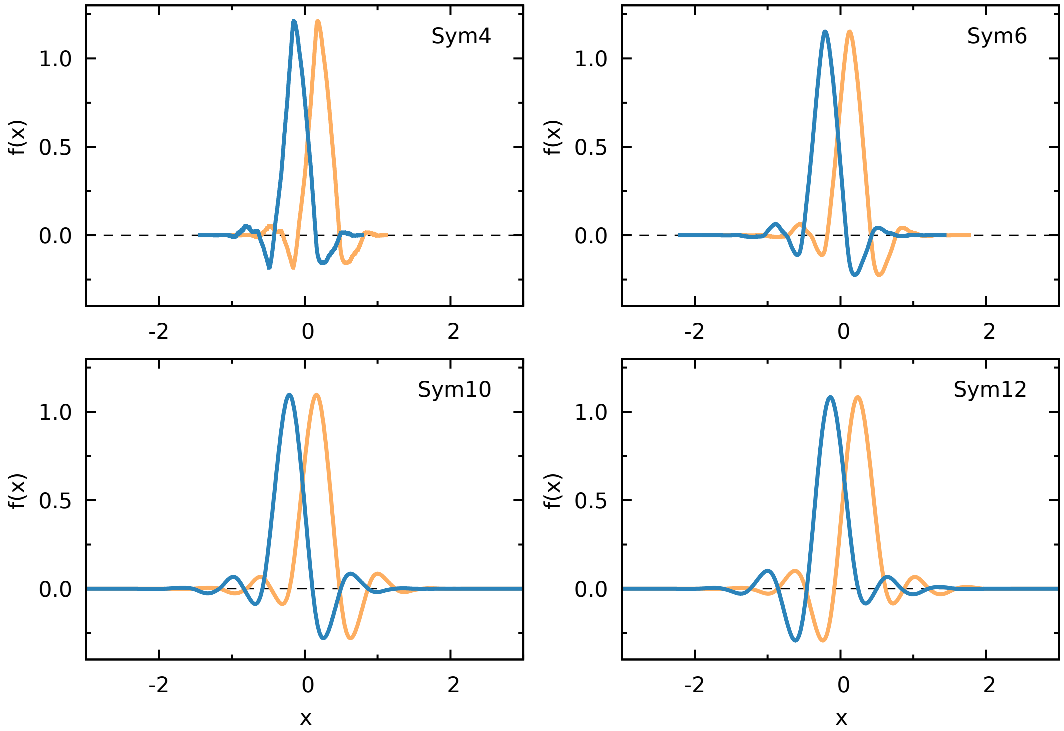


Figure 6.3: Visualization of different types of Symlet bases expanding a VES bias in the range $x \in [-3, 3]$. The Sym8 wavelets can be seen in Figure 6.1. Each basis consists of 22 shifted functions of the same type. Per basis set only two adjacent functions are shown, while the complete basis includes all shifted wavelets in the given range with the same spacing. Visible is the increasing length on which the individual wavelet functions are defined, from only slightly over 2 units to more than the full CV range of 6 units. This results in more overlapping functions when using wavelets with more vanishing moments, which makes more "irregular" features, i.e. features of higher polynomial order, representable. Also, the regularity of the wavelet increases from rougher spikes at Sym4 to smooth wave-like "oscillations" at Sym12.

derlying free energy surface of the system. This minimum number of vanishing moments depends on the shape of the free energy and can therefore vary. Based on these results, Sym8 wavelets were chosen for the benchmark simulations in the paper.

Secondly, the scaling of the individual basis functions can be chosen. This is equivalent to choosing the number of basis functions in the range of the bias. Using more basis functions allows representing finer features better. Therefore, simulations with different numbers of Sym8 basis functions were conducted

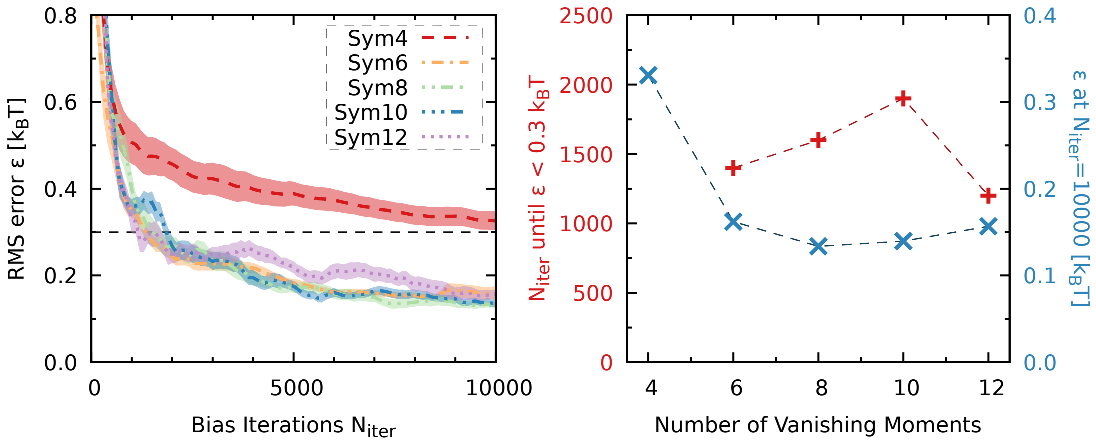


Figure 6.4: Study of different numbers of vanishing moments for the wavelet basis sets. *Left panel*: Average and standard error of the RMS error ϵ for different wavelet types as a function of the bias iterations. For each line, 20 independent simulations with 22 basis functions of a specific wavelet type were run, the form of the different functions can be seen in Figure 6.3. The shaded areas show the standard error and the black dashed line denotes the threshold value of $\epsilon = 0.3\ k_\mathrm{B}T$.
Right panel: Required number of bias iterations until the threshold value is reached, together with the final error at the end of the simulation for the different basis sets. The number of iterations is missing for Sym4 wavelets because the required accuracy was not reached.

on the double-well potential. Again, the protocol of Section 6.5.1 is used. The resulting time evolution of the root mean square errors are given in Figure 6.5, together with the time until the threshold error of $\epsilon < 0.3\ k_\mathrm{B}T$ is reached and the error at the end of the simulation.

The results show that there is a minimum number of basis functions required to represent the bias adequately. Using only 14 basis functions did not result in a good approximation of the FES, while good results were obtained when increasing the number of basis functions to around 20. A good representation is possible starting at 18 basis functions but even better with 22. Adding more basis functions does not improve the situation further and the final RMS errors differ not significantly. It is of note that using more basis functions leads to a higher number of coefficients to be optimized. The introduction of more optimization parameters complicates the optimization and slows convergence. Adding more basis functions after the basis can already adequately represent

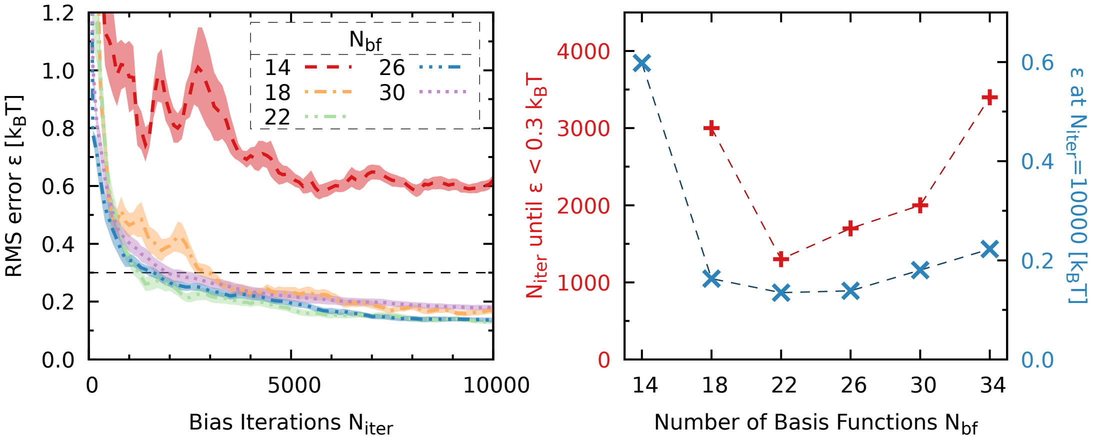


Figure 6.5: Study of different numbers of basis functions for the Sym8 basis set.
Left panel: Average and standard error of the RMS error ϵ for different numbers of Sym8 wavelet basis functions as a function of the bias iterations. For each line, 20 independent simulations with the specified number of basis functions were run. The shaded areas show the standard error and the black dashed line denotes the threshold value of $\epsilon = 0.3\ k_\mathrm{B}T$.
Right panel: Required number of bias iterations until the threshold value is reached, together with the final error at the end of the simulation for the different basis sets. The number of iterations is missing for 14 basis functions because the required accuracy was not reached.

the features of the system might thus lead to worse performance. This is visible when looking at the number of required iterations until the threshold error is reached: Increasing the number of basis functions above 22 yields slower convergence of the bias. In general, though, using too many basis functions is better than using too few.

6.5.3 *Gaussian basis functions: width parameter and number of basis functions*

As discussed in Section 6.2, in principle, the width parameter σ of Gaussian functions, that was defined in eq. (6.5), can be adjusted freely. In Ref. 19, where Gaussian basis functions had been previously employed for VES, $\sigma = d$ was chosen, with d being the distance between centers of adjacent Gaussians. To evaluate the influence of different choices, simulations with the protocol described in Section 6.5.1 are performed. Only basis sets with the same width parameter for all Gaussian functions are considered. The number of basis functions is set to 22, which fixes the centers of the Gaussians, and the width parameter is varied in the range $\sigma \in [0.5d, d]$. A width of $\sigma = 0.5d$ latter results in an only slightly larger overlap than the full width at half maximum of the Gaussians. The obtained errors of the FES approximations can be seen in Figure 6.6, together with the time until a threshold error of $\epsilon < 0.5\ k_\mathrm{B}T$ is reached and the error at the end of the simulation.

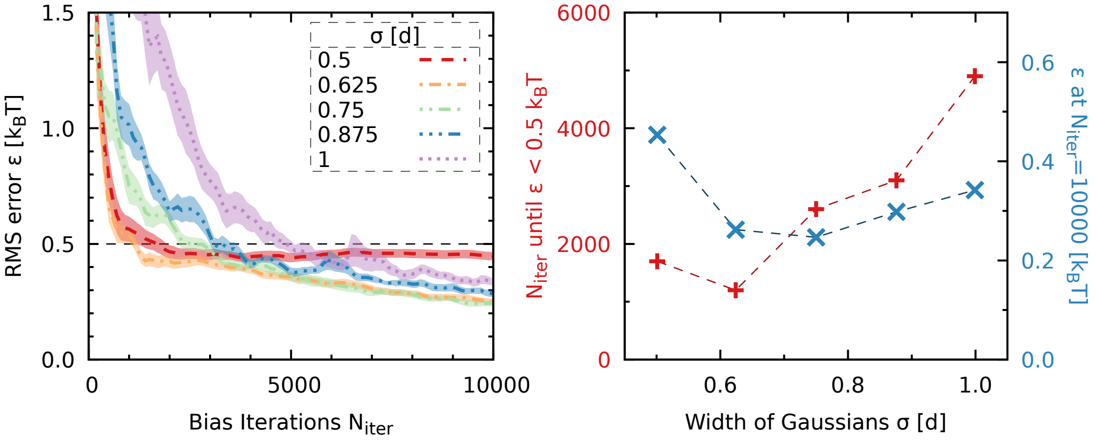


Figure 6.6: Study of the width parameter σ of the Gaussian basis sets.
Left panel: Average and standard error of the RMS error ϵ for different values of the Gaussian width parameter σ as a function of the bias iterations. For each line, 20 independent simulations with the same parameters were run. The width of the Gaussians is given in units of the distance between the centers of the Gaussian d. The shaded areas show the standard error and the black dashed line denotes the threshold value of $\epsilon = 0.5\ k_\mathrm{B}T$.
Right panel: Required number of bias iterations until the threshold value is reached, together with the final error at the end of the simulation for the different basis sets.

The time evolution of the RMS error shows only minor differences for all but the extremal choices. It can be seen that a too small choice ($\sigma = 0.5d$) results in a basis set that is not able to represent the underlying FES very well, as the error stays constant on a moderate level after the initial phase. On the other hand, the choice of Ref. 19 ($\sigma = d$) works well but can be improved further. Of note is that the bias did not converge correctly for this choice in 2 out of the 20 simulations. As non-converged simulations would largely impact the values of error measures, these were excluded and replaced by analog simulations.

Similar problems could not be observed with narrower Gaussians. On the contrary, choosing a smaller width parameter results in a significantly better approximation at short times. The time until the threshold error is reached reduces drastically when using smaller Gaussians compared to the choice $\sigma = d$. As a good middle ground, the range $\sigma \in [0.6d, 0.9d]$ gives the best results for this system. Within that range, the results are rather robust with respect to parameter changes. For the further studies, a value of $\sigma = 0.75d$ is chosen.

As second parameter of the basis set, the number of basis functions is investigated using the same range as for the Sym8 wavelets in the previous section. Again, the protocol of Section 6.5.1 is used. The width of the Gaussians is set to $\sigma = 0.75d$ and the number of basis functions is varied in the range $N_{\mathrm{bf}} \in [14, 34]$. For each basis set, the quality of the obtained FES is assessed. The results are given in Figure 6.7.

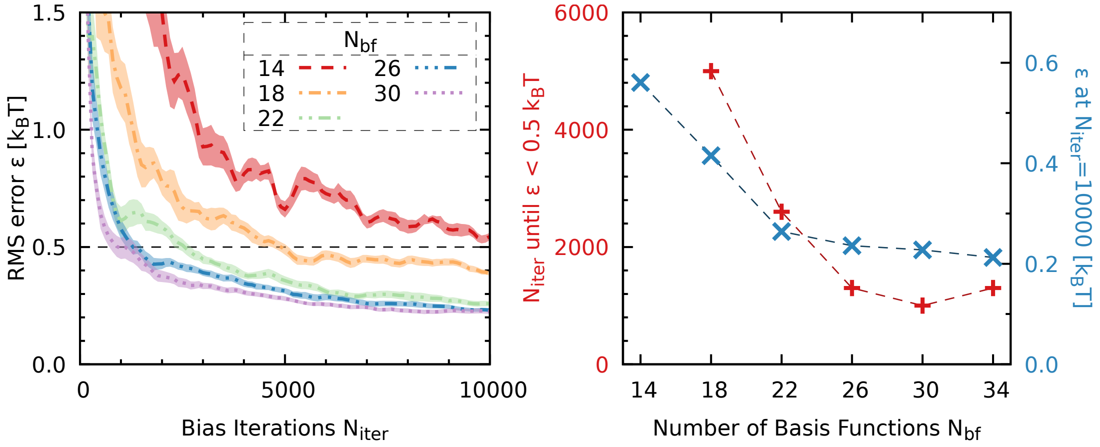


Figure 6.7: Study of different numbers of basis functions for the Gaussian basis set.
Left panel: Average and standard error of the RMS error ϵ for different numbers of Gaussian basis functions as a function of the bias iterations. For each line, 20 independent simulations with the specified number of basis functions were run. The shaded areas show the standard error and the black dashed line denotes the threshold value of $\epsilon = 0.5\ k_{\mathrm{B}}T$.
Right panel: Required number of bias iterations until the threshold value is reached, together with the final error at the end of the simulation for the different basis sets. The number of iterations is missing for 14 basis functions because the required accuracy was not reached.

For this study, the trend within the investigated range is very clear: A higher number of Gaussian basis functions results in a better approximation of the FES at the end of the simulation. When fewer than 20 basis functions are used, only a very coarse representation of the bias is possible. On the other hand, for more than 20 basis functions, the differences are minimal and only slight improvements can be seen, when adding more basis functions. The time until the threshold error of $\epsilon = 0.5\ k_{\mathrm{B}}T$ is reached shows a minimum in the investigated range, that is at 30 basis functions. Due to the increased number of optimization coefficients, it is expected that increasing the number of basis functions beyond the investigated maximum of 34, would further increase the time until a good approximation is obtained. For fewer basis functions, the time gradually increases until the condition is no longer met within the simulation time when using only 14 basis functions. Still, already for 22 basis functions both the approximation and the time of convergence do not differ significantly from the best results. For a fair comparison with the other localized functions, this number will be used in the following.

6.5.4 *Cubic B-spline basis functions: number of basis functions*

The basis set consisting of cubic B-splines has only the number of basis functions as a tunable parameter. Using the same protocol as before (Section 6.5.1), VES simulations with spline bases and different numbers of basis functions are performed. The obtained error of the FES approximations is given in Figure 6.8, together with the time until a threshold error of $\epsilon < 0.5\ k_{\mathrm{B}}T$ is reached and the error at the end of the simulation.

When using a spline basis, a good representation of the FES at the end of the simulation is obtained for all sets with fewer than 30 basis functions. If only 14 are used, the number of iterations until the threshold of $\epsilon < 0.5\ k_{\mathrm{B}}T$ is reached is significantly longer than for $18 \leq N_{\mathrm{bf}} \leq 26$. The best approximation with the fastest convergence is obtained with 22 basis functions. Starting from 30 basis functions, the results from the individual simulations differ significantly and some of them do only yield a very coarse approximation. Although a

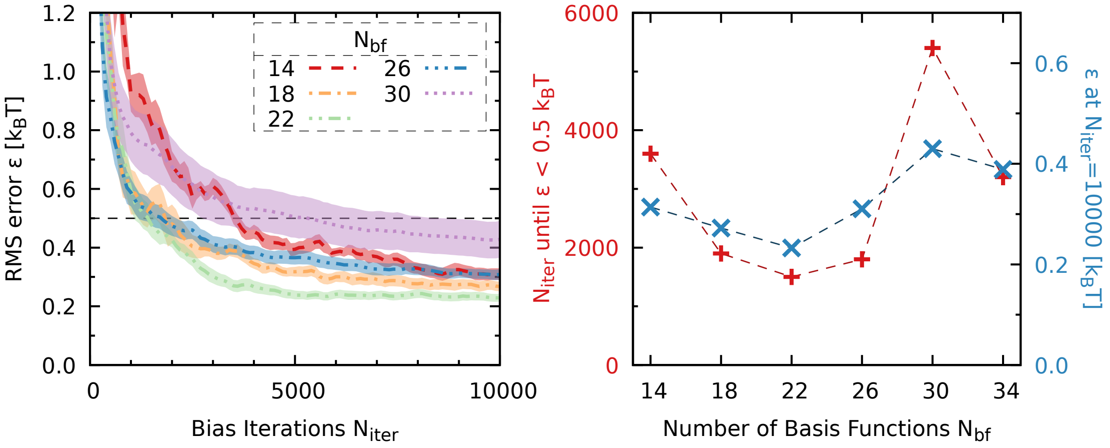


Figure 6.8: Study of different numbers of basis functions for the cubic B-spline basis set. *Left panel*: Average and standard error of the RMS error ϵ for different numbers of cubic B-spline basis functions as a function of the bias iterations. For each line, 20 independent simulations with the specified number of basis functions were run. The shaded areas show the standard error and the black dashed line denotes the threshold value of $\epsilon = 0.5\ k_\text{B}T$.
Right panel: Required number of bias iterations until the threshold value is reached, together with the final error at the end of the simulation for the different basis sets.

good approximation is still obtained most of the time, the large differences between runs and the possibility of obtaining clearly subpar results signals that great care has to be taken to not use too many basis functions for the cubic B-spline basis set. For comparison simulations with the other basis sets in the next section, 22 basis functions will be used.

7

VES SIMULATIONS WITH LOCALIZED BASIS FUNCTIONS

To evaluate the performance of the new basis functions, simulations on different systems are performed, going from model potentials in one and two dimensions, to a realistic system of the association process of calcium with carbonate in water. Their performance is compared to VES simulations with global functions (Legendre or Chebyshev polynomials, whichever is found to perform better) and in the case of calcium carbonate association also to well-tempered metadynamics simulations.

The data supporting the results of this chapter is openly available at Zenodo [198] (DOI: `10.5281/zenodo.5851773`). All the input files and analysis scripts required to reproduce the reported results are available on PLUMED-NEST (www.plumed-nest.org), the public repository of the PLUMED consortium [39], as plumID:22.001 at `https://www.plumed-nest.org/eggs/22/001`. The presented results are also currently under preparation for a publication [199].

7.1 ONE-DIMENSIONAL DOUBLE-WELL POTENTIAL

The first system to be considered is the one already used for determining optimal parameters of the localized basis sets in Section 6.5. A single particle is moved in an artificial potential of the form

$$U(x) = x^4 - 4x^2 + 0.7x \tag{7.1}$$

using the `ves_md_linearexpansion` tool of the VES code as a Langevin dynamics solver.

The same setup and parameters that were specified in Section 6.5.1 are used: The temperature is set to $T = 0.5/k_{\mathrm{B}}$, such that the barrier height from left to right is about 10 $k_{\mathrm{B}}T$. The friction coefficient is set to $\gamma = 10$ and each simulation is run for 5×10^6 steps with time step $\delta t = 0.005$.

Using the x-coordinate of the potential as the CV, the reference FES is given by the potential of eq. (7.1), $F(x) = U(x)$, ignoring an additive constant. The reference FES, together with estimates from an exemplary simulation, can be seen in the left panel of Figure 7.1. VES Simulations with four different basis sets are run, Sym8 wavelets, Gaussians, cubic B-Splines, and Legendre polynomials. The number of basis functions is fixed to 22 for each set. The VES bias is expanded in the range $x \in [-3, 3]$ and the coefficients α of the bias potential are updated every 500 time steps. A uniform target distribution is employed. For the optimization, the averaged stochastic gradient descent optimization algorithm of eq. (3.29) is used. The stepsize parameter μ of the optimization algorithm is adjusted to yield the fastest convergence for each basis set. It is set to $\mu = 0.5$ for simulations using localized basis functions and decreased to $\mu = 0.1$ for the simulations with Legendre polynomials. For each basis set, 20 independent simulations are run. These are started in the global minimum of the potential (at $x = -1.41$) with different random seeds for the initial velocities and random forces.

The performance of the basis functions is evaluated by the RMS error given in eq. (3.49). Analog to the protocol of Section 6.5, the FES is determined directly from the bias every 5×10^4 steps via eq. 3.21. For each FES estimate, the RMS error ϵ is calculated, where the reference was obtained directly from the potential. The time evolution of all simulations can be seen in Figure 7.2.

For better comparison, the time evolution of the average value $\langle \epsilon \rangle$ and the standard error $\hat{\sigma}_{\epsilon}$ is also calculated. It is given in the right panel of Figure 7.1. It can be observed that, on average, the FES (or equivalently the bias) converges considerably faster with the localized basis functions than with the delocalized Legendre polynomials. Furthermore, the localized basis functions converge to a

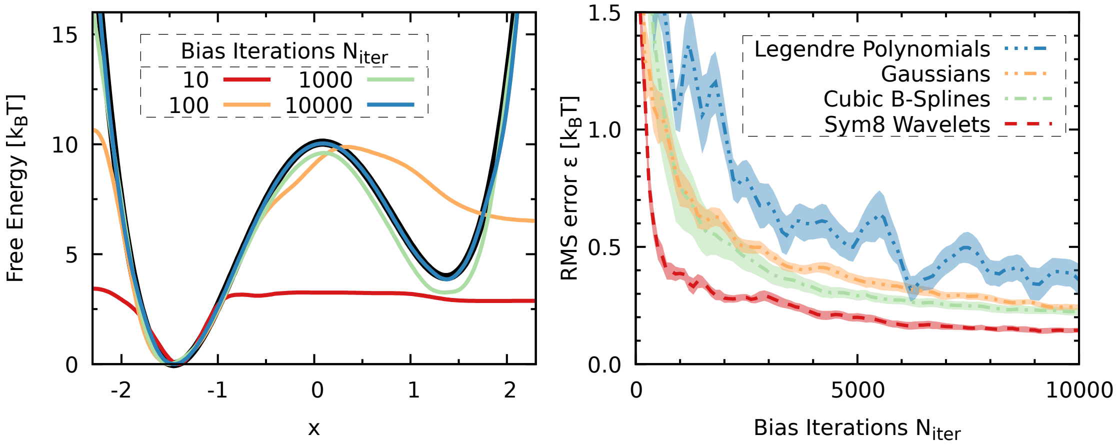


Figure 7.1: Results for VES simulations with different basis sets on the one-dimensional double-well potential.
Left panel: The reference FES, along with the FES estimates obtained directly from the bias using the wavelet basis functions at different number of bias iterations for one exemplary run. The estimates have been shifted such that $\min F(x) = 0$.
Right panel: The RMS error of the FES directly from the bias for the different basis functions as a function of the number of bias iterations. The lines denote the average over 20 independent runs and the shaded areas the corresponding standard error.

better estimate of the FES as indicated by the smaller RMS error. The wavelets perform the best of the three localized basis functions.

Noticeable in the right panel of Figure 7.1 are also considerably larger fluctuations in the average RMS error and larger standard error for the Legendre polynomials. The reason for this is twofold, as can be seen by looking at the RMS error for the individual runs, which is given in Figure 7.2. First, focusing on the difference between Sym8 wavelets and Legendre polynomials, the bias potential (and thus the error of the FES) is fluctuating more when using Legendre polynomials than with Sym8 wavelets within each individual simulation. Second, there is a more significant difference between the runs for the Legendre polynomials. In comparison, the wavelets show much more robust behavior with considerably smaller fluctuations within individual runs and smaller differences between runs. A similar effect can be seen for the Gaussians and cubic B-splines, though they do not behave as well as the wavelets.

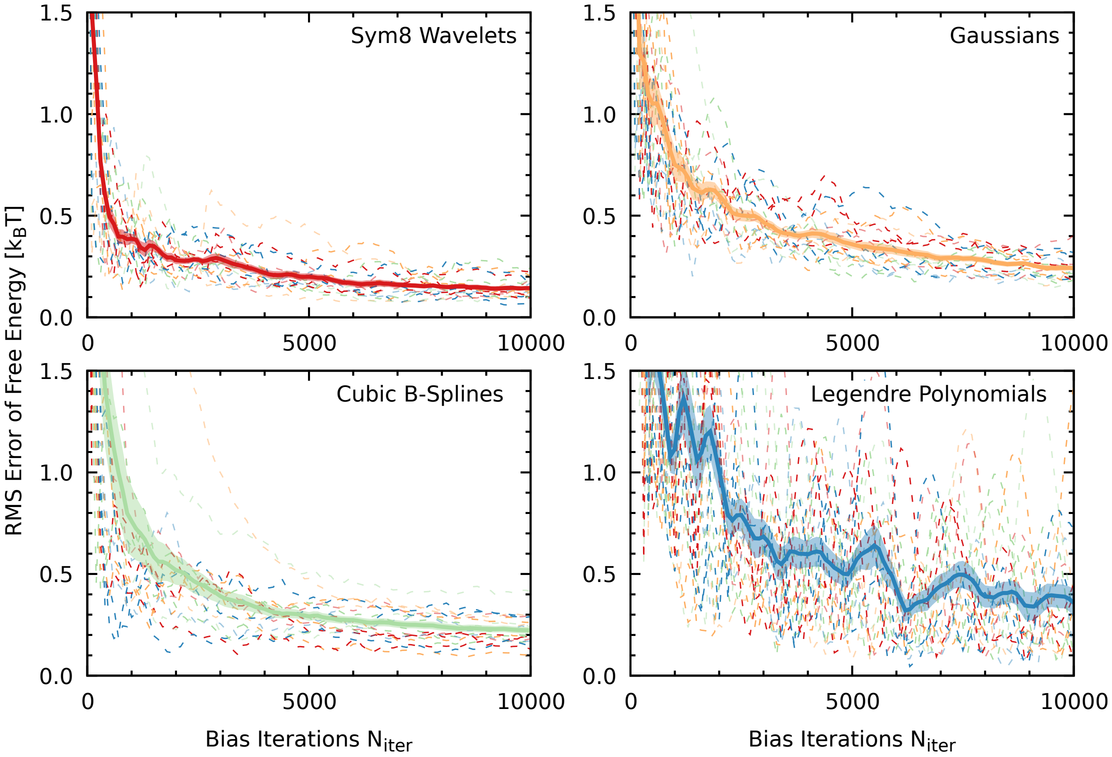


Figure 7.2: RMS error of the FES of all conducted VES simulations on the one-dimensional double-well potential.
Each panel shows the obtained RMS error of the FES calculated directly from the bias potential as a function of the number of bias iterations for a different basis set (*top left*: Sym8 wavelets, *top right*: Gaussians, *bottom left*: cubic B-splines, *bottom right*: Legendre polynomials). Each dashed line gives the result for an individual run, while the solid lines denote the average over the 20 runs and the shaded areas the corresponding standard error.

Therefore, for this simple system, already the benefits of using localized basis functions are visible.

7.2 TWO-DIMENSIONAL WOLFE-QUAPP POTENTIAL

The second model potential to be considered is the two-dimensional Wolfe-Quapp potential [200, 201], given by

$$U(x,y) = x^4 + y^4 - 2x^2 - 4y^2 + xy + 0.3x + 0.1y. \tag{7.2}$$

The potential has two states separated by a barrier along the y-coordinate, while the system is relatively mobile along the x-coordinate. Still, due to a strong coupling between the x- and y-coordinate, it is essential to consider both coordinates as biased CVs to get a good sampling. Both the x-coordinate and the y-coordinate are taken as CVs, such that the reference FES will be given by the potential, $F(x, y) = U(x, y)$ (up to an additive constant). The reference FES can be seen in Figure 7.3 along with projections on the x- and y-coordinates. Using the VES method, both CVs are biased in the interval $[-3, 3]$.

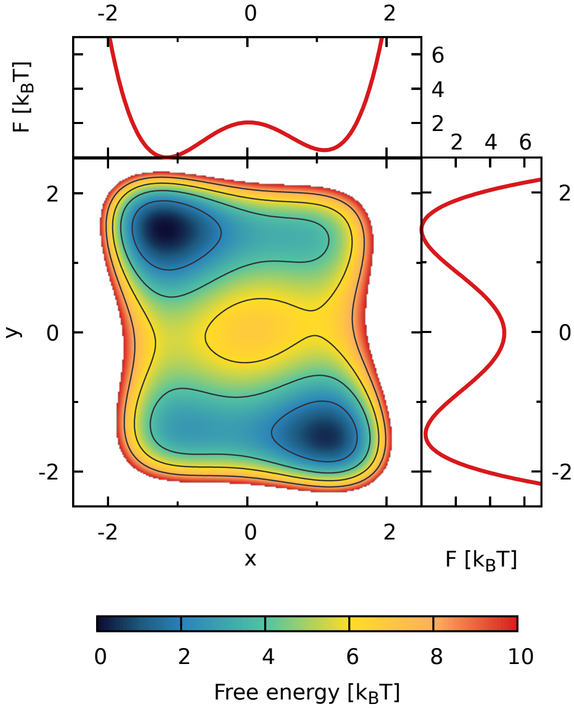


Figure 7.3: The reference FES of the Wolfe-Quapp system given by eq. (7.2) along with free energy projections on the x- and y-coordinates.

The two-dimensional bias potential is thus expanded in a tensor product basis set of one-dimensional basis functions. For each CV, 22 one-dimensional basis functions are used, resulting in 484 two-dimensional basis functions in total. The temperature is set to $T = 1/k_{\mathrm{B}}$. Using the averaged stochastic gradient descent optimization algorithm, the stepsize parameter is set to $\mu = 0.5$ for all simulations. For each basis set, 20 independent simulations are run, where the

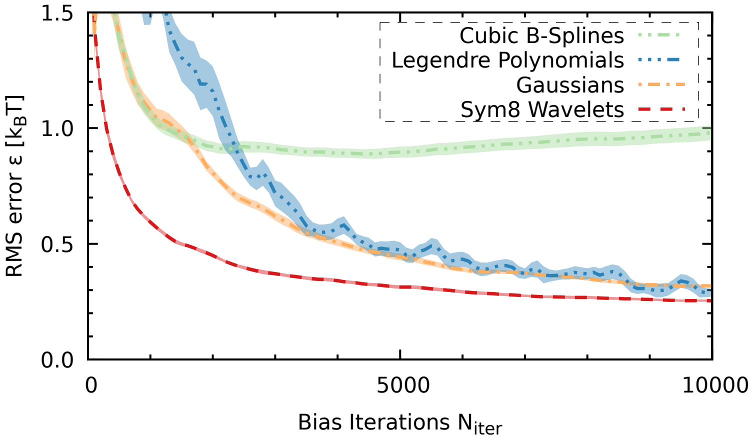


Figure 7.4: RMS error for VES simulations with different basis sets on the two-dimensional Wolfe-Quapp potential.
Shown is the RMS error of the FES directly from the bias for the different basis functions as a function of the number of bias iterations. The lines denote the average over 20 independent runs and the shaded areas the corresponding standard error.

particle is initially placed in the global minimum at $\{-1.17, 1.48\}$. Otherwise, the same basis functions and simulation parameters are employed as for the one-dimensional potential in the previous section.

Analog to the protocol of the previous section, the FES is determined directly from the bias every 5×10^4 steps via eq. (3.21). For each FES estimate, the RMS error ϵ is calculated, where the reference was obtained directly from the potential. The time evolution of the average value $\langle \epsilon \rangle$ and the standard error $\hat{\sigma}_\epsilon$ is presented in Figure 7.4. Except for the cubic B-splines, similar trends as for the one-dimensional system in the previous section can be seen. The simulations using localized basis functions reach a small RMS error much faster than the Legendre polynomials. For this system, the final error of the simulations with Gaussians is of the same order as for the Legendre polynomials, while the final error with Sym8 wavelets is still smaller. The simulations with cubic B-splines perform the worst of all the basis functions and do not yield usable results for this system. The standard error of the simulations with Legendre polynomials is also larger than for simulations with Gaussians or wavelets.

To further investigate, the free energy difference between the two main states is calculated. Because the main contributions come from the regions with low free energy, the exact definition of the two states has only minor influence as long as it includes the area around the minimum at $\{-1.17, 1.48\}$ and $\{1.12, -1.49\}$, respectively. Thus, for simplicity the $y = 0$ line is chosen to separate the states, that is $A : y \in [-3, 0)$ and $B : y \in (0, 3]$. From the potential of eq. (7.2), the reference value $\Delta F_{\text{ref}} = 0.223$ is calculated.

In Figure 7.5, the free energy difference between the two states is shown for all four basis sets as a function of bias iterations for the individual runs together with the average value. Here, the previously given analysis can be

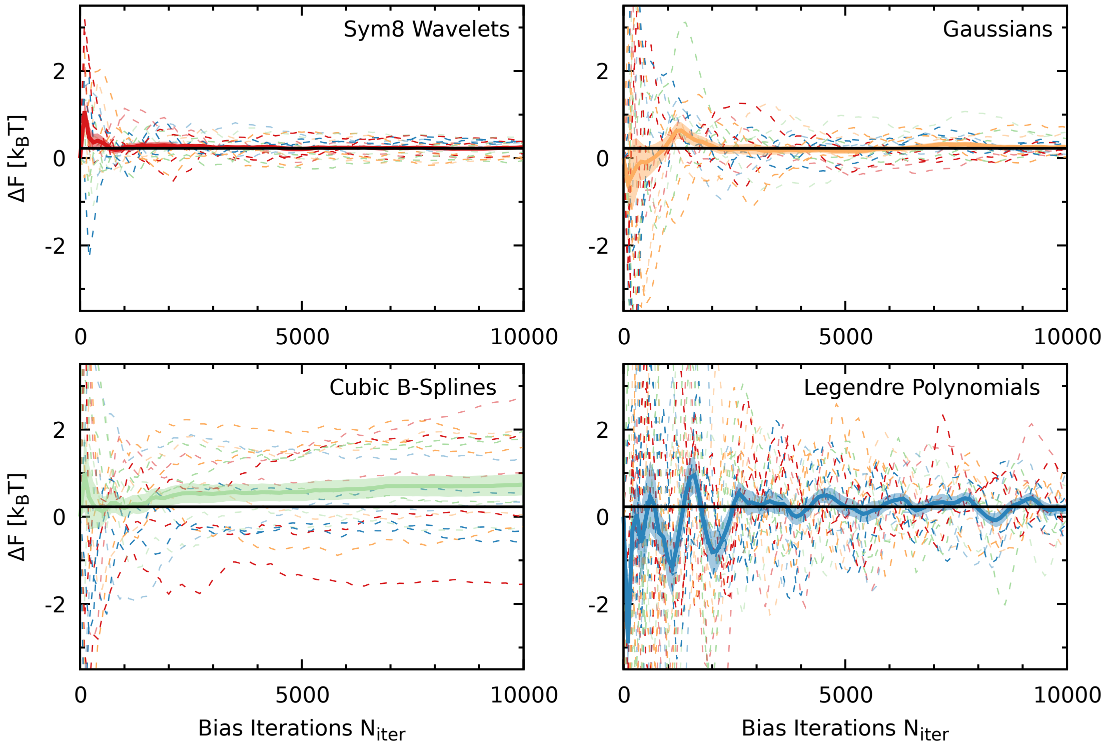


Figure 7.5: Free energy difference estimates of all conducted VES simulations on the two-dimensional Wolfe-Quapp potential.
Each panel shows the obtained free energy difference ΔF (Section 3.4, eq. 3.51) between the two states separated by the $y = 0$ line as a function of the number of bias iterations for a different basis set (*top left*: Sym8 wavelets, *top right*: Gaussians, *bottom left*: cubic B-splines, *bottom right*: Legendre polynomials). Each dashed line gives the result for an individual run, while the solid lines denote the average over the 20 runs and the shaded areas the corresponding standard error. The reference value of $\Delta F_{\text{ref}} = 0.223$ is shown as black solid line.

further refined: The wavelet simulations exhibit far smaller fluctuations within individual runs than all other simulations, with the Gaussians coming a close second. The individual runs of the Legendre polynomials fluctuate most during the simulation time. This can also be seen when looking at the average value: For simulations with Gaussians and especially Sym8 wavelets, the average value shows only minor fluctuations after an initial transient. It is therefore clear that the wavelet and Gaussian simulations converge substantially better and faster than the ones with Legendre polynomials.

7.3 ROTATED WOLFE-QUAPP POTENTIAL

To test the behavior when biasing only a suboptimal CV, a model potential that is a rotated and scaled version of the Wolfe-Quapp potential of eq. (7.2) is considered. It has been previously used as a test case for biasing suboptimal CVs [21, 25]. Following Ref. 25, the Wolfe-Quapp potential is rotated by an angle of $\theta = -0.15\pi$ and the resulting potential is given by the equation

$$\begin{aligned} U(x,y) = 1.34549x^4 + 1.90211x^3y + 3.92705x^2y^2 - 6.44246x^2 \\ -1.90211xy^3 + 5.58721xy + 1.33481x + 1.34549y^4 \\ -5.55754y^2 + 0.904586y. \end{aligned} \tag{7.3}$$

The respective FES at $T = 1/k_{\mathrm{B}}$ is shown in Figure 7.6 together with projections on the x- and y-coordinates.

Only the x-coordinate is taken as a biased CV, which results in missing orthogonal slow degrees of freedom (the y-coordinate). The reference FES for the x-coordinate can be obtained by integrating over the y-coordinate, $F(x) = -\beta^{-1} \log \int \mathrm{d}y\, e^{-\beta U(x,y)}$. The Langevin dynamics parameters are set to the same values as in the previous section ($T = 1/k_{\mathrm{B}}$, $\gamma = 10$, $\delta t = 0.005$). The VES bias potential is expanded in the range $x \in [-3, 3]$ and the number of basis functions is fixed to 22 for each basis set. As before, a uniform target distribution is used and the coefficients of the bias potential are updated every 500 steps. For each of the basis sets (Sym8 wavelets, Gaussians, cubic B-Splines, and Legendre

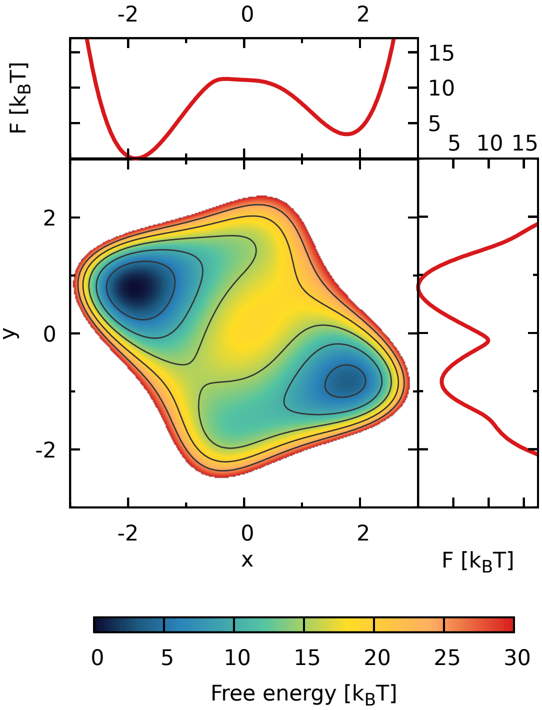


Figure 7.6: The reference FES of the rotated Wolfe-Quapp system given by eq. (7.3) along with free energy projections on the x- and y-coordinates.

polynomials), 20 independent simulations are run, that are initialized in the global minimum.

For this system, using the averaged stochastic gradient descent optimization algorithm did not result in good convergence for the localized basis functions. Therefore, the Adam algorithm [164] is used instead. Details of the Adam algorithm were presented in Section 3.2.3 and the explicit update rule for the coefficients is given in eq. (3.32). It has three metaparameters, which are set to the literature values: $c_1 = 0.9$, $c_2 = 0.999$, and $\epsilon = 10^{-8}$. High sensitivity of the convergence to the stepsize η of the Adam algorithm was noticed when performing initial trial simulations. Although the standard value of $\eta = 0.001$ works in most cases, the convergence of the bias is slow, especially for simulations with Sym8 wavelets. Increasing it to $\eta = 0.005$ provides much better behavior, whereas increasing it even further results in non-converging simulations with Legendre polynomials. In the following, $\eta = 0.005$ is used for all simulations with the Adam algorithm but it shall be noted explicitly that the choice of parameters seems crucial for good convergence.

While the usage of the Adam algorithm helps to improve the convergence for this system, worse performance in comparison to the averaged stochastic gradient descent algorithm was found when testing it on the systems that were described in the previous sections. Therefore, it seems apparent that further investigation is needed to understand the optimal choice for stochastic optimization in the VES method. The choice very likely depends on the form of the bias potential (e.g., a linear expansion versus a neural network or a model) and the basis functions used. An interesting idea might be to combine ideas from different algorithms, similar as it was done in Ref. 25 where the authors introduced a combination between AdaGrad and the averaged stochastic gradient descent algorithm. However, detailed investigations of the stochastic optimization algorithm used within VES are beyond the scope of the current work.

For the assessment of the performance of the different basis sets, the same measures as before are used: The RMS error ϵ and the free energy difference ΔF between the two major states. To calculate them, the FES is determined directly from the bias every 5×10^4 steps via eq. 3.21. From the estimates of all simulations, the RMS error ϵ with respect to the reference obtained from the potential is calculated independently. The time evolution of the average value $\langle \epsilon \rangle$ and the standard error $\hat{\sigma}_\epsilon$ is given in Figure 7.7. The usage of the Adam optimizer for this system might explain a slightly different behavior compared to the results for the previous two systems, where the averaged stochastic descent algorithm was used.

Additionally, the free energy difference between the two main states is calculated. For simplicity, these are defined by the line $x = 0$, although it is clear that the suboptimal CV does not really separate the states well. The reference value, $\Delta F_{\text{ref}} = 3.315$, is calculated from the projection on the x-coordinate of the potential. In Figure 7.8, the free energy difference between the two states is shown for all four basis sets as a function of bias iterations for the individual runs together with the average value. As expected, due to the usage of a suboptimal CV, the convergence behavior is slightly worse than for the previous

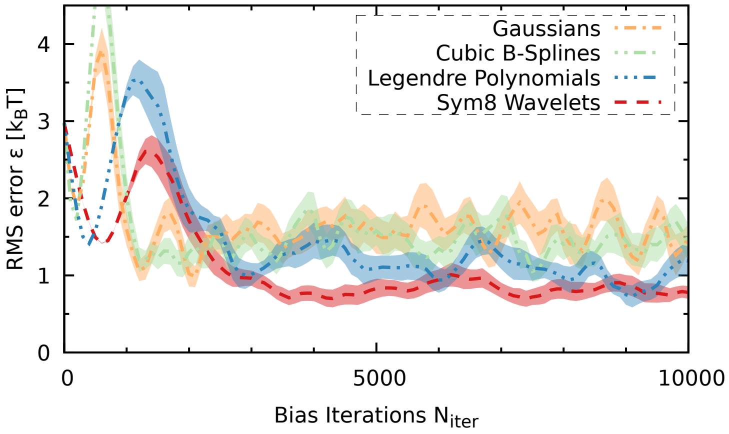


Figure 7.7: Results for the rotated Wolfe-Quapp potential.
Shown is the RMS error of the FES directly from the bias for the different basis functions as a function of the number of bias iterations. The lines denote the average over 20 independent runs and the shaded areas the corresponding standard error.

two systems, and a longer simulation time is needed to obtain adequate convergence. Nevertheless, the wavelet simulations exhibit quite decent convergence behavior that, as before, is more robust than for the Legendre polynomials. The simulations with Gaussians and the cubic B-splines perform worse than both wavelets and Legendre polynomials.

After the localized basis functions have been tested on three different model systems, certain conclusions are drawn. The wavelet basis functions exhibit much more robust convergence behavior than the Legendre polynomials. For the wavelets, the fluctuations of the bias potential within individual runs are smaller. Additionally, the difference between independent runs is considerably smaller. The Gaussian and the cubic B-spline basis functions perform worse than the wavelets for all considered systems and do not yield usable results for some systems. Therefore, their usage is not recommended. Having established the excellent performance of the wavelets in model systems, their usage is now tested on a more realistic system.

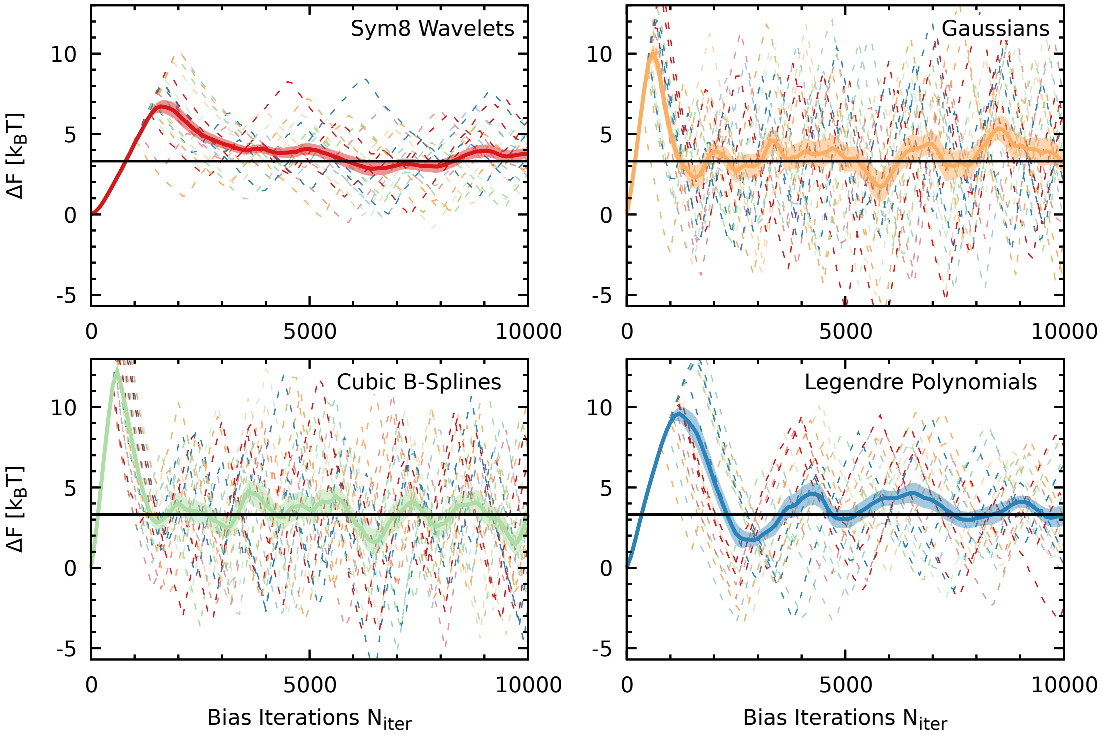


Figure 7.8: Free energy difference estimates of all conducted VES simulations on the two-dimensional rotated Wolfe-Quapp potential.
Each panel shows the obtained free energy difference ΔF (Section 3.4, eq. 3.51) between the two states separated by the $x = 0$ line as a function of the number of bias iterations for a different basis set (*top left*: Sym8 wavelets, *top right*: Gaussians, *bottom left*: cubic B-splines, *bottom right*: Legendre polynomials). Each dashed line gives the result for an individual run, while the solid lines denote the average over the 20 runs and the shaded areas the corresponding standard error. The reference value of $\Delta F_{\mathrm{ref}} = 3.315$ is shown as black solid line.

7.4 ASSOCIATION PROCESS OF CALCIUM CARBONATE

To study the performance of wavelet basis functions for a realistic system, the association process of a calcium carbonate ion-pair in water is considered. For this, molecular dynamics simulations with explicit solvent are performed. As molecular dynamics software, the the LAMMPS code [202] (5Jun2019 release) is used. It was interfaced with the PLUMED2 code [38, 39] to perform biased enhanced sampling simulations with the VES method. In Refs. 203, 204, a custom calcium carbonate force field was developed, which is used together with the SPC/Fw water model [205].

Force field

For the water molecules, the bonds and angles between the oxygen and hydrogen are restricted with a harmonic restraint. Additionally, the oxygen atoms of all water molecules interact with a Lennard-Jones potential [56] (see eq. (2.10)) that has a cutoff at 9 Å.

An harmonic restraint is also applied to the bonds of the carbonate ion. The potential fixing the angles O–C–O of the carbonate ion is a special Class2 type potential described in Ref. 206. Additionally, the distance between the carbon atom and the plane formed by the three oxygen atoms of the carbonate ion is constrained.

The interaction between the calcium and carbonate ion is realized by a Buckingham style potential [57] that includes a Mei-Davenport-Fernando (MDF) taper [207]. Instead of setting a fixed cutoff distance, the taper gradually turns off the strength of the potential in a transition area, that is, the potential is now given by

$$U_{\mathrm{MDF}}(r) = U(r)\, f_{\mathrm{MDF}}(r), \tag{7.4}$$

with

$$f_{\mathrm{MDF}}(r) = \begin{cases} 1 & \text{for} \quad r < r_m, \\ (1-r)^3(1+3r+6r^2) & \text{for} \quad r_m < r < r_{\mathrm{cut}}, \\ 0 & \text{for} \quad r \geq r_{\mathrm{cut}}, \end{cases} \tag{7.5}$$

where r_m denotes the smaller distance of the spheric transition area and r_{cut} the cutoff distance at which the potential strength is zero. Potentials of the same type are used for the interactions between the carbonate oxygens and the water atoms. The calcium ion and the water oxygens interact by a Lennard-Jones potential with an additional MDF temper.

The charged interactions between all atoms are conveyed by a Coulomb potential, where the long-range interactions are calculated in reciprocal space via the PPPM method [60, 61]. Neighbor lists with the default LAMMPS parameters are used to reduce the number of interactions to consider at each time step.

The exact parameters of the force field, as well as an input file for LAMMPS, can be found in the supporting information of Ref. 204. At the time of the publication of Ref. 204, the pair potentials with the MDF taper were not available in the standard version of LAMMPS and had to be included in the code manually. In the provided input file of Ref. 204, these custom potentials are denoted by *gulp*. As the potentials with a MDF temper are now available, the provided force field can be run in the standard version of LAMMPS by replacing all *gulp* style potentials with the *mdf* ones in the input. An updated version of the input file can be found on Zenodo [198].

Computational details

The computational setup follows the one of a previous metadynamics study of the association process [120] using this force field. There, well-tempered metadynamics simulations with 25 parallel walkers were used to characterize the FES at different temperatures. To be able to compare the results to this previous study, the setup and parameters are tried to be matched as close as possible.

In a periodic cubic box, a single Ca^{2+}– CO_3^{2-} ion-pair and 2448 water molecules are placed. The system is equilibrated in the NPT ensemble at a constant temperature of 300 K and a constant pressure of 1 Bar for 500 ps. All subsequent simulations are performed in the NVT ensemble using a constant temperature of 300 K and cubic box with side lengths 41.69 Å. A time step of 0.001 ps is used for all simulations. Unbiased MD simulations of 5 ns length are run from which in total 75 snapshots are selected to use as initial configurations for the biased simulations. All simulations are performed at a constant temperature of 300 K using a Nosé-Hoover thermostat [64, 65, 208] with a chain length of 5 and a relaxation time of 0.1 ps. For the NPT equilibration, a Nosé-Hoover barostat with a relaxation time of 1 ps to keep a constant pressure of 1 Bar is employed.

The same CVs as in Ref. 120 are used, namely the distance between the Ca and C atoms, $r_{\mathrm{Ca-C}}$, and the coordination number of Ca with water, N_{coord}. In practice, the coordination number is calculated using a switching function

$$N_{\mathrm{coord}} = \sum_i s_i = \sum_i \frac{1 - \left(\frac{r_i - d_0}{r_0}\right)^n}{1 - \left(\frac{r_i - d_0}{r_0}\right)^m}, \tag{7.6}$$

where the index i runs over all oxygen atoms of the water molecules and r_i denotes the distance from the atom to the calcium atom. The parameters of the switching function are set to the same values as in Ref. 204, that is, $r_0 = 1.0$ Å, $d_0 = 2.1$ Å, $n = 4$, $m = 8$.

Mirroring the original work [120], the technique of multiple walkers [132] is used. Convergence of the bias is improved by running 25 walkers in parallel, where each walker starts from a different initial configuration. As basis functions for the VES method, Sym10 wavelets and Chebyshev polynomials are employed. For the CV corresponding to the distance between the Ca ion and C atom of the carbonate ion, 60 basis functions are used in the range $r_{\mathrm{Ca-C}} \in [2\,\text{Å}, 12\,\text{Å}]$. For the CV corresponding to the coordination number, 30 basis functions are used in the range $N_{\mathrm{Coord}} \in [5, 9]$. The total number of two-dimensional basis functions is then 1200. Due to the usage of multiple walkers, the coefficients of the bias potential are updated more frequently, that is every 10 MD steps (the total number of data points for each iteration is then 250). For the Sym10 wavelets, the averaged stochastic gradient descent optimization algorithm with a step size of $\mu = 0.001$ is used. For simulations with Chebyshev polynomials, this did not always result in stable simulations and the step size was lowered to $\mu = 0.0005$ for these. A well-tempered target distribution (see eq. (3.24)) with a bias factor of 5 is employed. The target distribution is updated iteratively every 100 bias potential updates (1000 MD steps). Each walker is run for 3 ns, resulting in a cumulative simulation time of 75 ns.

For comparison, additional well-tempered metadynamics (WTMetad) [106] simulations are performed, using the same setup as in Ref. 120. The bias factor is set to 5. For the Gaussians, an initial height of 1 $k_{\mathrm{B}}T$ is used and the widths are set to 0.2 Å and 0.1 for the distance and coordination number CV, respectively.

Gaussians are deposited every 1 ps (1000 MD steps). As for the VES simulations, each walker is run for 3 ns, resulting in a cumulative simulation time of 75 ns.

To focus the sampling in the part of the configuration space of interest for the association process, an artificial repulsive wall at a Ca–C distance of 11 Å is added in all simulations to prevent the ions from moving further apart. In practice, this is implemented by adding a harmonic bias of the form $\kappa(r_{\mathrm{Ca-C}} - r_0)^2$ when $r_{\mathrm{Ca-C}} \geq r_0$. In the present case, the parameters are set to $\kappa = 12$ eV and $r_0 = 11$ Å.

Besides obtaining FES estimates from the bias via eq. (3.21) or by summing over the Gaussians via eq. (3.8) for VES and metadynamics, respectively, another estimate is also obtained for all simulations via a reweighted kernel density estimation. By reweighting the trajectories via eq. (3.19), where the delta function is replaced by a Gaussian kernel, a smooth estimate can be obtained. For this, the implementation of kernel density estimation of PLUMED2 is used, where the bandwidths of the Gaussians are set to 0.05 Å and 0.05 for the Ca–C distance and coordination number CV, respectively. To avoid effects from the initial phase where the bias still fluctuates significantly, the first 200 ps of each walker trajectory are ignored. From the rest of the trajectories, samples are taken every 0.1 ps. For the metadynamics simulations, the time-dependent constant $c(t)$ is required when using the reweighting scheme described in Refs. 77, 143. During the metadynamics simulations, the constant is calculated every time a Gaussian is added using a grid of 275 × 300 points over the domain $r_{\mathrm{Ca-C}} \in [2, 13]$, $N_{\mathrm{Coord}} \in [3, 10]$ and stored together with the trajectory data.

To assess the stability of the simulations, for each of the three biasing setups (VES with wavelets, VES with Chebyshev polynomials, WTMetaD) three independent runs with different initial configurations are performed. The configurations are selected from the 5 ns of unbiased simulation and are the same for each of the biasing setups.

Results

For all simulations, the two-dimensional FES estimate from the bias at the end of the simulation, as well as the estimate from reweighting the trajectories are

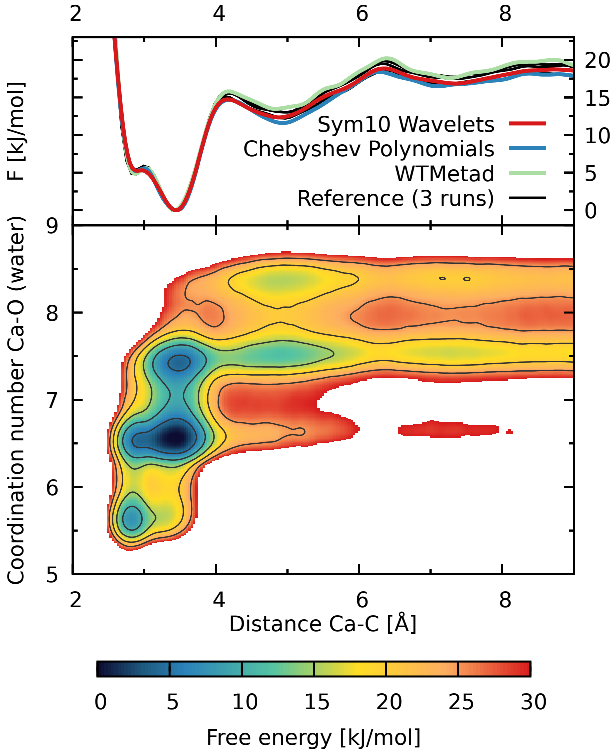


Figure 7.9: Obtained free energy surfaces for the Calcium Carbonate system.
Top panel: Projections on the distance CV $r_{\mathrm{Ca-C}}$ from the FESs obtained directly from the bias potential. Only one of the runs is shown for each biasing setup. The reference data are obtained from Ref. 120.
Bottom panel: FES as a function of both biased CVs obtained by reweighting one of the simulations with Sym10 wavelets.

calculated. An exemplary two-dimensional FES from reweighting is preented in the lower part of Figure 7.9. It can be seen, that to fully understand the association process it is necessary to consider both CVs as the solvation state of the calcium, measured by the coordination number CV, is closely coupled to the calcium-carbon distance. The minima of the FES with $r_{\mathrm{Ca-C}} < 4$ correspond to the states with contact ion-pair. The lowest state of the FES is the monodentate associated state at around 3.5 Å. At lower coordination number and smaller distance, a second minimum corresponding to the bidentate state can be seen. For larger Ca–C distance, the ions are no longer in direct contact but are separated by solvent. The states with a distance of around 5 Å correspond to the solvent-shared ion-pair, while the states around 7 Å denote where the solvation shells of the two ions barely touch. For even larger distances, the two ions are fully solvated.

Additionally to visual inspection of the 2D FESs, projections of the FES estimates on the distance CV $r_{\mathrm{Ca-C}}$ are investigated more closely. From the authors of Ref. 120, it was possible to obtain the FES as a function of this CV for three independent runs at 300 K. While these are qualitatively similar, there are some visible fluctuations between the three data sets. Therefore, these are only used to qualitatively assess the estimates from the newly performed simulations but not as a reference for quantitative performance measures. In the upper part of Figure 7.9, these reference sets are shown together with one exemplary FES directly from the bias for each biasing setup. Besides this direct comparison of selected runs, Figure 7.10 shows the obtained FES estimates of all simulations for both methods. All free energy profiles are aligned such that their minimum is at zero.

Looking at these estimates, it is observed that all free energy profiles obtained from the simulations are in a decent agreement with each other and the reference results from Ref. 120. All of the simulations capture the small barrier between the mono- and bidentate states at about 3 Å reasonably well, though it should be mentioned that this barrier in the one-dimensional profile does not represent the true barrier of the physical process due to integration over the solvent degree of freedom (i.e., the coordination number CV). For the states above 4 Å, some differences between runs are observed. However, similar variance can be observed between the three reference runs from Ref. 120. Therefore, it is difficult to say what the correct free energy profile is. Furthermore, the results in Figure 7.10 are obtained at the end of the simulations and do not reflect that the bias, and thus the obtained FES, fluctuates during the simulation. Indeed, one of the main conclusions from the previous Sections 7.1–7.3 was that the fluctuations of the bias potential within individual runs were considerably smaller for the wavelets as compared to the polynomial basis functions.

To gauge the time evolution of the bias potential and FES, the free energy difference between the contact ion-pair and the loosely associated states of calcium carbonate is considered. For simplicity, the region in CV space with a Ca–C distance smaller than 4 Å is selected as the contact ion-pair state and

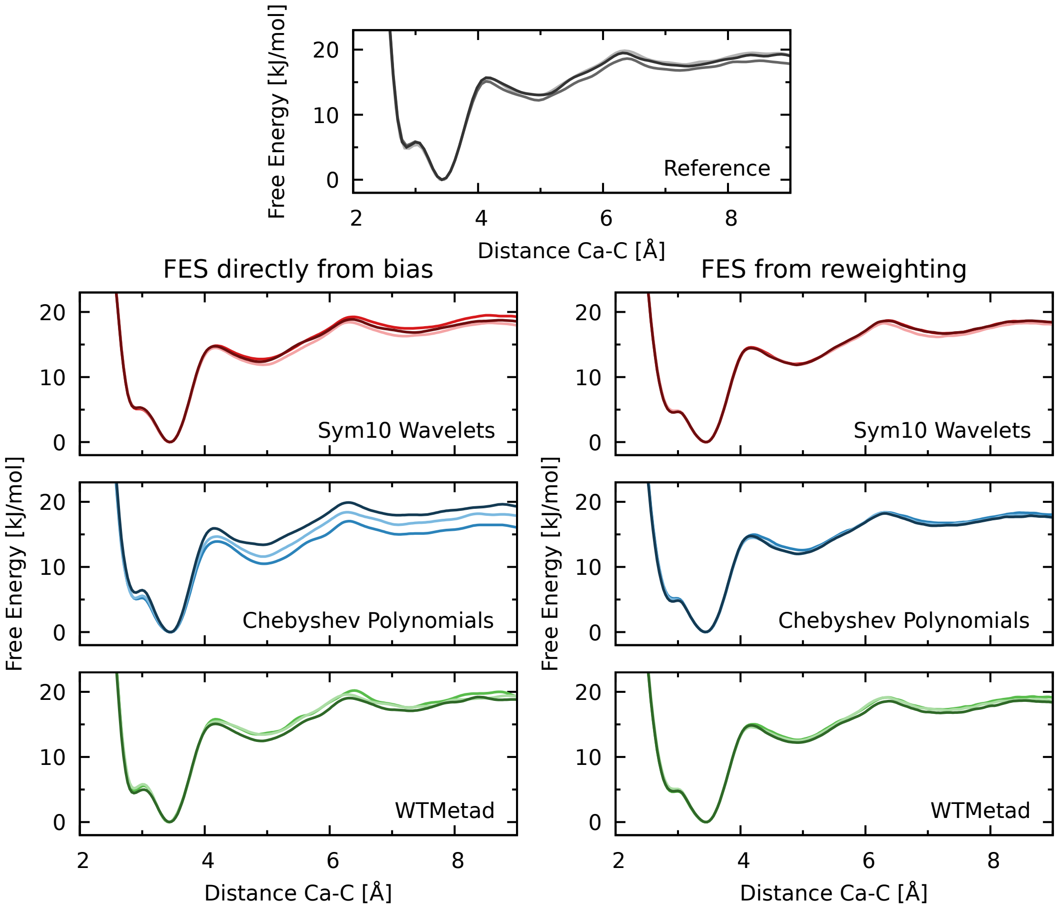


Figure 7.10: FESs as a function of the ion distance for the calcium carbonate system. Each panel shows results from three independent simulations with differently shaded lines.
Top panel: Reference FESs obtained from Ref. 204.
Left column: FES estimates directly from the bias potential for three runs each of Sym10 wavelets, Chebyshev polynomials, and Metadynamics simulations.
Right column: FES estimates obtained by reweighting the trajectories of the same runs.

the region with distances between 4 Å and 8 Å as the loosely associated state. The free energy difference is calculated according to eq. (3.51). It is of note, that this selection of the two regions does not necessarily coincide with the chemical definitions of the contact ion-pair and loosely associated states [120]. Here, the free energy difference is mainly employed to monitor the stability of the bias potential and the obtained FES. In the upper panels of Figure 7.11, the free energy difference obtained every 10 ps simulation time per walker is shown.

For each of the biasing setups, the results of the three independent runs are shown for the FES obtained directly from the bias as well as the FES obtained via reweighting.

When looking at the estimates of the free energy distances obtained directly from the bias (the left column), it is visible that the free energy differences obtained from the wavelet simulations converge faster and show less fluctuations than the ones from the Chebyshev polynomial simulations. In particular, there are considerably larger fluctuations during the runtime of the Chebyshev polynomial simulations. Furthermore, there is less difference between independent runs for the wavelets as compared to the Chebyshev polynomials. Therefore, when comparing the two biasing setups using VES, the same conclusions as for the model systems in the previous Sections 7.1–7.3 can be drawn: simulations with the wavelet basis exhibit smaller fluctuations of the bias potential within individual runs and less difference between different independent runs than when using the Chebyshev polynomials as basis functions.

The metadynamics simulations show a convergence behavior that is slightly worse compared to the wavelet simulations, but still better than for the Chebyshev polynomial simulations.

To further quantify the behavior of the simulations, the average and the standard deviation of the free energy difference over the last nanosecond of each simulations are calculated. A visualization is shown in the bottom row of Figure 7.11 while the numerical values are given in Table 7.1.

Continuing with the values obtained directly from the bias, some spread in the time-averaged values can be seen between the individual runs, though all simulations agree with each other within 1 kJ/mol. It is of note, that there is a similar spread in the three reference metadynamics simulations from Ref. 120 that are shown as grey dotted lines in the bottom row of Figure 7.11. Therefore, it cannot be said for certain what the exact reference value of the free energy difference is. Noticeably, and consistent with the time evolution of the free energy differences in the upper rows, the wavelet simulations have the smallest standard deviation values while the values are three to six times larger for the Chebyshev polynomial and metadynamics simulations.

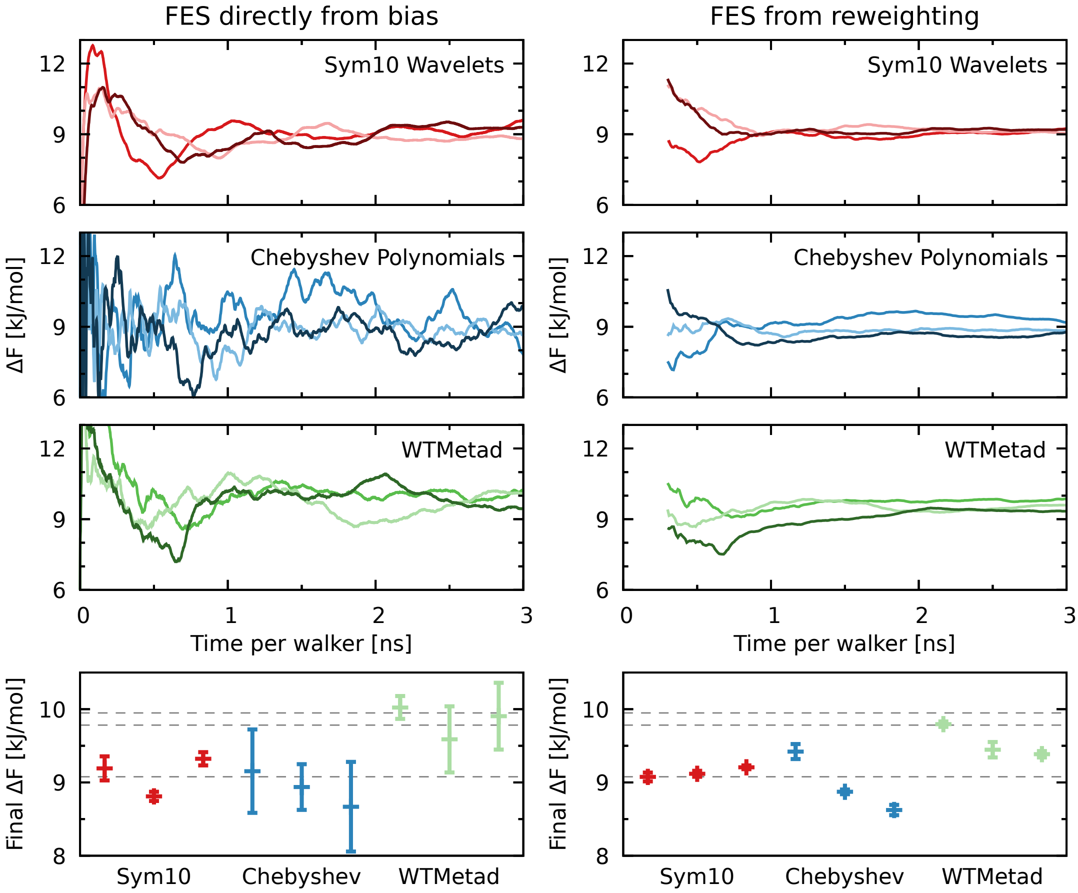


Figure 7.11: Free energy differences between the contact ion-pair and loosely associated states of the Calcium Carbonate system.
Top three rows: Time evolution of the free energy difference between the region with Ca–C distance smaller 4 Å and the region with Ca–C distance between 4 Å and 8 Å. For each biasing setup, results from three independent runs are shown, where the different color shades represent the individual runs.
Bottom row: The average of the free energy differences obtained over the last nanosecond by using 100 samples taken every 10 ps for each simulation. The error bars show the standard deviation. Additionally the results from Ref. 120 are given as grey dotted lines.
The *left column* uses the FES obtained directly from the bias, while the *right column* uses the FES obtained through reweighting where the first 200 ps of each simulation were omitted.

	run 1	run 2	run 3	average
Sym10 (bias)	9.2 ± 0.2	8.81 ± 0.06	9.32 ± 0.09	9.1 ± 0.2
Sym10 (reweighted)	9.08 ± 0.06	9.12 ± 0.04	9.21 ± 0.03	9.13 ± 0.04
Chebyshev (bias)	9.2 ± 0.6	8.9 ± 0.3	8.7 ± 0.6	8.9 ± 0.1
Chebyshev (reweighted)	9.4 ± 0.1	8.87 ± 0.03	8.62 ± 0.07	9.0 ± 0.2
WTMetad (bias)	10.0 ± 0.2	9.6 ± 0.5	9.9 ± 0.5	9.8 ± 0.1
WTMetad (reweighted)	9.79 ± 0.04	9.4 ± 0.1	9.38 ± 0.04	9.5 ± 0.1
Reference	9.94	9.08	9.78	9.6 ± 0.3

Table 7.1: Calculated values for the free energy difference between the contact ion-pair and loosely associated states of the Calcium Carbonate system.
All values are given in units of kJ/mol. For every simulation, the values were calculated from the average of the last nanosecond both directly from the bias as well as via reweighting. The given uncertainty of the runs is the standard deviation of the values of the last nanosecond. Only a single value per run could be obtained from the reference data and it is therefore not possible to give an uncertainty in this case. For the average of the three runs, the given uncertainty is the standard error from the combination of the three runs, where the uncertainties of the individual runs were neglected.

Therefore, from the results of the left column of Figure 7.11, it can be concluded that the wavelets perform the best when considering the difference between independent simulations and fluctuations within runs.

So far, the discussion was only about the estimates of the free energy differences obtained directly from the bias potential. An alternative way to obtain the FES is through reweighting. In fact, it is always a good practice to estimate the FES both directly from the bias potential and via reweighting and compare the results. The reweighting procedure assumes that the bias potential (i.e., the weights) is quasi-stationary. Therefore, it is expected that the wavelets perform better in this respect.

The right column of Figure 7.11 shows the free energy difference values obtained from the reweighted FESs. The average values and the standard deviation over the last nanosecond are given in the bottom row. Less fluctuation of the free energy difference can be seen for all of the simulations when compared to the estimates directly from the bias in the corresponding left panels. All of the wavelet results agree well with each other while there is more spread for the Chebyshev polynomial and the metadynamics simulations, but as before the spread is within 1 kJ/mol. The reweighted metadynamics values tend to be lower than values obtained directly from the bias potential and closer to the wavelet results. Like for the results obtained directly from the bias potential, it is found also for the reweighted results that the wavelets perform the best when considering the difference between independent simulations and fluctuations within runs.

Overall, from the simulations of the calcium carbonate association, it is concluded that the wavelet basis functions exhibit excellent performance. The simulations using wavelets show considerably better convergence behavior than the ones with Chebyshev polynomials. On top, the wavelet simulations also show better convergence behavior than the metadynamics simulations.

8

CONCLUSIONS

In this part of the thesis, the usage of Daubechies wavelets as basis functions for variationally enhanced sampling was introduced. Their performance was evaluated on Langevin dynamics simulations of model systems and molecular dynamics simulations of the calcium carbonate association process. Overall, the localized wavelet basis functions exhibit excellent performance and much more robust convergence behavior than the delocalized Chebyshev and Legendre polynomials used as basis functions within VES so far. In particular, the wavelet bases exhibit far smaller fluctuations of the bias potential within individual runs and smaller differences between independent runs. Less fluctuation of the bias potential is important when obtaining FESs and other equilibrium properties through reweighting, as the reweighting procedure assumes a quasi-stationary bias potential. Based on the overall results, wavelets can be recommended as basis functions for variationally enhanced sampling.

Additionally, Gaussians and cubic B-splines were tested as other types of localized basis functions. However, the Gaussian and the cubic B-spline basis functions were shown to perform worse than the wavelets for all the model systems in Sections 7.1–7.3 and did not even yield usable results in some cases. Therefore, Gaussians and cubic B-splines are currently not recommended to be used as basis functions for the VES method.

One attractive feature of the wavelets basis functions is the multiresolution property displayed in eq (5.23). Starting with the father wavelets at some given scale, a more accurate approximation of the FES can be obtained by adding mother wavelets at finer scales. In this work, only a single level of father wavelets was employed to expand the bias potential. An interesting future work would be to go beyond this and implement a multiresolution bias potential where the resolution can be increased on the fly during the simulation. Coupling this with a method to evaluate the quality of the current bias potential on the fly (for example, by using the effective sample size [102, 209, 210]) would allow to automatically construct the VES bias potential with a predefined accuracy, without the need to adapt the parameters manually.

Also, the present work was focused on a single type of wavelets, the family of Daubechies wavelets in their least asymmetric form. Although no results were shown, the Daubechies wavelets with extremal phase had also been tested initially. Due to clearly worse results compared to the Symlets, they were not included in the extensive studies. Nevertheless, other wavelet families could yield better performance for specific systems.

The basis formed by Daubechies wavelets spans an infinite range in theory. In the presented implementation, only basis functions with significant contributions within the bias range were taken into account. For the systems under investigation in this work, the bias was always expanded in a range significantly larger than the relevant states or processes of interest and no problems could be observed from this approach. However, it might result in inaccuracies at the boundaries if the bias cannot be expanded beyond the states of interest, for example if the relevant CV has only a finite range. Therefore, wavelets that are designed to be used on finite intervals, such as the boundary wavelets [211], might be worthwhile to investigate for systems with relevant states close to the boundaries of the bias.

Another possibility would be multiwavelets, which are used for example in electronic structure calculations [212]. There, instead of using a single scaling and mother wavelet function, a combination of several functions is used. The refinement equations (5.24) then use vectors for $\phi(x)$ and $\psi(x)$ and the filter coefficients h_k and g_k have a matrix form [213]. One specific type of multiwavelets that could be considered are the ones developed by Donovan, Geronimo, and Hardin [214, 215]. These are piecewise-polynomial wavelets that are compactly supported and sufficiently regular for the usage as basis functions in VES.

Part III

BIRTH-DEATH AUGMENTED SAMPLING

"Markovian process, lead us not in vain
Prove to our descendants what we did to them
Then make us go away"

— Greg Graffin [216]

9

INTERACTING REPLICAS OF MOLECULAR SIMULATIONS

The molecular dynamics simulations of the association process of calcium carbonate that are presented in Section 7.4 use the technique of multiple walkers. The idea for this, as already explained in Section 3.3, is that a single simulation might require long simulation times for good coverage of the phase space related to the problem under investigation. To reduce wall-clock time, multiple copies of the system are simulated in parallel at the same time. This is similar to running an ensemble of multiple short and independent simulations instead of a single long one, which is often beneficial for estimating macroscopic properties of the system [71].

Nevertheless, even if all states are sufficiently sampled by individual simulations, the approach of using multiple independent simulations does not overcome the rare event problem in systems with high barriers between metastable states: While local observables (such as the position and shape of free energy minima) are typically estimated correctly, good estimates of global observables (for example, the free energy difference between states) can only be obtained if multiple transitions between the states are observed within the simulation time. Simulating multiple copies of the system in parallel allows to couple the simulations to get better global sampling. Figure 9.1 shows schematically that this coupling can be viewed as adding interactions between imaginary particles (the "replicas" or "walkers") that represent the state of the system in one simulation. An example for such a coupled scheme is metadynamics simulations with multiple walkers [132], like the ones performed in Section 7.4. There, the replicas interact only indirectly by sharing the same bias potential.

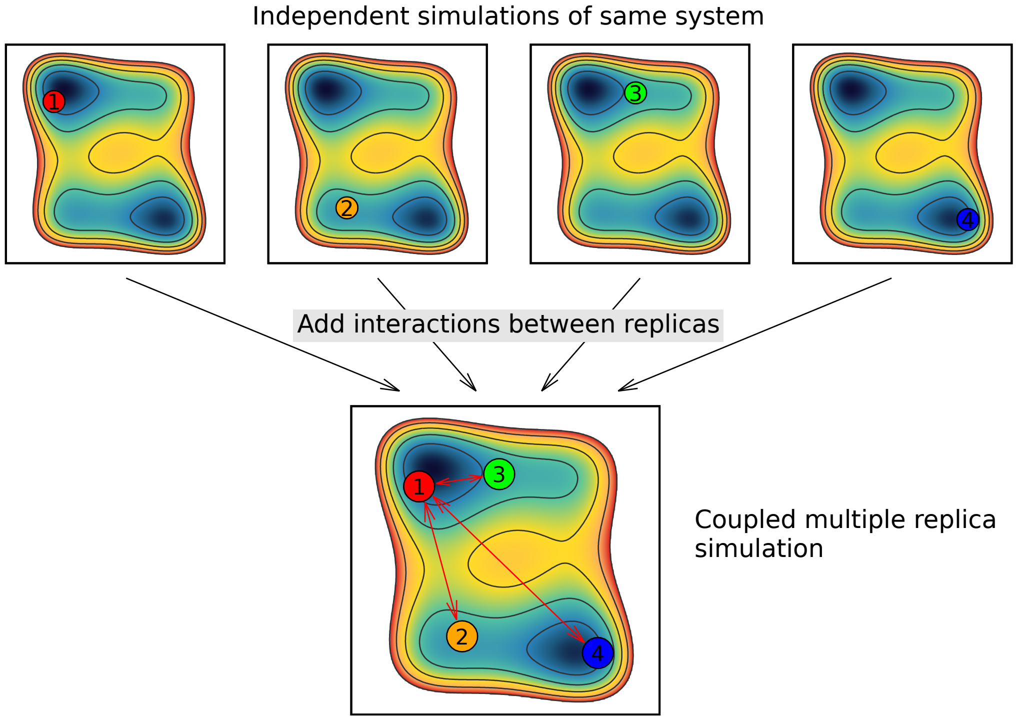


Figure 9.1: Schematic description of a simulation with multiple interacting replicas. The top row shows snapshots of four independent simulations where the current state of the system is depicted as colored circle on a two-dimensional (free) energy surface. The bottom row shows the situation when adding interactions between the different simulations: The dynamics of the simulations is no longer independent but they can be thought of as a simulation with four coupled particles exploring the same underlying (free) energy surface. Only exemplary interactions of the first simulation (particle) are shown with red arrows.

Other algorithms introduce direct interactions between the particles. While some interacting particle methods only enhance the dynamics of the particles with a term depending on all particles [171, 172], in the following the focus is on cloning methods that involve the duplication and killing of particles. These have been applied to solve reaction diffusion problems [31], train neural networks [32], explore branching graphs [33], solve quantum mechanical problems [34], or generate polymer configurations [35]. To sample rare event systems, methods that force sampling of rare trajectories [36] or distribute the particles favourably [30, 37] have been developed.

In this part of the thesis, Langevin sampling is combined with a birth-death algorithm that is a modification of the algorithm proposed in Ref. 30. The method leads to equilibrium sampling, which means that the combination of all walkers samples the equilibrium distribution of the system. This is done by killing particles in states that are currently sampled too much, and duplicating them where more sampling would be needed. The decision to perform a birth-death event is made by calculating an estimate of the current particle density via smoothing kernels and subsequent evaluation of a birth-death probability. The probability of the event then depends on the distance of the estimate to the desired equilibrium density as well as the time between evaluations.

The obtained equilibrium sampling contrasts nicely with bias-based enhanced sampling methods, such as metadynamics and VES, that do not focus on sampling the equilibrium distribution but rather enhance sampling of the full configurational space. Using both approaches together might lead to an interesting method that combines exploration and exploitation: First, the biasing method results in quick exploration of the configurational space and detects new metastable states that were not sampled beforehand due to high barriers. Second, the birth-death algorithm results in a quick equilibration of the simulated replicas, such that the information of the new states is quickly exploited and good global estimates are obtained. While this combination is out of the scope of the current work, future research in that direction is to be kept in mind.

In the remainder of this thesis part, the fundamental development of the combination of Langevin dynamics with a birth-death algorithm is presented. Chapter 10 provides the theory for augmenting overdamped Langevin dynamics with birth-death processes, where convergence to the correct distribution can be proven mathematically. Additionally, some information for the general Langevin case is provided. Chapter 11 presents the algorithm and details of the implementation. In Chapter 12, the algorithm is applied to different model systems and the behavior of the method is investigated when iterating over different parameters. The method is compared to pure LD and it is shown that accurate sampling of rare event systems can be obtained where pure LD fails.

10

THEORY OF BIRTH-DEATH AUGMENTED LANGEVIN DYNAMICS

10.1 FOKKER-PLANCK EQUATIONS FOR LANGEVIN DYNAMICS

To describe the stochastic nature of Langevin dynamics, some fundamental ideas of stochastic processes need to be introduced. The notation will mostly follow Ref. 47, while some ideas are also adapted from Refs. 42 and 217.

Markov processes and the Chapman-Kolmogorov equation

Here, only stochastic processes with the Markov property are considered, which means roughly that the process does not have memory and the time evolution at a specific point in time depends only on the present state of the system. A related concept is that of Markov chains [42]: Consider for a moment a system with discrete states $x_i \in \Gamma$, where the state of the system at time t is denoted by X_t. Although the state space is usually multidimensional, for a simpler notation and discussion, $x \in \mathbb{R}$ is assumed in the following. Generalization of the theory to the multi-dimensional case is straight-forward and the presented concepts hold just as well. Starting from some initial condition $X_0 = x_{t_0}$, the conditional probability of the system being in state x_{t_n} at time t_n, that is $X_n = x_{t_n}$ given that $X_1 = x_{t_1}, X_2 = x_{t_2}, \dots, X_{n-1} = x_{t_{n-1}}$, is denoted by

$$P\big(X_n = x_{t_n} \,\big|\, X_{n-1} = x_{t_{n-1}}, X_{n-2} = x_{t_{n-2}}, \dots, X_0 = x_{t_0}\big). \tag{10.1}$$

This probability depends on the probability of the system being in state $x_{t_{n-1}}$ at the preceding time as well as the probability of the system being in state $x_{t_{n-2}}$ at the time two earlier, and in fact on the probabilities of all previous states.

This becomes a Markov process if the conditional probability depends only on the directly previous state, that is

$$P\big(X_n = x_{t_n} \,\big|\, X_{n-1} = x_{t_{n-1}}, \dots, X_0 = x_{t_0}\big) = P\big(X_n = x_{t_n} \,\big|\, X_{n-1} = x_{t_{n-1}}\big), \quad (10.2)$$

and then the corresponding time series of states $X = \{X_0, X_1, \dots, X_t, \dots\}$ is called a Markov chain. This leads to an interpretation of the conditional probability of eq. (10.2) as the transition probability for going from state x_i to state x_j

$$P_{ij} = P\big(x_i \to x_j\big) = P\big(X_{n+1} = x_j | X_n = x_j\big), \quad (10.3)$$

where the transition probabilities are assumed to satisfy

$$P_{ij} \geq 0, \quad (10.4a)$$

$$\sum_j P_{ij} = 1. \quad (10.4b)$$

From this description of a Markov process in discrete time and with discrete states, a process continuous in both can easily be obtained as well [47, 217]. For this, the state space variable $x \in \mathbb{R}$ and time $t \in \mathbb{R}$ are defined to be continuous and the conditional probability of eq. (10.3) is switched for a transition (probability) density ρ defined by

$$P(X_{t+\delta t} \in \Gamma \,|\, X_t = x) = \int_\Gamma \mathrm{d}y\, \rho(y, t + \delta t \,|\, x, t) \quad (10.5)$$

with the normalization

$$\int \mathrm{d}x\, \rho(x, t | x_0, t_0) = 1. \quad (10.6)$$

Assuming the initial condition

$$\rho(x, t_0 | x_0, t_0) = \delta(x - x_0), \quad (10.7)$$

the evolution of this density can be described by the Chapman-Kolmogorov equation

$$\rho(x,t|x_0,t_0) = \int_{\mathbb{R}} \mathrm{d}x_1\, \rho(x,t|x_1,t_1)\rho(x_1,t_1|x_0,t_0). \tag{10.8}$$

This equation states that the probability of going from state x_0 to state x can be obtained by integrating over all intermediate states x_1. The factorization in the integral is possible due to the Markov property of the process.

Kramers-Moyal expansion and Fokker-Planck equation

The next step is to derive the time evolution of the transition density, the so-called Fokker-Planck or forward Kolmogorov equation [217, 218]. Assuming the intermediate time t_1 is close to the final time t in eq. (10.8), that is $\tau = t - t_1$ is small, the moments of the transition density

$$\langle (x-x_1)^n\rangle(t,t_1) = \int \mathrm{d}x\, (x-x_1)^n \rho(x_1,t_1|x,t) \tag{10.9}$$

are then also small. For $n \geq 1$, they can therefore be approximated up to linear order in τ by

$$\langle (x-x_1)^n\rangle(t,t_1) = n! D^{(n)}(x,t)\tau + \mathcal{O}(\tau^2), \tag{10.10}$$

where the $D^{(n)}$ are the so-called Kramers-Moyal coefficients. If the moments exist, the probability density is uniquely determined by them:

$$\rho(x_1,t_1|x,t) = \sum_{n=0}^{\infty} \left(-\frac{\partial}{\partial x}\right)^n \delta(x-x_1) \frac{1}{n!} \langle (x-x_1)^n\rangle(t,t_1). \tag{10.11}$$

Entering this expansion into the Chapman-Kolmogorov equation (10.8) yields

$$\begin{aligned} \rho(x,t|x_0,t_0) &= \int_{\mathbb{R}^d} \mathrm{d}x_1 \sum_{n=0}^{\infty} \left(-\frac{\partial}{\partial x}\right)^n \delta(x-x_1) \frac{1}{n!} \langle (x-x_1)^n\rangle(t,t_1)\rho(x_1,t_1|x_0,t_0) \\ &= \sum_{n=0}^{\infty} \left(-\frac{\partial}{\partial x}\right)^n \frac{1}{n!} \langle (x-x_0)^n\rangle(t,t-\tau)\rho(x,t-\tau|x_0,t_0). \end{aligned} \tag{10.12}$$

To obtain the difference quotient, the $n = 0$ term is substracted and both sides are divided by τ:

$$\begin{aligned} &\frac{\rho(x,t|x_0,t_0) - \rho(x,t-\tau|x_0,t_0)}{\tau} \\ &= \frac{1}{\tau}\sum_{n=1}^{\infty}\left(-\frac{\partial}{\partial x}\right)^n \frac{1}{n!}\langle (x-x_0)^n\rangle(t,t-\tau)\rho(x,t-\tau|x_0,t_0) \end{aligned} \tag{10.13}$$

Taking the limit $\tau \to 0$ and approximating the terms on the right hand side by

$$\langle (x-x_0)^n\rangle(t,t-\tau) = n!D^{(n)}(x,t-\tau)\,\tau + \mathcal{O}(\tau^2) \approx n!D^{(n)}(x,t)\,\tau \tag{10.14}$$

$$\rho(x,t-\tau|x_0,t_0) \approx \rho(x,t|x_0,t_0) \tag{10.15}$$

leads to the generalized Fokker-Planck equation

$$\frac{\partial}{\partial t}\rho(x,t|x_0,t_0) = \sum_{n=1}^{\infty}\left(-\frac{\partial}{\partial x}\right)^n D^{(n)}(x,t)\,\rho(x,t|x_0,t_0) \tag{10.16}$$

$$= \mathcal{L}\rho(x,t|x_0,t_0), \tag{10.17}$$

where the Fokker-Planck operator $\mathcal{L}$ was introduced. For the stochastic processes considered in the following, only the first two Kramers-Moyal coefficients are non-zero, that is the drift $D^{(1)}$ and diffusion $D^{(2)}$ coefficients.

Fokker-Planck equation for overdamped Langevin dynamics

After having derived the generalized Fokker-Planck equation, the special equation for overdamped Langevin dynamics can be given. Setting the physical parameters $\beta = D = 1$ in eq. (2.19), the time evolution of a particle with initial position $\vec{r}(0) = \vec{r}_0$ is given by

$$\mathrm{d}\vec{r}(t) = -\nabla U(\vec{r}(t))\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}\overrightarrow{W}(t), \tag{10.18}$$

where $U(\vec{r})$ is a smooth d-dimensional potential and $\overrightarrow{W}$ is d-dimensional Wiener process, as before. The solution $X = (\vec{r}(t))_{t\geq 0}$ to eq. (10.18) is a Markov Process with a unique stationary distribution, the density of which is given by $\pi(\vec{r}) =$

$C \cdot \mathrm{e}^{-U(\vec{r})}$. Since adding a constant to the potential U does not change the dynamics, the normalizing constant of the stationary density is set to $C = 1$. Then $U(\vec{r}) = -\log \pi(\vec{r})$, which allows to write everything in terms of $\pi(\vec{r})$ in the following. Additionally, the argument $\vec{r}$ will be omitted for functions of the space variable unless it is specifically needed.

The Fokker-Planck operator associated with the Markov process X of the overdamped Langevin dynamics of eq. (10.18) is given by

$$\mathcal{L}f = \Delta f - \nabla \cdot (f \nabla \log \pi) = \nabla \cdot \left(f \nabla \log \frac{f}{\pi} \right) \tag{10.19}$$

for any smooth function f. The transition densities ρ_t (omitting the explicit dependency on the initial state and the current spatial position) then follow the corresponding Fokker-Planck equation

$$\frac{\partial \rho_t}{\partial t} = \mathcal{L}\rho_t, \tag{10.20}$$

with the stationary solution

$$\mathcal{L}\pi = 0. \tag{10.21}$$

The Fokker-Planck equation of particle distributions is also known as the Smoluchowski equation.

10.2 OVERDAMPED LANGEVIN DYNAMICS WITH A BIRTH-DEATH PROCESS

To introduce non-local moves, the Fokker-Planck equation corresponding to overdamped Langevin dynamics (10.18) can be augmented by a non-local term α_π depending on the current transition densities

$$\frac{\partial \rho_t}{\partial t} = \mathcal{L}\rho_t - \alpha_\pi(\rho_t)\rho_t. \tag{10.22}$$

Following the idea from Ref. 30, this term is interpreted as a birth-death term of the form

$$\alpha_\pi(f,g)(x) = \log \frac{f(x)}{\pi(x)} - \int \log\left(\frac{f(y)}{\pi(y)}\right) g(y)\,\mathrm{d}y. \tag{10.23}$$

where $\alpha_\pi(\rho_t) = \alpha_\pi(\rho_t, \rho_t)$ is used for shorter notation. Because $\alpha_\pi(\pi) = 0$, the stationary solution of this birth-death augmented Fokker-Planck equation is still π.

10.2.1 *Interacting particle approach*

The goal is now to obtain a stochastic differential equation that corresponds to the Fokker-Planck-Birth-Death equation (10.22), that means finding a purely probabilistic object which is to eq. (10.22) what the overdamped Langevin dynamics of eq. (10.18) is to the corresponding Fokker-Planck equation of eq. (10.20). Therefore, an interacting particle system is introduced similar to the description in Ref. 30. First, eq. (10.22) is approximated by

$$\frac{\partial \rho_t}{\partial t} = \mathcal{L}\rho_t - \Lambda(\rho_t)\rho_t, \tag{10.24}$$

where $\Lambda(\cdot)$ is a smoothened approximation of the birth-death term $\alpha_\pi(\cdot)$. Second, the transition densities are replaced by interacting particles. This is done by choosing a system of N particles with positions $\vec{r}_1(t), \vec{r}_2(t), \dots, \vec{r}_N(t)$ that have the empirical measure

$$\mu_t^N = \frac{1}{N}\sum_{i=1}^{N} \delta_{\vec{r}_i(t)}. \tag{10.25}$$

These particles move independently according to the overdamped Langevin dynamics defined in eq. (10.18) but additionally interact via a birth-death mechanism. The probability of killing or duplicating a particle grows with time, which in practice is handled by independent exponential clocks: Each particle has an independent exponential clock that strikes with the rate $|\Lambda(\mu_t^N)(\vec{r}_i(t))|$

depending on the current configuration of the system. If the exponential clock for the i-th particle strikes at time t, then the particle is either killed or duplicated, depending on the sign of Λ:

- If $\Lambda(\mu_t^N)(\vec{r}_i(t)) > 0$, then the i-th particle is killed and a uniformly chosen other particle is duplicated.
- If $\Lambda(\mu_t^N)(\vec{r}_i(t)) < 0$, then the i-th particle is duplicated and a uniformly chosen other particle is killed.

The empirical measure μ_t^N is a sum of delta distributions and therefore not a smooth function. For the approximation of the birth-death term α_π by Λ, the empirical density has to be smoothened. Convolutions with a centered Gaussian kernel

$$K_{\mathbf{\Sigma}}(x) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{x^\top \mathbf{\Sigma}^{-1} x}{2}\right), \tag{10.26}$$

with the covariance matrix $\mathbf{\Sigma}$ and its determinant $|\mathbf{\Sigma}|$ are considered, where d is the dimensionality of the potential U, as before. Since $\mathbf{\Sigma}$ is assumed to be fixed during simulations, the shorter notation $K = K_{\mathbf{\Sigma}}$ will be used. In the following, the covariance matrix is typically chosen to be diagonal, $\Sigma_{ij} = \delta_{ij}\sigma_i^2$ with $\sigma_i > 0$. In this case, $\boldsymbol{\sigma} = \{\sigma_1, \ldots, \sigma_d\}$ is referred to as the bandwidths of the smoothing kernel.

In Ref. 30, the authors use the approximation

$$\Lambda^0(f) = \alpha_\pi(K * f, f), \tag{10.27}$$

where the operator $*$ denotes the convolution. This choice has one crucial shortcoming, because

$$\Lambda^0(\pi) = \alpha_\pi(K * \pi, \pi)(x) = \log \frac{K * \pi(x)}{\pi(x)} - \int \log\left(\frac{K * \pi(y)}{\pi(y)}\right) \pi(y)\, \mathrm{d}y \neq 0, \tag{10.28}$$

which means that π is not a stationary solution of the corresponding approximation of the Fokker-Planck-Birth-Death equation when entering Λ^0 into eq. (10.24).

Therefore, two alternative approximations shall be proposed here. The first is obtained by adding a correction term to Λ^0:

$$\Lambda^{\mathrm{ad}}(f) = \Lambda^0(f) - \Lambda^0(\pi) = \alpha_\pi(K * f, f) - \alpha_\pi(K * \pi, \pi). \tag{10.29}$$

For this choice, $\Lambda^{\mathrm{ad}}(\pi) = 0$, and the stationary solution π is recovered. The second option is to do a "multiplicative" correction of Λ^0 by also convoluting the stationary distribution:

$$\Lambda^{\mathrm{mu}}(f) = \alpha_{K*\pi}(K * f, f) \tag{10.30}$$

Again, $\Lambda^{\mathrm{mu}}(\pi) = 0$ and π is a stationary solution of eq. (10.24) when using Λ^{mu}.

In the limit of Gaussian kernels with zero width, that is $\mathbf{\Sigma} \to 0$, the kernel can be interpreted as the Dirac delta. Then all three approximations, Λ^0, Λ^{ad}, and Λ^{mu}, coincide with the original birth-death term α_π.

10.2.2 *Convergence behavior*

Although the details are out of the scope of this thesis, the convergence behavior of the proposed interacting particle system has been investigated with mathematical methods. Two properties could be established when using the approximation Λ^{mu}: First, it can be shown that for fixed times $t > 0$, the empirical measure μ_t^N converges weakly to the solution ρ_t of eq. (10.24), if the number N of particles tends to infinity. Secondly, under reasonable assumptions the limiting distribution ρ_t and the equilibrium distribution π coincide, as the time t goes to infinity.

This means that in the limit of infinitely many particles and long simulation times, the equilibrium distribution π of the system is obtained. The equilibrium distribution can therefore be sampled with the proposed birth-death augmented Langevin dynamics approach. The corresponding mathematical proofs of convergence are currently under preparation for publication [219] together with the numerical analysis that is presented in the following chapters.

These proofs were carried out for the approximation Λ^{mu} because it turned out to be mathematically convenient. Similar properties were established in Ref. 30 for the approximation Λ^{0}. Equivalent properties likely also exist for Λ^{ad}, but are currently not theoretically proven. Nevertheless, all approximations will be considered in the following numerical studies.

10.3 GENERAL LANGEVIN DYNAMICS WITH A BIRTH-DEATH PROCESS

Examining the behavior of the birth-death term for overdamped Langevin dynamics makes it feasible to mathematically prove convergence to the right distribution. For accurate simulations of physical and chemical systems, often more general dynamics are required that take inertia into account and thus have to track not only the particle's position $\vec{r}(t)$, but also its momentum $\vec{p}(t)$. This is described by the general (or underdamped) Langevin equations

$$\mathrm{d}\vec{r}(t) = \frac{\vec{p}(t)}{m}\,\mathrm{d}t, \tag{10.31a}$$

$$\mathrm{d}\vec{p}(t) = -\nabla U(\vec{r}(t))\,\mathrm{d}t - \gamma\vec{p}(t)\,\mathrm{d}t + \sqrt{\frac{2m\gamma}{\beta}}\,\mathrm{d}\vec{W}(t), \tag{10.31b}$$

that were already given in eq. (2.16). As before, m denotes the particle mass, γ is a friction constant, and β^{-1} is the thermal energy.

The solution $(X, Y) = (\vec{r}(t), \vec{p}(t))_{t\geq 0}$ of eq. (10.31) is a $2d$-dimensional Markov process that has a unique invariant distribution Π. For a sufficiently damped system, the relevant time scales of the momentum dissipation are much shorter than the ones related to the spatial position of the particle. Then in equilibrium position and momentum become independent and the marginal of the distribution Π with respect to the position $\vec{r}$ coincides with π. If this holds, it is justified to introduce a birth-death mechanism by following the approach for the birth-death augmented overdamped Langevin dynamics in the last section and still use the same birth-death term $\Lambda(\mu_t^N)(\vec{r}_i(t))$ that depends only on the positions and ignores the momenta. This means that if the i-th particle is killed (or duplicated), the entire tuple $(\vec{r}_i, \vec{p}_i)$ is killed (or duplicated). While this assumption of independent momentum and position distributions is taken without further conditions here, it is investigated in numerical simulations in Section 12.2.2 of the next chapter. There, it will be verified that the birth-death method can also be used to successfully sample the equilibrium distribution π in the general Langevin dynamics case.

11

IMPLEMENTATION OF BIRTH-DEATH AUGMENTED SAMPLING

11.1 EXPLICIT EXPRESSIONS FOR THE BIRTH-DEATH TERMS

For the particle-based view that was presented in Section 10.2.1, the mathematical expression of the birth-death term can be derived explicitly. Entering eq. (10.23) into eq. (10.30) yields

$$\begin{aligned}\Lambda^{\mathrm{mu}}(f)(x) &= \log K * f(x) - \log K * \pi(x) \\ &\quad - \int \big(\log K * f(y) - \log K * \pi(y)\big) f(y)\,\mathrm{d}y.\end{aligned} \tag{11.1}$$

When plugging the empirical measure of the N particles

$$\mu_t^N = \frac{1}{N}\sum_{k=1}^{N} \delta_{\vec{r}_k(t)} \tag{11.2}$$

as f into Λ^{mu} of eq. (11.1), the convolution of the delta function with the kernel has to be calculated. Because

$$K * \delta_y(x) = \int K(x-z)\delta_y(z)dz = K(x-y) \tag{11.3}$$

holds, eq. (11.1) can be rewritten as

$$\begin{aligned}\Lambda^{\mathrm{mu}}(\mu_t^N)(\vec{r}_i) = \log\left[\frac{1}{N}\sum_{j=1}^{N} K(\vec{r}_i - \vec{r}_j)\right] - \log(K * \pi)(\vec{r}_i) \\ - \frac{1}{N}\sum_{k=1}^{N}\left\{\log\left[\frac{1}{N}\sum_{j=1}^{N} K(\vec{r}_k - \vec{r}_j)\right] - \log(K * \pi)(\vec{r}_k)\right\}.\end{aligned} \tag{11.4}$$

where the time dependence of the particle positions $\vec{r}_1(t), \vec{r}_2(t), \dots, \vec{r}_N(t)$ was dropped for notational convenience. Also, in the sequel $\Lambda_i^{\mathrm{mu}} := \Lambda^{\mathrm{mu}}(\mu_t^N)(\vec{r}_i)$ will be written for short. Expressions for the other approximations can be derived similarly:

$$\begin{aligned}\Lambda^{0}(\mu_t^N)(\vec{r}_i) = \log\left[\frac{1}{N}\sum_{j=1}^{N} K(\vec{r}_i - \vec{r}_j)\right] - \log(\pi(\vec{r}_i)) \\ - \frac{1}{N}\sum_{k=1}^{N}\left\{\log\left[\frac{1}{N}\sum_{j=1}^{N} K(\vec{r}_k - \vec{r}_j)\right] - \log(\pi(\vec{r}_k))\right\}\end{aligned} \tag{11.5}$$

$$\begin{aligned}\Lambda^{\mathrm{ad}}(\mu_t^N)(\vec{r}_i) &= \log\left[\frac{1}{N}\sum_{j=1}^{N} K(\vec{r}_i - \vec{r}_j)\right] - \log(\pi(\vec{r}_i)) \\ &\quad - \frac{1}{N}\sum_{k=1}^{N}\left\{\log\left[\frac{1}{N}\sum_{j=1}^{N} K(\vec{r}_k - \vec{r}_j)\right] - \log(\pi(\vec{r}_k))\right\} \\ &\quad - \log(K * \pi)(\vec{r}_i) + \log(\pi(\vec{r}_i)) \\ &\quad + \int \Big[\log(K * \pi)(\vec{r}) - \log(\pi(\vec{r}))\Big]\pi \, \mathrm{d}\vec{r} \\ &= \log\left[\frac{1}{N}\sum_{j=1}^{N} K(\vec{r}_i - \vec{r}_j)\right] - \log(K * \pi)(\vec{r}_i) \\ &\quad - \frac{1}{N}\sum_{k=1}^{N}\left\{\log\left[\frac{1}{N}\sum_{j=1}^{N} K(\vec{r}_k - \vec{r}_j)\right] - \log(\pi(\vec{r}_k))\right\} \\ &\quad + \int \Big[\log(K * \pi)(\vec{r}) - \log(\pi(\vec{r}))\Big]\pi \, \mathrm{d}\vec{r}\end{aligned} \tag{11.6}$$

In all following numerical simulations, the covariance matrices of the Gaussian kernel are chosen as diagonal, $\Sigma_{ij} = \delta_{ij}\sigma_i^2$ with the bandwidths $\boldsymbol{\sigma} = \{\sigma_1, \dots, \sigma_d\}$ for the d dimensions of the potential $U(\vec{r})$ satisfying $\sigma_i > 0$. Then eq. (10.26) turns into

$$K(\vec{r}) = K_{\boldsymbol{\sigma}}(\vec{r}) = \frac{1}{(2\pi)^{d/2} \prod_{i=1}^{d} \sigma_i} \exp\left(- \sum_{i=1}^{d} \left(\frac{x_i}{\sqrt{2}\sigma_i} \right)^2 \right), \tag{11.7}$$

where $\vec{r} = \{x_1, \dots, x_d\}$.

11.2 CALCULATION OF THE BIRTH-DEATH PROBABILITIES

The used algorithm mostly follows Algorithm 1 of Ref. 30 but with modifications to make it less computationally involving and the mentioned different approximations of the birth-death term.

In the algorithm, the exponential clock associated with each particle is replaced with the probability of performing a birth or death event at a specific time, which will be called birth-death probability q. After the calculation at fixed times of the simulation, the probability q is subsequently evaluated against a random number z drawn uniformly from $[0, 1)$ and the associated event is executed in case of success (i.e., $z \leq q$). Assuming that a time interval of length θ has passed since the last evaluation, the birth-death probability of particle i is

$$q_i = 1 - \exp\left(-|\Lambda_i|\theta\right). \tag{11.8}$$

In Ref. 30, it was proposed to attempt birth-death events after every Langevin time step δt, that is $\theta = \delta t$. While this choice works, it is computationally more efficient to attempt birth-death events less often. Therefore, the time interval between subsequent attempts is generalized to be $\theta = M\delta t$ with $M \in \mathbb{N}$, that is, the birth-death probabilities are calculated every M Langevin time steps. The birth-death probabilities become

$$q_i = 1 - \exp\left(-|\Lambda_i| M\delta t\right). \tag{11.9}$$

with an integer parameter M that can be varied freely. This parameter will be refered to as birth-death stride in the following. The Λ values, which require the computationally involving density estimate ρ_t, are calculated less often with increasing M. For $M > 1$, the calculations are less often than in the original proposal of Ref. 30. Additionally, the longer time between interactions of the different particles is beneficial for implementations that parallelize the computation of the individual replicas of the system. On top of the overhead from the computation itself, all replicas need to be time synced when attempting the birth-death interactions. It will be tested in Section 12.1.4 if using longer times between birth-death attempts results in deviating sampling.

The original algorithm in Ref. 30 iterates over the particles, where it individually calculates the birth-death probability and executes successful events immediately. The values Λ_i have therefore to be calculated for each particle individually, or at least recalculated from the new positions after each successful birth-death event. Differently, in the newly proposed algorithm all birth-death rates Λ_i are calculated only once from the positions before the birth-death step. Only the order in which the birth-death events are applied is randomized. No disadvantages could be found from this approach as long as the probabilities of birth-death events remain low. In Section 12.1.4, the case of high birth-death probabilities will be examined more closely.

11.3 DESCRIPTION OF THE ALGORITHM

As described in Section 10.3, the same birth-death method is applied to both overdamped and general Langevin dynamics. The algorithm was therefore implemented in a custom python code, together with Langevin solvers. The Euler-Maruyama scheme [69, 70] was chosen for overdamped Langevin dynamics, while the Bussi-Parinello scheme [48] was chosen for the general Langevin dynamics case. The code performs independent Langevin dynamics of particles in artificial potentials and couples the particles via birth-death steps at user-specified intervals. A description of the steps of the algorithm is provided in Algorithm 1.

The code is published under the LGPL license and can be found in a public Github repository (`https://github.com/bpampel/bdld`). Version v0.3.1 [220] was used for all numerical simulations that are presented in the following.

Algorithm 1: Birth-Death augmented Langevin dynamics
The algorithm remains the same regardless of the Langevin dynamics solver L or the birth-death approximation Λ chosen.

```
Input: Potential $U$ corresponding to the equilibrium distribution $\pi$
       $N$ particles with initial positions $X = \{\vec{r}_i\}_{i=1}^N$
           and momenta $P = \{\vec{p}_i\}_{i=1}^N$
       Langevin solver $L(X, P, U, \delta t)$ with corresponding parameters
       number of Langevin time steps $T$
       Langevin time step $\delta t$
       number of Langevin time steps between birth-death attempts $M$
       smoothing Kernel $K$
Output: A set of particles whose empirical measure approximates $\pi$

$m \leftarrow 0$                  // Langevin steps since birth-death step
for $t \leftarrow 1$ to $T$ do
    update $X$ and $P$ by Langevin solver $L(X, P, U, \delta t)$
    $m \leftarrow m + 1$
    if $m = M$ then
        Calculate $\Lambda$ for all particles
        Draw $N$ random numbers $\{z_i\}_{i=1}^N$ uniformly from $[0, 1)$
        Make list $\chi$ of indices $i$ for which $z_i \leq 1 - \exp(-|\Lambda_i| M \delta t)$
        Shuffle $\chi$ randomly
        foreach $i \in \chi$ do
            Select particle $j$ uniformly from all other particles
            if $\Lambda_i > 0$ then
                $\vec{r}_i \leftarrow \vec{r}_j$; $\vec{p}_i \leftarrow \vec{p}_j$
            else if $\Lambda_i < 0$ then
                $\vec{r}_j \leftarrow \vec{r}_i$; $\vec{p}_j \leftarrow \vec{p}_i$
            end if
        end foreach
        $m \leftarrow 0$
    end if
end for
```

12

SIMULATIONS WITH BIRTH-DEATH PROCESSES

In this chapter, the proposed Algorithm 1, which was presented in the last chapter, is verified through numerical simulations. Also, the different options that were proposed for the birth-death approximation Λ are investigated. Additionally, the influence of the input parameters (number of particles N, bandwidth of the kernel K, stride between birth-death calculations M) is examined by varying them systematically. As test cases, simulations of the movement of sets of particles in different artificial potentials are chosen. The focus is on prototypical rare-event systems, where many applications are envisioned that can benefit from the method.

After the basic algorithm has been verified with overdamped Langevin dynamics, Section 12.2 assesses the behavior of the algorithm for the general Langevin case, which provides an important step towards real-life applications. The input files and data supporting the results of this chapter are openly available at Zenodo [221] (DOI: `10.5281/zenodo.5873265`).

12.1 OVERDAMPED LANGEVIN DYNAMICS

12.1.1 *Comparison of approximations* Λ

Instead of going directly to rare-event systems, the behavior when using the different approximations Λ shall be in the foreground first. Therefore, a system with a moderate barrier height is used beforehand, such that transitions are also observed within moderate simulation time by pure Langevin sampling. Overdamped Langevin dynamics sampling (with $\beta = D = 1$) of a one-dimensional double-well system is chosen, that is described by the mathematical expression

$$U(x) = x^4 - 4x^2 + 0.2x, \tag{12.1}$$

and a plot of which can be seen as black lines in Figure 12.1 or in Figure 6.2. This is the same potential that was already described in in eq. (7.1) and simulated with the VES method in Section 7.1.

It features two metastable states at the local minima, at $x_L \approx -1.4$ and $x_R \approx 1.4$, separated by a barrier of about 4.3 energy units from the left state. The barrier is centered at the local maximum at $x_0 \approx 0$. In equilibrium, the probability of a particle being in the basin of attraction $B_L : x \in (-\infty, x_0)$ of the lower energy metastable state x_L on the left is $\pi(B_L) = \int_{-\infty}^{x_0} \pi(x)\, \mathrm{d}x \approx 0.63$. Similarly, $\pi(B_R) = \pi((x_0, \infty)) \approx 0.37$ is the probability for the higher energy metastable state x_R.

First, the influence of the newly proposed approximations Λ^{mu} and Λ^{ad} on the sampling is tested and compared to the approximation Λ^0. For this, 100 particles are used and an initial distribution far from equilibrium is chosen: Only 10 particles are placed in the more likely state at x_L, while the remaining 90 particles start in the less likely state at x_R. The overdamped Langevin solver with a time step of $\delta t = 0.001$ is used for in total 2,000,000 steps. The number of steps between birth-death attempts is fixed to $M = 100$, while both the approximation Λ as well as different kernel widths σ are varied between the simulations.

To assess the correctness of the sampling, estimates of the free energy are constructed via histogramming. While omitting the first 100,000 time steps, the positions of all particles are collected into a histogram H with 200 equally sized bins in the range $x \in [-2.5, 2.5]$. At the end of the simulations the free energy is calculated via eq. (3.46),

$$F(x) = -\beta^{-1} \log H(x) + C, \tag{12.2}$$

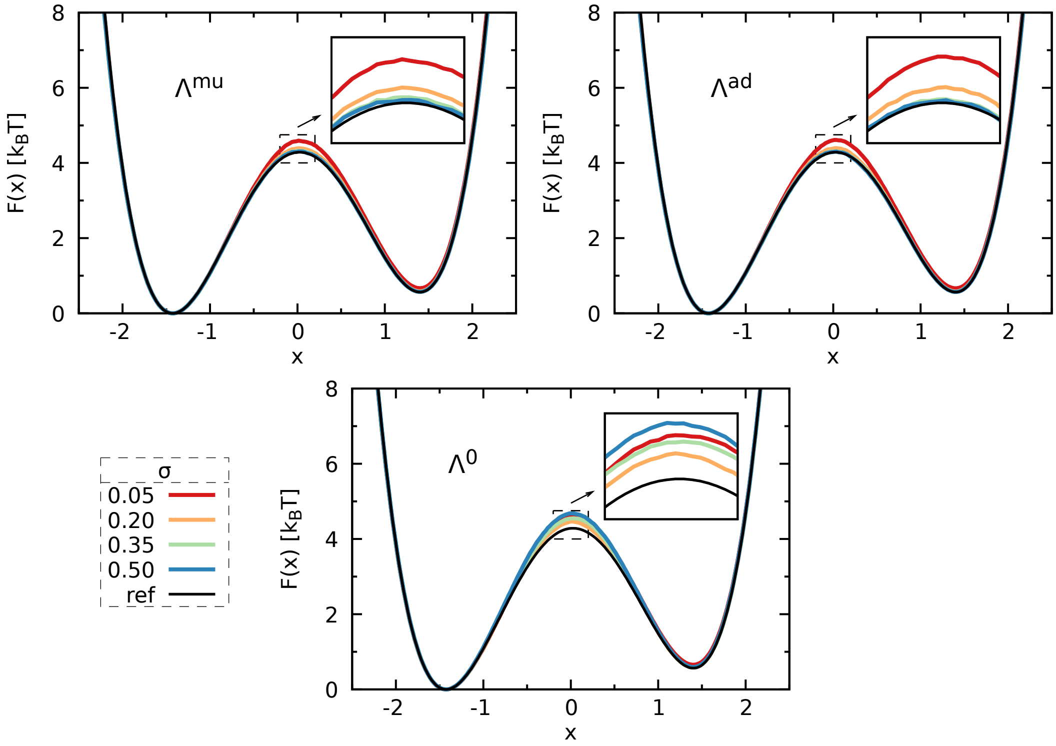


Figure 12.1: Comparison of different approximations Λ via the obtained FES estimates. Each plot shows estimates obtained after 2,000,000 steps of simulation using different values of the kernel bandwidth σ. The black line denotes the reference obtained directly from the potential.

where the constant C is chosen such that $\min F(x) = 0$. Results for selected kernel bandwidths σ can be seen in Figure 12.1, together with the reference from the potential.

Although all obtained FES qualitatively agree with the reference and sample the basins correctly, deviation from the reference FES is observed in the barrier region for some simulations. Therefore, the height of the barrier when starting from the left state is estimated by the highest value of the FES in the respective region (x_L, x_R). The obtained values for all simulations are shown in the left panel of Figure 12.2 as a function of the kernel width σ together with the value from the reference.

To numerically assess the correctness of the sampling, the Kullback-Leibler divergence $D_{\mathrm{KL}}(\pi \,|\, \eta)$ is used, which was defined in eq. (3.50). As reference,

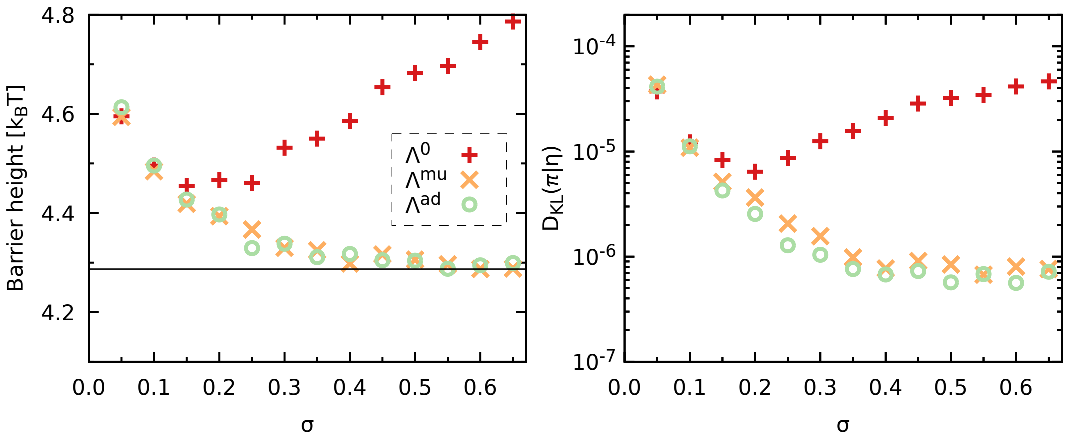


Figure 12.2: Performance comparison of the different approximations Λ.
Left panel: Height of barrier from left basin as a function of the kernel width σ for the three approximations. The reference from the potential is shown as black horizontal line.
Right panel: Kullback-Leibler divergence between the obtained probability distributions and the reference distribution from the potential.

the equilibrium probability distribution π is calculated from the potential. Then the estimated distribution η, obtained from normalizing the histogram H of the simulations, is plugged into eq. (3.50) together with the reference. The resulting values for all simulations can be seen in the right panel of Figure 12.2.

Looking at the FES estimates in Figure 12.1, it is observed that although all simulations give a correct estimate for the low-energy regions, the barrier region is not always sampled correctly. For the original proposal Λ^0, the barrier region is always undersampled compared to the reference, which results in a higher estimate of the barrier as it can be seen in the left panel of Figure 12.2. On the other hand, for both new proposals Λ^{mu} and Λ^{ad} this occurs only if the bandwidth of the kernel is small. The kernel density estimate of eqs. (11.4–11.6) is likely responsible for this effect, as a very small bandwidth results in a very spiky density. The lowest value of σ which results in correct sampling, σ_{crit}, depends on the system and the used number of particles, which will be investigated further in Sections 12.1.3 and 12.2.3. Similar conclusions about the behavior of the different approximations can be drawn when looking at the Kullback-Leibler divergence in the right panel of Figure 12.2: All simulations with the approximation Λ^0 yield worse estimates of the true equilibrium distribution (a larger value of the KL divergence) when compared to the new approximations Λ^{mu} and Λ^{ad}. The slightly lower values for Λ^{ad} compared to Λ^{mu} are a coincidence that comes from using only a single simulation per data point. The same simulations were repeated with different initial seeds of the random number generator. The results from different runs show fluctuations on the same order as the differences visible here and no clear trend which of the two approximations is better. In the following, the discussion will therefore be focussed on only one approximation. The approximation Λ^{mu} is chosen because it provides better properties for mathematical analysis while showing the same behavior in numerical simulations as Λ^{ad}. All presented data will use this approximation unless explicitly stated otherwise.

The performance of the different approximations is further examined by evaluating how quickly the birth-death algorithm manages to distribute the particles in the two basins B_L and B_R according to the equilibrium distribution. Figure 12.3 shows the number N_L of particles in the basin B_L of the lower energy state divided by the total number N of particles for the initial part of the simulation. Additionally, the data of a simulation without the birth-death steps is shown, which is a pure overdamped Langevin dynamics simulation with the same parameters. Clearly, the simulation without the birth-death steps tends only slowly towards the equilibrium value ($\pi(B_L) \approx 0.63$). The moderate barrier of the system allows for transitions from the Langevin dynamics within the simulation time, although the respective time scale of transition is long. In contrast, all simulations that employ the birth-death scheme quickly approach the equilibrium value of the particle distribution and reach it within only a few thousand Langevin time steps. After the initial phase, only small fluctuations around the equilibrium value can be seen. No difference in equilibration speed is visible between the birth-death simulations. The aforementioned differences to equilibrium sampling for the approximation Λ^0 and small values of the kernel width σ are therefore verified to mostly affect the barrier region, and are not significant for the low lying regions of the system that are most relevant for the number of particles in each state.

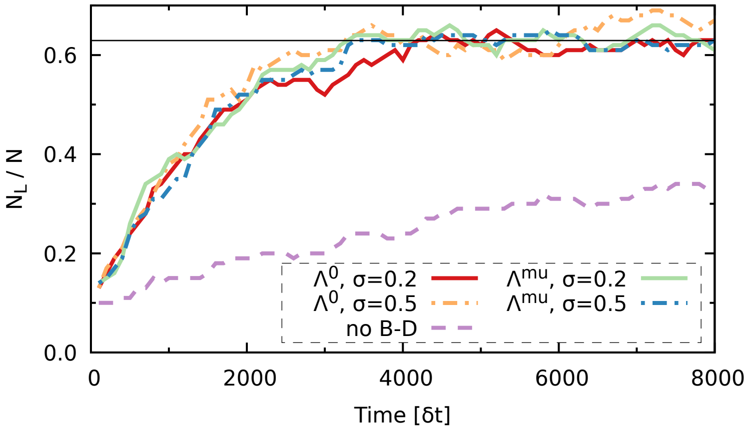


Figure 12.3: Fraction of particles in the left basin for different approximations Λ during the initial part of the simulations. For both Λ^0 and Λ^{mu}, two simulations with different kernel widths are given. Solid lines denote a kernel width $\sigma = 0.2$, while dotted and dashed lines denote $\sigma = 0.5$. The dashed purple line denotes a Langevin dynamics simulation without birth-death steps. The black horizontal line shows the equilibrium value $\pi(B_L) \approx 0.63$ from the potential.

12.1.2 *Behavior for large kernel bandwidths*

In the last section, it was observed that for Λ^{mu} and Λ^{ad} good sampling is obtained as long as the bandwidth σ of the kernel is chosen large enough. Of interest is, what happens when the bandwidth is increased further or even in the limit $\sigma \to \infty$. This will be deduced theoretically and then verified via simulations.

In the limit of infinite width, the Gaussian kernel of eq. (10.26) converges pointwise to a constant. Because the probability densities f used in the birth-death scheme are normalized,

$$\lim_{\sigma\to\infty} \frac{K_\sigma * f}{K_\sigma} = \lim_{\sigma\to\infty} \int_{-\infty}^{\infty} \mathrm{d}y \, \frac{K_\sigma(x-y)}{K_\sigma(x)} f(y) = \int_{-\infty}^{\infty} \mathrm{d}y \, f(y) = 1 \tag{12.3}$$

holds, where dominated convergence was used to switch the integral and limit. Then

$$\lim_{\sigma\to\infty} \frac{K_\sigma * f}{K_\sigma * \pi} = \lim_{\sigma\to\infty} \frac{K_\sigma * f}{K_\sigma} \frac{K_\sigma}{K_\sigma * \pi} = 1 \tag{12.4}$$

follows, which can be entered into the approximation Λ^{mu} of eq. (10.30) to yield

$$\lim_{\sigma \to \infty} \Lambda^{\mathrm{mu}}(f) = 0. \tag{12.5}$$

This means that the probabilities of birth-death events become zero in the limit of infinitely large kernel widths, or more casually speaking, that the birth-death term is gradually turned off for large kernel widths.

To demonstrate this effect, simulations with the overdamped Langevin solver are performed for the one-dimensional double-well potential defined in eq. (12.1). The simulation protocol is kept the same as in Section 12.1.1: 100 particles are used with an initial distribution of 10 in the left state at x_L and 90 at the right minimum x_R. A total of 2,000,000 Langevin steps are run with a time step of $\delta t = 0.001$. Every 100 Langevin steps, the birth-death probabilities are calculated via the approximation Λ^{mu}, and the respective accepted events are executed. The width of the Gaussian kernels is varied in the range $\sigma \in [0.5, 5]$ in steps of 0.5. Additionally, one simulation without the birth-death algorithm is performed. The resulting distribution of the particles in the two states, as well as the percentage of accepted birth-death events is presented in Figure 12.4. The rate of accepted birth-death events P_{acc} is calculated by counting the number of executed birth-death steps during the simulation and dividing it by the total number of attempted birth-death moves.

Looking at the time evolution of the particle distribution, it can be observed that with increasing bandwidth σ the convergence towards the equilibrium distribution takes more and more time. For the largest values of σ, the evolution gradually approaches the behavior of the simulation without birth-death events, as predicted by the theory. It is of note that the same set of random numbers was used for the noise term of the Langevin solver in all simulations to reduce differences from the Langevin dynamics, which is visible in the similarity of the time evolution. Looking at the percentage of accepted birth-death events in the right panel of Figure 12.4, a drastic reduction of about one order of magnitude is observed between the smallest and largest chosen value of σ. This exponential decrease confirms the assertion that large bandwidths can be used to gradually turn off the birth-death algorithm. This can also be explained intuitively: In the

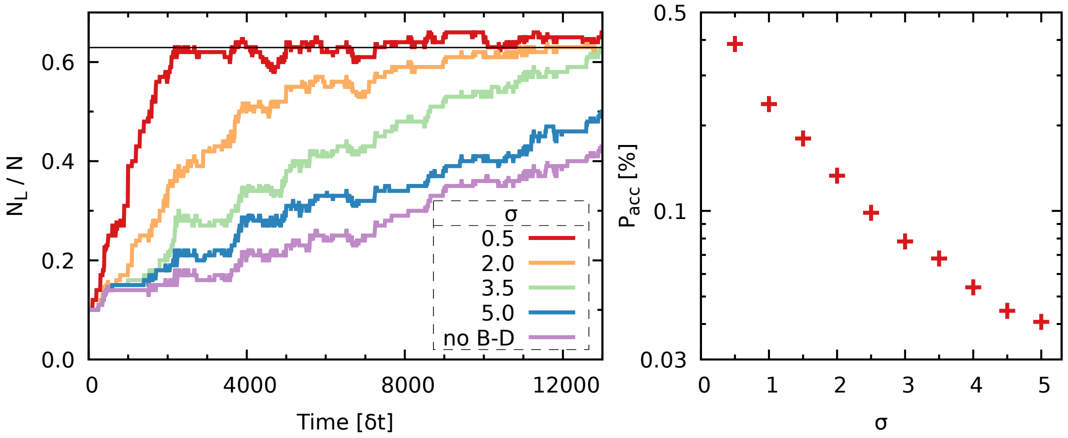


Figure 12.4: Effect of choosing large bandwidths on the performance of the birth-death algorithm.
Left panel: Number of particles in the left state as a function of simulation time for the different bandwidths σ as well as one simulation without birth-death events. Shown are only the first 13,000 time steps. The solid black line is the equilibrium value calculated from the potentials.
Right panel: Percentage of accepted birth-death events from the total number of birth-death events.

limit of infinite width, the Gaussians locally resemble a uniform distribution, so that the values of the first and second terms in eq. (11.4) are independent of the particle position. Since the third term is the average of these two terms over all particles, the birth-death rates become zero. For the presented algorithm, this means that the birth-death part of the dynamics can be gradually turned off by increasing the bandwidth. In the limit of infinitely wide kernels, pure Langevin sampling is performed.

Looking at the other approximations, the picture is less clear: Starting with Λ^{ad}, the limit $\sigma \to \infty$ does not give the same result as for Λ^{mu}. While the first term of the definition

$$\begin{aligned} \Lambda^{\mathrm{ad}}_{\pi}(f) = \log \frac{K_\Sigma * f}{K_\Sigma * \pi} - \int \left(\log \frac{K_\Sigma * f}{\pi} \right) f \, \mathrm{d}x \\ + \int \left(\log \frac{K_\Sigma * \pi}{\pi} \right) \pi \, \mathrm{d}x, \end{aligned} \tag{12.6}$$

also goes to zero for $\sigma \to \infty$ due to eq. (12.4), the remaining terms become in general only independent of the position x. The birth-death term becomes not necessarily zero, but it is still constant over the range. Hence, even though the birth-death mechanism may not be turned off entirely in the limit, it no longer distinguishes between the different particles and does birth-death events randomly.

For the approximation Λ^0, the limit $\sigma \to \infty$ does not yield a constant over the range or even zero but continues to be position-dependent.

In summary, when using the approximation Λ^{mu} the effect of the birth-death term gradually reduces with increasing kernel bandwidth. This effect seems to be only significant for kernel widths that are clearly too large, such as the unreasonabe $\sigma = 5$ (compare also Figure 12.3 for results with different smaller widths). In practice, σ should therefore be chosen large enough to get a smooth density estimate for correct sampling, while keeping in mind that too large values reduce the effectiveness of the method. For the other approximations, the behavior in the limit of large kernel widths is more complex, which makes usage of Λ^{mu} favorable.

12.1.3 *Influence of the number of particles N on the critical bandwidth* σ_{crit}

Up to this point, only results for simulations with 100 particles were shown. It was already found that there is a critical kernel bandwidth σ_{crit} below which the sampling is no longer accurate in the high-energy regions. The dependence of σ_{crit} on the potential is hard to investigate systematically, but the influence of the number of particles for a fixed system shall be investigated in the following. The protocol of Section 12.1.1 is used to perform additional simulations with 10, 30, 300, and 1000 particles. The simulation parameters are kept the same as before: The particles were placed in the two states of the 1D potential of eq. (12.1) with a relative distribution of 0.1 to 0.9, as before. The number of Langevin steps per particle is adjusted in the simulations to always have a total number of 200,000,000, matching the previous simulations with 100 particles. The time step is set to $\delta t = 0.001$. Every 100 Langevin steps, the birth-death probabilities are

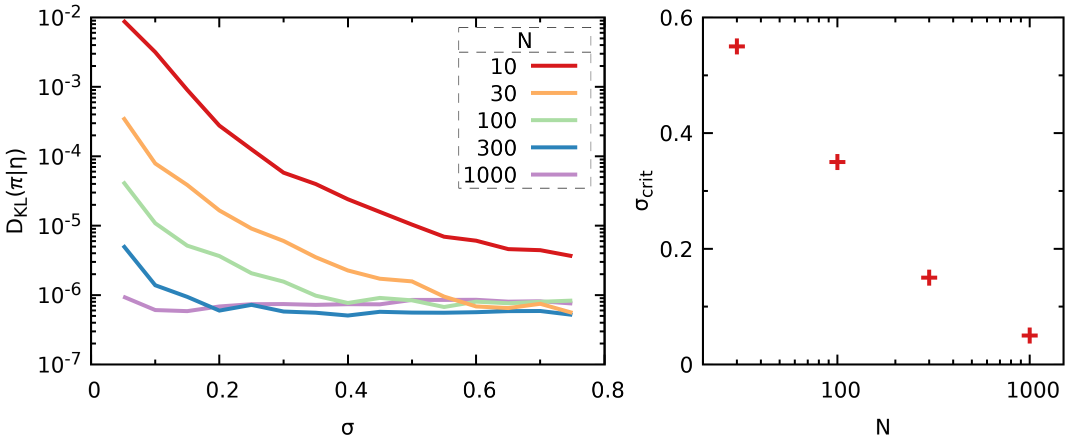


Figure 12.5: Influence of the number of particles N on the critical bandwidth σ_{crit}.
Left panel: Kullback-Leibler divergences from the estimated probability distribution to the equilibrium distribution for simulations with different number of particles N and bandwidths σ.
Right panel: Value of the critical bandwidth σ_{crit} defined as the lowest value of σ for which $D_{KL}(\pi|\eta) \leq 10^{-6}$ for the simulation series given in the left panel. The value for 10 particles is missing as the criterion is not met in the investigated range.

calculated via the approximation Λ^{mu}, and the respective accepted events are executed. For each fixed number of particles, the bandwidth is varied in the range $\sigma \in [0.05, 0.75]$ in steps of 0.05.

The Kullback-Leibler divergence $D_{\text{KL}}(\pi|\eta)$ from the obtained sampling distribution η to the true equilibrium distribution π is calculated for all simulations. In the left panel of Figure 12.5, the result for each set of simulations with fixed particle number is shown as a function of the bandwidth σ. The KL divergence decreases with increasing σ for all sets of simulations. At larger σ nearly identical values below 10^{-6} are obtained for all sets except $N = 10$, which means that sampling very close to the equilibrium distribution is achieved. For the present purpose, σ_{crit} is defined as the lowest bandwidth at which the threshold $D_{\text{KL}}(\pi|\eta) \leq 10^{-6}$ is achieved. The right panel of Figure 12.5 shows the values that were obtained with this criterion. For the simulations with just 10 particles, the required sampling accuracy could not be observed within the range of σ under investigation. The value of σ_{crit} decreases with increasing number of particles.

This observed effect of a critical bandwidth can be explained by the mechanism of the density estimation used for the birth-death probabilities. A very small bandwidth and a small number of particles lead to a rather spiky density estimate with sparse isolated peaks at the particle positions, while increasing the bandwidth and the number of particles yields a smoother density estimate.

Taking an exponential fit to the obtained critical bandwidth values shown in the right panel of Figure 12.5 yields approximately $\sigma_{\text{crit}} \propto N^{-1/2}$. An exponential behavior is in line with the literature, where theoretical values for the relationship between the number of data points and the optimal width of the Gaussian kernel used in the density estimation have been deduced [222], assuming a normal distribution of the data.

Concludingly, the parameter σ has to be chosen large enough to get a smooth estimate of the density, where the critical bandwidth decreases with the number of particles used. Further investigation of this behavior is required before an a-priori rule-of-thumb can be given.

12.1.4 *Influence of the birth-death stride M*

Next, the effect of having multiple Langevin dynamics time steps in between birth-death attempts is estimated. To this end, a set of simulations with the same double-well potential as before is performed. The parameter M is varied while keeping the number of particles $N = 100$ and the kernel width $\sigma = 0.4$ of the approximation Λ^{mu} fixed. Starting with a particle distribution of 10 to 90, the protocol and parameters of Section 12.1.1 are used again to obtain FES estimates via histogramming. The resulting FESs are shown in the top left panel of Figure 12.6.

Also, in the right panel of the figure, the time evolution of the particle distribution in the two states is shown, which was obtained the same way as in the previous section. The bottom left part of Figure 12.6 shows two different values as a function of the birth-death stride M: The upper part presents the percentage of accepted birth-death events P_{acc}, given by the number of accepted birth-death moves divided by the total number of attempted ones. The other set

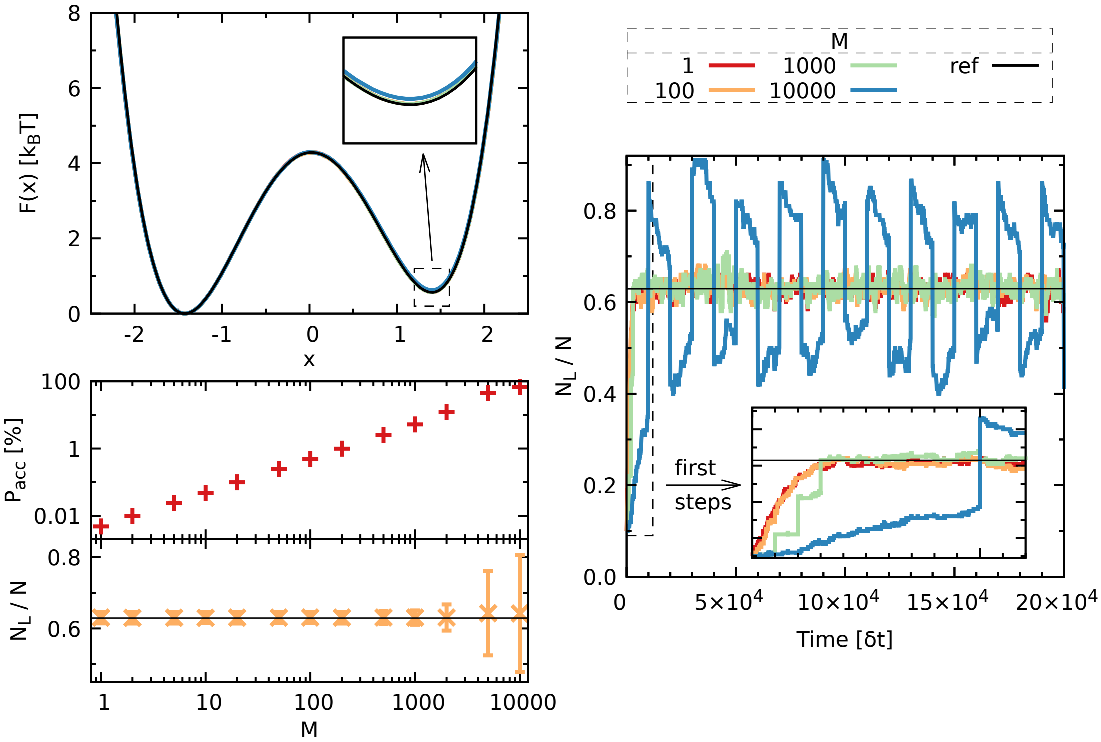


Figure 12.6: Influence of the birth-death stride M.
Top left panel: Estimates of the free energy from sampling by using different values M of Langevin dynamics step between attempted birth-death events. All but the data for $M = 10000$ cannot be distinguished from the reference.
Bottom left panel: Percentage of accepted birth-death events of the total number of birth-death attempts (top) and the time-averaged fraction of particles in the left state (bottom). The error bars of the bottom part denote the standard deviation and the black horizontal line is the expected equilibrium value. The first 50,000 time steps were omitted before averaging over the fraction of particles at all time steps.
Right panel: Fraction of particles in the left state as a function of simulation time. The colored lines are from the same simulations as in the top right panel, the black horizontal line is the expected equilibrium value. The inset shows a magnification of the first 12,000 time steps where each abscissa tic denotes 1,000 steps.

of values shows the time-averaged fraction of particles in the left state $\langle N_L/N \rangle$, together with the respective standard deviation. The first 50,000 Langevin time steps were excluded before averaging over the values at all time steps.

Looking at the top left panel of Figure 12.6, it is observed that all simulations yield a good estimate of the FES, however, the one with $M = 10000$ shows a slight deviation at the minimum of the higher state. The estimate of the equilibrium distribution of the particles in the two states, which can be calculated from the histogram, is slightly off for this simulation: $\eta(B_L) = 0.64$ instead of 0.63 that the reference and all other simulations give. Examining the time evolution of the particle distribution, all simulations with $M < 10000$ reach the correct equilibrium value within the first 4,000 time steps and there are only small fluctuations around this value afterward. A smaller M results in slightly faster equilibration, although this effect is found to be rather small.

For the simulation with $M = 10000$, the birth-death events result in overshooting, such that the number of particles in the left states becomes either too small or too large directly after the events were performed. Between the birth-death events, the slow equilibration process due to the Langevin dynamics can be seen because the moderate barrier makes transitions only rare but not completely unlikely. The overshooting happens because the birth-death probabilities are calculated for all particles at once and then the respective events are performed simultaneously. The time between birth-death calculations enters exponentially in the event probabilities given by eq. (11.9). For large values of M, the event probabilities become very large, in the case of $M = 10000$ around 70 % of the particles are killed or duplicated each time. Because the birth-death probabilities are calculated for all particles at the beginning of the birth-death step, changes of the particle distribution and of the resulting birth-death probabilities after each accepted birth-death event are not taken into account. As long as only a few birth-death events occur, these changes are not significant and only minor differences are observed during the birth-death step of the algorithm.

To verify that the combination of high average birth-death probabilities together with the bulk calculation and execution of the events is indeed responsible for this, an additional simulation with $M = 10000$ is run using the same parameters and seed for the random number generator, but with a slightly modified algorithm: Instead of executing all accepted birth-death events simultaneously, the probabilities are recalculated during the birth-death step whenever an accepted event has been carried out. Hence, the changed particle distribution is taken into account during the course of the birth-death step. The resulting time evolution of the particle distribution can be seen in Figure 12.7, together with the previous data without recalculation of the probabilities.

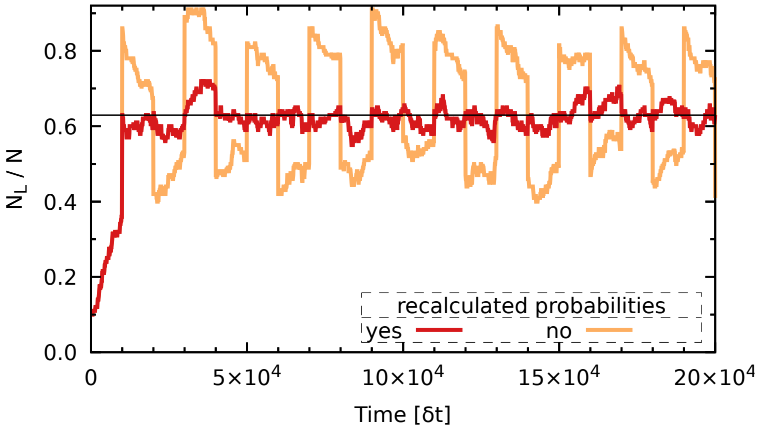


Figure 12.7: Effect of recalculating the birth-death probabilities after each accepted birth-death event.
Shown is the fraction of particles in the left state as a function of simulation time. The data is for $M = 10000$ when recalculating the birth-death probabilities after each accepted event (red) or in bulk (orange), the black horizontal line is the expected equilibrium value. Depicted are only the first 200,000 time steps. The orange line is the same data as for the $M = 10000$ line in the right panel of Figure 12.6.

Looking at the first birth-death step at $t = 10000\,\delta t$, the fraction of particles in the left state was roughly 0.35 immediately before it. This is far from the equilibrium distribution of 0.63 which results in high probabilities of getting killed for the particles in the right state, and high probabilities of getting duplicated for particles in the left state. When applying the birth-death moves simultaneously for all particles without recalculation of the probabilities in between, this results in overshooting, such that too many particles transfer to the left state. On subsequent birth-death steps, the same effect is observed, which results in large fluctuations around the mean value. For the simulation where the events are applied one at a time and the probabilities are recalculated in case of acceptance, this overshooting effect does not occur and the birth-death step results in particle distributions close to the equilibrium. To assess the change numerically, the rate of accepted birth-death events P_{acc} and the standard deviation of the fraction of particles in the left states is calculated for this simulation as well. The average birth-death probability decreases significantly from $P_{\text{acc}} = 0.68$ to $P_{\text{acc}} = 0.21$, and the standard deviation as a measure of the fluctuation of N_L/N shrinks from 16.4 to 3.3.

The per-particle approach with recalculation of the probabilities after each accepted event thus solves the problem of overshooting but requires a lot more computational effort. Because the lower effort is the reason for not calculating the probabilities at every Langevin timestep in the first place, it is concluded that as long as birth-death events remain relatively rare, performing multiple Langevin steps between birth-death attempts helps to speed up simulations without negative side effects. To quantify this for the given system, another look at the lower left panel of Figure 12.6 is taken: Significant changes in the average particle distribution are observed only for $M > 1000$, which corresponds to $P_{\text{acc}} \gtrapprox 5\,\%$. Monitoring this probability during the simulation provides a simple way to ensure that the scheme of Algorithm 1 can be used in the proposed form.

12.2 GENERAL LANGEVIN DYNAMICS

12.2.1 *Dependency of the speed of equlibration on the barrier height*

After investigating the influence of the parameters of the birth-death algorithm, the behavior in the general Langevin case is evaluated. To simultaneously assess the speed of convergence for different barrier heights, the double-well potential of eq. (12.1) is generalized, using two parameters a and b:

$$U(x) = ax^4 - 4ax^2 + bx \tag{12.7}$$

In Table 12.1, sets of coefficients are given where a is systematically increased and then b is set such that the equilibrium distribution of the particles in the two states remains fixed at about $\pi(B_L) \approx 0.63$ to $\pi(B_R) \approx 0.37$, as it was in Section 12.1.1. This results in double-well potentials with similar properties of the basins, but systematically increased barrier height F_B. In the remainder of this section, the different potentials will be referred to by their associated barrier height F_B, where for shorter notation only the integer part is used. For example,

Table 12.1: Coefficients for potentials according to eq. (12.7) with increasing barrier height F_B while keeping the equilibrium distribution of particles in the two states fixed ($\approx 63\,\%$ in lower energy state).

a	b	barrier height
1	0.2	4.285
2	0.1918	8.272
4	0.1889	16.267
8	0.1877	32.262

$F_B = 4$ refers to the potential with $a = 1$ and $b = 0.2$ that was previously described in eq. (12.1).

For each set of coefficients, simulations with a general Langevin integrator, according to the Bussi-Parinello scheme [48] of eq. (2.20), are performed. The parameters of the Langevin dynamics are set to $\beta = 1$, $m = 1$ and $\gamma = 10$. In total 100 particles are simulated for 2,000,000 steps with a time step $\delta t = 0.005$, where again 10 particles are initially placed in the left and 90 in the right basin. Birth-death events are performed after every 100th time step with the approximation Λ^{mu} and a Gaussian kernel with bandwidth $\sigma = 0.5$. For comparison, pure Langevin dynamics simulations without birth-death events but otherwise the same parameters are performed additionally. The obtained FESs from histogramming the samples with the same protocol as in Section 12.1.1 can be seen in Fig. 12.8.

For the simulations with birth-death, the estimated FES agrees well with the reference in all cases, although the higher barrier regions are not sampled due to the insufficient simulation time. On the contrary, it is observed that the pure Langevin dynamics simulations are only sampling the system correctly if the barrier is low. Already for a slightly higher barrier of $F_B = 8$, there is a visible difference in the height of the right minimum, that is, it estimates it to be lower in energy than the reference. For higher barriers, the estimates from pure Langevin sampling are completely off. They give the right state as lower in energy than the left one by several $k_{\mathrm{B}}T$, opposite to the reference FES of the system obtained directly from the potential.

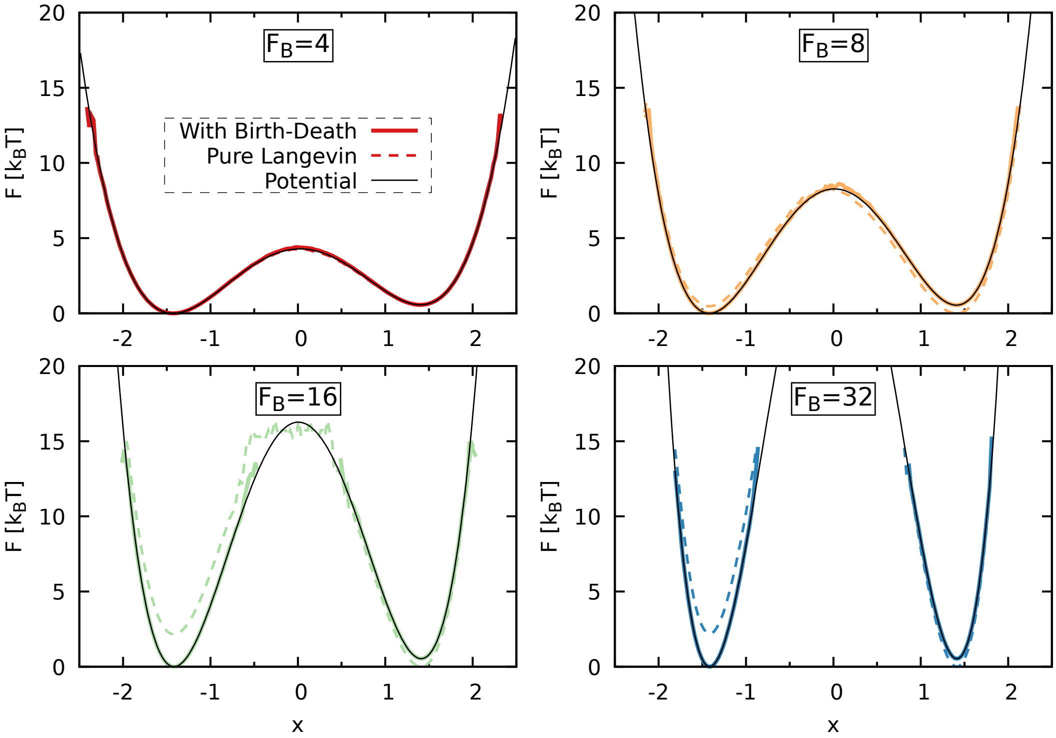


Figure 12.8: FES estimates from general Langevin sampling with and without birth-death moves.
Shown are estimates of the FES from sampling four potentials with different coefficients according to Table 12.1. The solid lines are from simulations with birth-death events, the dashed lines from pure Langevin dynamics simulations with independent particles, and the thin black lines denote the reference from the potential.

The reason for this can be seen when looking at the distribution of the particles in the two states, which is presented in Figure 12.9. With increasing barrier height, crossings between the states by Langevin dynamics become rarer. While the probability of crossing the barrier is not zero, they are too slow to equilibrate the particles across the two states within the simulation time for the higher barriers. In fact, not a single transition could be observed from Langevin dynamics when simulating the system with a barrier height of 32 $k_\mathrm{B}T$.

On the other hand, the simulations with birth-death events reach the equilibrium distribution of the particles within the first 1,000 steps of the simulation. It is clearly visible that the barrier height has only a negligible influence on the speed of equilibration, which is in accordance with a similar theoretic result

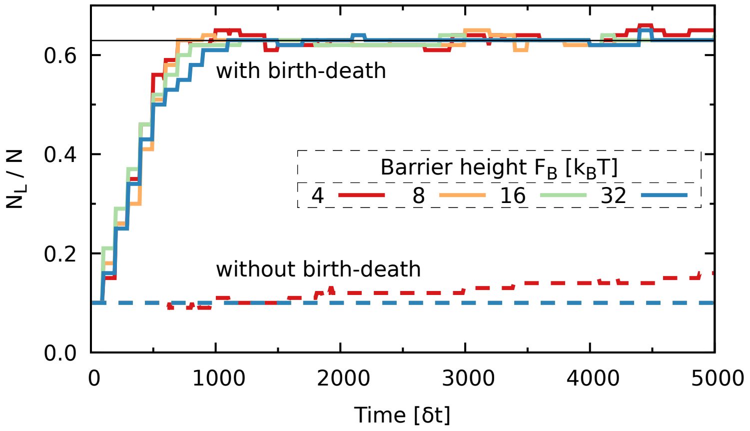


Figure 12.9: Number of particles in the left state as a function of simulation time for the different potentials. Shown are only the first 5,000 time steps. The lines without birth-death are on top of each other for all but the $F_B = 4$ case. The solid black line is the equilibrium value calculated from the potentials.

for an overdamped system with no smoothing kernel (compare Theorem 3.3 of [30]). Here, this theoretical derivation is found to be also true when using an approximation (in this case Λ^{mu}) to the birth-death term.

12.2.2 *Investigation of momentum behavior*

As discussed in Section 10.3, it was chosen for the general Langevin case to have the birth-death rates depend only on the particles' positions and not on their momenta. This approach is valid if it does not severely disturb the distribution of the momenta. An undisturbed distribution can be assumed if the momenta equilibrate on timescales shorter than the average time between accepted birth-death events.

Here, this assumption is verified by looking at the time evolution of the momenta as well as their distribution. For this, data from the simulations with the general Langevin integrator of the previous section 12.2.1 are considered. The first 200,000 time steps of the simulation with birth-death events of the potential with the lowest barrier ($F_B = 4$) are used. The momentum of every Langevin time step from all 100 particles is collected in a histogram with 1,000 bins in the range $[-5,5]$, which is subsequently compared to the expected distribution after normalization. The theoretical equilibrium distribution of the momentum is given by the Maxwell-Boltzmann distribution of eq. (2.14). In one dimension, eq. (2.14) simplifies to [223]

$$P(p) = \sqrt{\frac{\beta}{2\pi m}} \exp\left(\frac{-\beta p^2}{2m}\right). \tag{12.8}$$

A plot of the probability distributions can be seen in the left panel of Figure 12.10. Good agreement between the data from the simulation and the expected equilibrium distribution is observed.

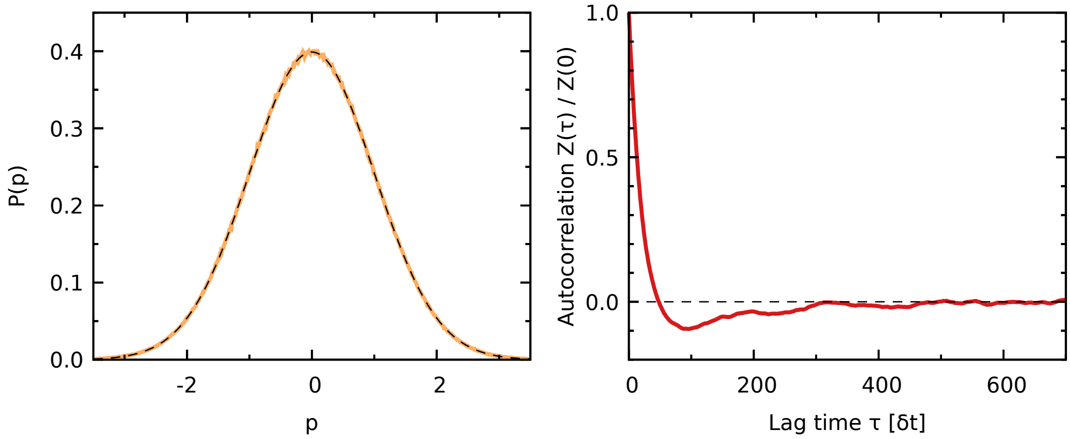


Figure 12.10: Analysis of momentum behavior when using the birth-death scheme with general Langevin dynamics.
Left panel: Momentum distribution from the simulation (orange solid line) compared to the expected Maxwell-Boltzmann distribution (black dashed line).
Right panel: Normalized autocorrelation function of the momentum of one selected particle. The lag time is given in units of the time step.

To further verify that the proposed approach does not distort the momentum distribution, the relevant timescales are investigated. The autocorrelation function of the momentum is defined as

$$Z(\tau) = \langle p(\tau) \cdot p(0) \rangle, \tag{12.9}$$

where the average runs over the time series of a particle within a simulation. The momentum autocorrelation is calculated from the first 200,000 time steps of the simulation data for selected particles. The results are very similar for all chosen particles, the plot of one exemplary autocorrelation function is presented in the right panel of Figure 12.10. It is also of note, that there are no visible differences to autocorrelation functions calculated from pure Langevin dynamics simulations of the same system.

The momentum autocorrelation, as exemplarily shown in the right panel of Figure 12.10, decays to zero within a few hundred Langevin time steps. This time scale of momentum equilibration can be compared to the average time between accepted birth-death events. During the simulation, birth-death moves are attempted after every 100th Langevin time step, of which 1.67 % are accepted. The average time between birth-death moves is therefore roughly 6,000 Langevin time steps. This is more than one order of magnitude larger than the time required for the momentum to equilibrate. It is concluded that the birth-death events have no noticeable effect on the momentum distribution. Therefore the presented approach of Algorithm 1 is justified even with general Langevin dynamics in the presented case.

12.2.3 *Simulations in higher dimensions*

All the previous results were for a system with only one spatial dimension. Because the theory from Chapter 10 holds for higher dimensions just as well, the performance of the method is tested on the two-dimensional Wolfe-Quapp potential [200, 201]

$$U(x,y) = x^4 + y^4 - 2x^2 - 4y^2 + xy + 0.3x + 0.1y, \tag{12.10}$$

that was previously described in eq. (7.2) and used for simulations in Section 7.2. Transitions between the states in y-direction are rare events, while the mobility in x-direction is high. The potential can be seen in Figure 7.3.

Simulations are run for 200,000 time steps with the overdamped Langevin integrator using 1,000 particles and otherwise the same parameters as for the one-dimensional case in Section 12.2.1. The initial distribution is again chosen to be far from equilibrium: 100 particles are placed in the metastable state at $\{-1.17, 1.48\}$ and 900 particles in the other local minimum at $\{1.12, -1.49\}$. The bandwidths of the Gaussian kernel are chosen to be symmetric in both directions, $\sigma = \sigma_x = \sigma_y$, because the low-energy regions of the potential have roughly the same size in both dimensions. It shall be noted that this is not a requirement and asymmetric kernels can be employed just as well. The kernel bandwidths are varied in the range $\sigma \in [0.05, 0.75]$ in steps of 0.05 in independent simulations. Additionally, a pure Langevin dynamics simulation without the birth-death events but otherwise the same simulation protocol is performed for comparison.

Omitting the first 10,000 time steps, the samples are again collected in a histogram. 200 equally sized bins in the range $x, y \in [-2.5, 2.5]$ are used for each direction. Because visual inspection of the FES in 2D is difficult, the FES is also projected on the y-direction. A selection of the resulting estimates of the FES as a function of the y-coordinate is presented in Fig. 12.11.

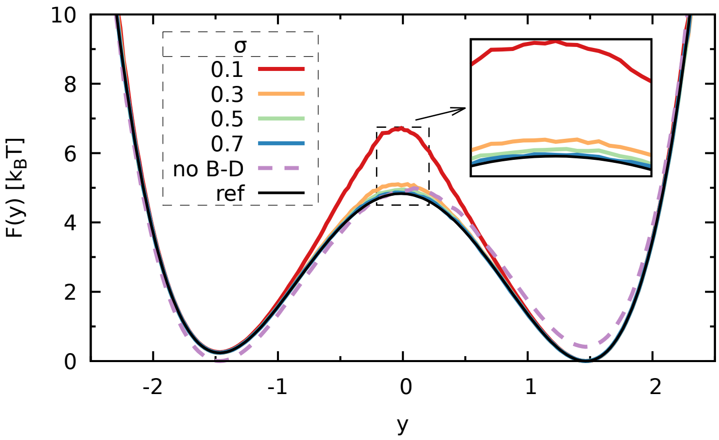


Figure 12.11: Estimates of the FES of the Wolfe-Quapp potential projected on the y-direction. Colored solid lines are from simulations with birth-death events using different kernel widths σ. The lines for $\sigma = 0.55$ and $\sigma = 0.75$ are hardly distinguishable because they are basically on top of each other. The dashed line is from a pure Langevin dynamics simulation. For clarity, it was omitted in the inset. The black line is the expected free energy calculated from the potential.

Similar behavior is observed as for the 1D potential: Above a certain kernel width σ_{crit}, the estimates from sampling are very close to the reference calculated from the potential. For too narrow kernels the barrier regions are overestimated. The simulation without the birth-death algorithm does not manage to adequately sample the system and significantly deviates not only in the barrier region but also in the relative height of the two states.

To numerically assess the correctness of the sampling, the Kullback-Leibler divergence $D_{\text{KL}}(\pi \,|\, \eta)$ from the obtained sampling distribution η to the true equilibrium distribution π is used, which is defined in eq. (3.50) and calculated in the same way as in Section 12.1.1. For this, not only the projections on the y-dimension but the full two-dimensional probability densities are used. The obtained KL divergences for different values of the kernel width σ are shown in the left panel of Figure 12.12, together with the KL divergence for the pure Langevin simulation.

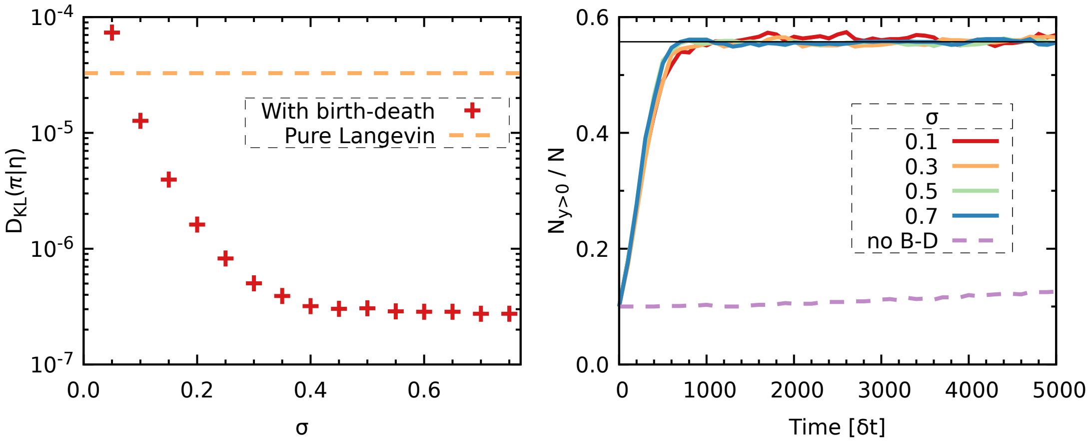


Figure 12.12: Comparison of performance of the birth-death method on the Wolfe-Quapp potential for different bandwidths σ.
Left panel: Kullback-Leibler divergences from the estimated probability distribution to the equilibrium distribution for simulations with different kernel widths σ. For comparison, the dashed horizontal line denotes the value from a simulation without the birth-death algorithm.
Right panel: Fraction of particles in the state with $y > 0$ as a function of simulation time. The black horizontal line is the expected equilibrium value. The different lines represent the same simulations as in Figure 12.11.

Additionally, a look at the equilibration process during the first time steps is taken. For this, the system is separated into two states with $y = 0$ as the dividing line. Monitoring the number of particles in these two states allows observing the slow transitions between the two main basins. A reference equilibrium value is obtained from the potential by integrating over the respective regions in the same way as it was done in Section 12.1.1 for the 1D potential. This yields $\pi(y > 0) \approx 0.56$ as the equilibrium probability for each particle to be in the region $y \in (0, \infty)$. The right panel of Figure 12.12 shows the time evolution of the number of particles in this state, $N_{y>0}$, for the first 5,000 time steps of the simulation.

Looking at the values of the Kullback-Leibler divergence in the left panel of Figure 12.12 confirms the findings from the FES projections: While suboptimal sampling is obtained for low kernel bandwidths, increasing σ above a certain value σ_{crit} yields sampling very close to the theoretical distribution denoted by low Kullback-Leibler divergence. The value of σ_{crit} can be determined to be approximately 0.4 here, although the discussion in Section 12.1.3 makes it clear that this value also depends on the number of particles employed in the simulation. To obtain a good estimate of the density from the current particle positions, with increasing dimension also more particles are required. Using the rather large number of 1,000 particles in this simulation ensured good estimates and allowed to focus on the behavior of the algorithm itself, while it is not required to obtain good sampling. For simulations of more complex systems, the number of parallel representations will typically be chosen much smaller.

Simulating the system without birth-death events clearly results in suboptimal sampling. The calculated Kullback-Leibler divergence for this simulation is about two orders of magnitude worse than the results for the birth-death augmented simulations with $\sigma > \sigma_{\text{crit}}$. This behavior can be explained when looking at the time evolution of the particle distribution in the right panel of Figure 12.12: Due to the moderate barrier, transitions by Langevin dynamics happen only on relatively long timescales. Subsequently, without the usage of additional moves, such as in the presented birth-death scheme, estimates from the sampling are not correct. With additional birth-death moves, the correct distribution is sampled and the correct equilibrium distribution of the particles is reached within less than 1,000 time steps. In conclusion, the presented algorithm can also be used to improve sampling in multiple dimensions.

13

SUMMARY AND OUTLOOK

In this part of the thesis, a new method to enhance sampling was introduced. It augments molecular sampling schemes with a cloning algorithm similar to the one proposed in Ref. 30. The presented algorithm enhances Langevin sampling with birth-death moves, which enables accurate sampling also for rare event systems. While theoretical treatment allows only to prove correct sampling for long times and infinitely many particles in the overdamped case, numerical simulations were able to show that the correct probability distribution is sampled also in simulations with finite particles.

The algorithm was tuned for better computational efficiency by increasing the time between birth-death calculations. It was found that this does not cause problems, as long as the average probabilities of birth-death events remain small. The proposed method also offers a way to gradually turn off the birth-death part of the algorithm.

Additionally, the algorithm can not only be used in the overdamped Langevin case, but with more general dynamics. When applying the birth-death algorithm only to the particle positions in simulations with general Langevin dynamics, correct sampling could also be observed.

A remarkable result of the algorithm was presented in Section 12.2.1: the speed of equilibration is independent of the barrier height of the simulated system. This is beneficial for rare event systems and might prove fruitful when the method is applied to physical or chemical systems that require more complex dynamics.

Although the theory of the method holds for potentials of any dimension, in practice the number of particles has to be increased with the number of dimensions to get smooth estimates of the particle distribution. This limits the dimensions in which the algorithm can be applied to only a few. For high dimensional systems with many degrees of freedom, such as typical chemical systems, the method will need to be modified. Similar to the CV-based enhanced sampling methods that were used in Part II, in these cases the birth-death algorithm might be applied only to a few selected slow degrees of freedom. Applying the birth-death moves only to the position distribution of the particles in the general Langevin simulations of Section 12.2 while omitting the faster degrees of freedom from the momentum can be seen as a first step in that direction.

Also, to calculate the birth-death probabilities Λ^{mu} during a simulation, eq. (10.30) requires knowledge of the equilibrium distribution π in the region of interest. Typically, this knowledge will not be available a priori but has to be obtained, for example, by estimation from the sampling. Therefore, further methodologival development is needed before the presented method can be applied to real chemical systems. Extensions of the algorithms that include on-the-fly estimates of π are currently under investigation, but this is beyond the scope of this work.

Part IV

CONCLUSION

“In this song, we’ll interrupt ourselves
And never come to an end
All conclusions remain to be drawn
Everything recalculated”

— Ampere [224]

14

CONCLUSION

This thesis introduced two newly developed aspects of enhanced sampling methods for molecular simulations. Part II analyzed the existing bias representation of the VES method and developed new basis functions. Part III presented a new sampling algorithm that employs birth-death processes to obtain equilibrium sampling even for systems with high barriers.

In Part II, criteria for good basis functions of the VES method were assembled. Numerical simulations were used to assess the performance of newly proposed localized basis sets. In particular, the basis sets constructed from Daubechies wavelets were shown to have favorable properties. For all model systems, as well as the simulation of the association process of calcium carbonate, better results than for the previously described basis sets were obtained. Especially the stability of the bias, both during the optimization process of an individual simulation as well as between different simulation runs, was found to be greatly improved. Additionally, the bias typically converged within shorter simulation times when compared to simulations that employed other basis functions. These are favorable properties when estimating observables from the simulation data. As an example, for the association process of calcium carbonate, the obtained free energy differences showed significantly smaller error bars when compared to the other biasing methods, both for the results of a single simulation and also when combining data from all three runs. Therefore, constructing the VES bias from wavelets might allow for shorter simulation times to obtain the same quality of estimates when compared to the previously proposed basis functions.

Although the presented results give a promising perspective of using wavelet-based bias potentials, the current methodology can be further extended: The multiresolution property of the wavelet basis allows to refine the resolution of the bias potential on the fly during a simulation. Also, different wavelet types might be better suited for specific systems under investigation. The presented work can therefore be seen as starting point for further studies of the performance of wavelet-based bias potentials.

Part III presented an orthogonal enhanced sampling approach to the biasing strategies of Part II: The newly proposed Algorithm 1 couples multiple copies of the system under investigation by birth-death processes. Simulating multiple replicas of the system helps to sample the configurational space more extensively and the birth-death algorithm ensures equilibrium sampling even in the presence of high barriers, independently of the barrier height. The mathematical proof of convergence to the correct sampling in the limit of infinitely many particles and infinite time was verified in the finite case via numerical simulations. Even with a moderate number of particles, equilibrium sampling was observed as long as smooth estimates of the particle density could be obtained via a kernel density estimation. The influence of several parameters of the algorithm on its behavior was studied, such as the number of particles or the width of the smoothing kernel.

To go toward simulations of chemical systems, the algorithm was also applied to more general dynamics (in the form of underdamped Langevin dynamics) while keeping the same protocol and omitting the particle momenta from the birth-death considerations. Omitting these fast degrees of freedom is the first step to a sparser application of the birth-death method: Because the number of dimensions that the method can be applied to is limited in practice, future development of the method toward a CV-based approach is suggested. Then, only a few selected slow degrees of freedom of the process of interest (the CVs) are taken into account for the birth-death events. This fits nicely with bias-based enhanced sampling methods, such as metadynamics and VES, that also employ a bias only on along certain CVs.

Using both bias-based and birth-death approaches together might help to solve difficulties in either method. The presented birth-death method leads to equilibrium sampling, which means that states with low probabilities are rarely sampled. The interacting particle approach can be seen as assigning each state a desired equilibrium number of particles. When simulating systems with metastable states at vastly different energy levels with a limited number of particles, this might be less than a single particle for the high energy states. The birth-death algorithm will then not be able to sample these states reliably. This can be cured by employing a biasing method that lowers the energy differences between states. Using, for example, the VES method with a well-tempered target distribution and biasing factor γ, reduces the sampled free energy difference between the lowest and all other states by a factor $1/\gamma$ which results in increased equilibrium probabilities at the higher states. On the other hand, the birth-death scheme helps to quickly equilibrate the system and prevents walkers from becoming correlated. Future developments might therefore lead to a combination of two orthogonal methods, where the biasing method explores and detects new states and the birth-death method exploits the new information to quickly obtain good statistics: The biasing method quickly explores the configurational space and allows sampling of previously inaccessible states by flattening barriers. The birth-death algorithm then quickly redistributes the replicas according to the equilibrium distribution, such that good estimates can be obtained already shortly after.

While the presented research of Parts II and III can be applied individualy to interesting problems, this thesis provides also a step towards the development of such a combined method. Using the robustness and quick convergence of the introduced wavelet-based bias potentials might allow for relatively short explorative segments of the algorithm, and combined with the birth-death approach this could lead to a method that provides good estimates from sampling in a efficient way within short simulation times.

ACKNOWLEDGMENTS

I would like to thank [redacted] for the opportunity of doing this research in his group at the MPIP and his supervision during the time. I express my gratitude to [redacted] for being an extraordinary supervisor and mentor whose guidance was integral to completing this thesis. Furthermore, I thank our collaborators at the mathematics department of the JGU, [redacted] and [redacted], that contributed significantly to the development of the birth-death method. I thank [redacted] for providing reference data for the calcium carbonate system. I am grateful to the TRR146 of the DFG for providing financial support during part of this research. I thank my collegues at the MPIP for their guidance and help on various kinds of problems, as well as the good conversations and times we shared. In gratitude, I would like to give a special thanks to my family and friends for their continuous support during the last years.

BIBLIOGRAPHY

[1] K. Kremer and G. S. Grest. "Dynamics of Entangled Linear Polymer Melts: A Molecular-dynamics Simulation." In: *The Journal of Chemical Physics* 92.8 (1990), pp. 5057–5086. DOI: 10.1063/1.458541.

[2] A. Iscen, N. C. Forero-Martinez, O. Valsson, and K. Kremer. "Acrylic Paints: An Atomistic View of Polymer Structure and Effects of Environmental Pollutants." In: *The Journal of Physical Chemistry B* 125.38 (2021), pp. 10854–10865. DOI: 10.1021/acs.jpcb.1c05188.

[3] J. A. McCammon, B. R. Gelin, and M. Karplus. "Dynamics of Folded Proteins." In: *Nature* 267.5612 (1977), pp. 585–590. DOI: 10.1038/267585a0.

[4] P. L. Geissler, C. Dellago, and D. Chandler. "Kinetic Pathways of Ion Pair Dissociation in Water." In: *The Journal of Physical Chemistry B* 103.18 (1999), pp. 3706–3710. DOI: 10.1021/jp984837g.

[5] D. E. Shaw et al. "Atomic-Level Characterization of the Structural Dynamics of Proteins." In: *Science* 330.6002 (2010), pp. 341–346. DOI: 10.1126/science.1187409.

[6] S. L. Rath and K. Kumar. "Investigation of the Effect of Temperature on the Structure of SARS-CoV-2 Spike Protein by Molecular Dynamics Simulations." In: *Frontiers in Molecular Biosciences* 7, 583523 (2020). DOI: 10.3389/fmolb.2020.583523.

[7] B. J. Alder and T. E. Wainwright. "Studies in Molecular Dynamics. I. General Method." In: *The Journal of Chemical Physics* 31.2 (1959), pp. 459–466. DOI: 10.1063/1.1730376.

[8] P. Langevin. "Sur La Théorie Du Mouvement Brownien." In: *Comptes-Rendus de l'Académie des Sciences* 146 (1908), pp. 530–532.

[9] P. Turq, F. Lantelme, and H. L. Friedman. "Brownian Dynamics: Its Application to Ionic Solutions." In: *The Journal of Chemical Physics* 66.7 (1977), pp. 3039–3044. DOI: 10.1063/1.434317.

[10] R. O. Dror, R. M. Dirks, J. Grossman, H. Xu, and D. E. Shaw. "Biomolecular Simulation: A Computational Microscope for Molecular Biology." In: *Annual Review of Biophysics* 41.1 (2012), pp. 429–452. DOI: 10.1146/annurev-biophys-042910-155245.

[11] D. E. Shaw et al. "Anton, a Special-Purpose Machine for Molecular Dynamics Simulation." In: *Communications of the ACM* 51.7 (2008), pp. 91–97. DOI: 10.1145/1364782.1364802.

[12] D. E. Shaw et al. "Anton 3: Twenty Microseconds of Molecular Dynamics Simulation before Lunch." *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. St. Louis, MO: Association for Computing Machinery, 2021. DOI: 10.1145/3458817.3487397.

[13] C. R. Trott et al. "Kokkos 3: Programming Model Extensions for the Exascale Era." In: *IEEE Transactions on Parallel and Distributed Systems* 33.4 (2022), pp. 805–817. DOI: 10.1109/TPDS.2021.3097283.

[14] J. C. Phillips et al. "Scalable Molecular Dynamics on CPU and GPU Architectures with NAMD." In: *The Journal of Chemical Physics* 153.4, 044130 (2020). DOI: 10.1063/5.0014475.

[15] S. Páll, A. Zhmurov, P. Bauer, M. Abraham, M. Lundborg, A. Gray, B. Hess, and E. Lindahl. "Heterogeneous Parallelization and Acceleration of Molecular Dynamics Simulations in GROMACS." In: *The Journal of Chemical Physics* 153.13, 134110 (2020). DOI: 10.1063/5.0018516.

[16] H. N. Khan, D. A. Hounshell, and E. R. H. Fuchs. "Science and Research Policy at the End of Moore's Law." In: *Nature Electronics* 1.1 (2018), pp. 14–21. DOI: 10.1038/s41928-017-0005-9.

[17] O. Valsson and M. Parrinello. "Variational Approach to Enhanced Sampling and Free Energy Calculations." In: *Physical Review Letters* 113.9, 090601 (2014). DOI: 10.1103/PhysRevLett.113.090601.

[18] O. Valsson and M. Parrinello. "Variationally Enhanced Sampling." *Handbook of Materials Modeling*. Ed. by W. Andreoni and S. Yip. 2nd ed. Cham, CH: Springer International Publishing, 2020, pp. 621–634. DOI: 10.1007/978-3-319-44677-6_50.

[19] R. Demuynck, S. M. J. Rogge, L. Vanduyfhuys, J. Wieme, M. Waroquier, and V. Van Speybroeck. "Efficient Construction of Free Energy Profiles of Breathing Metal—Organic Frameworks Using Advanced Molecular Dynamics Simulations." In: *Journal of Chemical Theory and Computation* 13.12 (2017), pp. 5861–5873. DOI: `10.1021/acs.jctc.7b01014`.

[20] R. Demuynck, J. Wieme, S. M. J. Rogge, K. D. Dedecker, L. Vanduyfhuys, M. Waroquier, and V. Van Speybroeck. "Protocol for Identifying Accurate Collective Variables in Enhanced Molecular Dynamics Simulations for the Description of Structural Transformations in Flexible Metal–Organic Frameworks." In: *Journal of Chemical Theory and Computation* 14.11 (2018), pp. 5511–5526. DOI: `10.1021/acs.jctc.8b00725`.

[21] L. Bonati, Y.-Y. Zhang, and M. Parrinello. "Neural Networks-Based Variationally Enhanced Sampling." In: *Proceedings of the National Academy of Sciences* 116.36 (2019), pp. 17641–17647. DOI: `10.1073/pnas.1907975116`.

[22] P. M. Piaggi, O. Valsson, and M. Parrinello. "A Variational Approach to Nucleation Simulation." In: *Faraday Discussions* 195 (2016), pp. 557–568. DOI: `10.1039/C6FD00127K`.

[23] J. McCarty, O. Valsson, and M. Parrinello. "Bespoke Bias for Obtaining Free Energy Differences within Variationally Enhanced Sampling." In: *Journal of Chemical Theory and Computation* 12.5 (2016), pp. 2162–2169. DOI: `10.1021/acs.jctc.6b00125`.

[24] M. Invernizzi, O. Valsson, and M. Parrinello. "Coarse Graining from Variationally Enhanced Sampling Applied to the Ginzburg–Landau Model." In: *Proceedings of the National Academy of Sciences* 114.13 (2017), pp. 3370–3374. DOI: `10.1073/pnas.1618455114`.

[25] M. Invernizzi and M. Parrinello. "Making the Best of a Bad Situation: A Multiscale Approach to Free Energy Calculation." In: *Journal of Chemical Theory and Computation* 15.4 (2019), pp. 2187–2194. DOI: `10.1021/acs.jctc.9b00032`.

[26] I. Daubechies. "Orthonormal Bases of Compactly Supported Wavelets." In: *Communications on Pure and Applied Mathematics* 41.7 (1988), pp. 909–996. DOI: `10.1002/cpa.3160410705`.

[27] S. Mohr, L. E. Ratcliff, P. Boulanger, L. Genovese, D. Caliste, T. Deutsch, and S. Goedecker. "Daubechies Wavelets for Linear Scaling Density Functional Theory." In: *The Journal of Chemical Physics* 140.20, 204110 (2014). DOI: `10.1063/1.4871876`.

[28] L. E. Ratcliff et al. "Flexibilities of Wavelets as a Computational Basis Set for Large-Scale Electronic Structure Calculations." In: *The Journal of Chemical Physics* 152.19, 194110 (2020). DOI: `10.1063/5.0004792`.

[29] M. Maiolo, A. Vancheri, R. Krause, and A. Danani. "Wavelets as Basis Functions to Represent the Coarse-Graining Potential in Multiscale Coarse Graining Approach." In: *Journal of Computational Physics* 300 (2015), pp. 592–604. DOI: `10.1016/j.jcp.2015.07.039`.

[30] Y. Lu, J. Lu, and J. Nolen. *Accelerating Langevin Sampling with Birth-death*. Preprint. 2019. arXiv: `1905.09863 [stat.ML]`.

[31] A. S. Sherman and C. S. Peskin. "A Monte Carlo Method for Scalar Reaction Diffusion Equations." In: *SIAM Journal on Scientific and Statistical Computing* 7.4 (1986), pp. 1360–1372. DOI: `10.1137/0907090`.

[32] G. Rotskoff, S. Jelassi, J. Bruna, and E. Vanden-Eijnden. "Global Convergence of Neuron Birth-Death Dynamics." *Proceedings of the 36th International Conference on Machine Learning*. 2019, pp. 9689–9698.

[33] D. Aldous and U. Vazirani. ""Go with the Winners" Algorithms." *Proceedings 35th Annual Symposium on Foundations of Computer Science*. Santa Fe, NM: IEEE Comput. Soc. Press, 1994, pp. 492–501. DOI: `10.1109/SFCS.1994.365742`.

[34] J. B. Anderson. "A Random-walk Simulation of the Schrödinger Equation: $H^{+}{}_{3}$." In: *The Journal of Chemical Physics* 63.4 (1975), pp. 1499–1503. DOI: `10.1063/1.431514`.

[35] P. Grassberger. "Go with the Winners: A General Monte Carlo Strategy." In: *Computer Physics Communications* 147.1-2 (2002), pp. 64–70. DOI: `10.1016/S0010-4655(02)00205-9`.

[36] C. Giardina, J. Kurchan, V. Lecomte, and J. Tailleur. "Simulating Rare Events in Dynamical Processes." In: *Journal of Statistical Physics* 145.4 (2011), pp. 787–811. DOI: `10.1007/s10955-011-0350-4`.

[37] P. Del Moral and J. Garnier. “Genealogical Particle Analysis of Rare Events.” In: *The Annals of Applied Probability* 15.4 (2005), pp. 2496–2534. DOI: `10.1214/105051605000000566`.

[38] G. A. Tribello, M. Bonomi, D. Branduardi, C. Camilloni, and G. Bussi. “PLUMED 2: New Feathers for an Old Bird.” In: *Computer Physics Communications* 185.2 (2014), pp. 604–613. DOI: `10.1016/j.cpc.2013.09.018`.

[39] The PLUMED consortium. “Promoting Transparency and Reproducibility in Enhanced Molecular Simulations.” In: *Nature Methods* 16.8 (2019), pp. 670–673. DOI: `10.1038/s41592-019-0506-8`.

[40] Rage Against the Machine. *Take the Power Back*. Lyrics written by Z. M. de la Rocha. In: *Rage Against the Machine*. CD, Epic Records. 1992.

[41] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. “Equation of State Calculations by Fast Computing Machines.” In: *The Journal of Chemical Physics* 21.6 (1953), pp. 1087–1092. DOI: `10.1063/1.1699114`.

[42] D. P. Landau and K. Binder. *A Guide to Monte Carlo Simulations in Statistical Physics*. 4th ed. Cambridge: Cambridge University Press, 2014. DOI: `10.1017/CBO9781139696463`.

[43] D. Frenkel and B. Smit. *Understanding Molecular Simulation: From Algorithms to Applications*. 2nd ed. Computational Science Series. San Diego: Academic Press, 2002. ISBN: 978-0-12-267351-1.

[44] T. Schlick. *Molecular Modeling and Simulation: An Interdisciplinary Guide*. 2nd ed. Interdisciplinary Applied Mathematics 21. New York: Springer, 2010. DOI: `10.1007/978-1-4419-6351-2`.

[45] F. Jensen. *Introduction to Computational Chemistry*. 3rd ed. Chichester, UK; Hoboken, NJ: Wiley, 2017. ISBN: 978-1-118-82595-2.

[46] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. 2nd ed. Oxford, UK: Oxford University Press, 2017. ISBN: 978-0-19-880319-5.

[47] G. A. Pavliotis. *Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations*. Texts in Applied Mathematics 60. New York, NY: Springer New York, 2014. DOI: `10.1007/978-1-4939-1323-7`.

[48] G. Bussi and M. Parrinello. “Accurate Sampling Using Langevin Dynamics.” In: *Physical Review E* 75.5, 056707 (2007). DOI: `10.1103/PhysRevE.75.056707`.

[49] L. Verlet. “Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules.” In: *Physical Review* 159.1 (1967), pp. 98–103. DOI: 10.1103/PhysRev.159.98.

[50] W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson. “A Computer Simulation Method for the Calculation of Equilibrium Constants for the Formation of Physical Clusters of Molecules: Application to Small Water Clusters.” In: *The Journal of Chemical Physics* 76.1 (1982), pp. 637–649. DOI: 10.1063/1.442716.

[51] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman. “A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules.” In: *Journal of the American Chemical Society* 117.19 (1995), pp. 5179–5197. DOI: 10.1021/ja00124a002.

[52] J. Wang, R. M. Wolf, J. W. Caldwell, P. A. Kollman, and D. A. Case. “Development and Testing of a General Amber Force Field.” In: *Journal of Computational Chemistry* 25.9 (2004), pp. 1157–1174. DOI: 10.1002/jcc.20035.

[53] K. Vanommeslaeghe et al. “CHARMM General Force Field: A Force Field for Drug-like Molecules Compatible with the CHARMM All-Atom Additive Biological Force Fields.” In: *Journal of Computational Chemistry* 31 (2009), pp. 671–690. DOI: 10.1002/jcc.21367.

[54] R. B. Best, X. Zhu, J. Shim, P. E. M. Lopes, J. Mittal, M. Feig, and A. D. MacKerell. “Optimization of the Additive CHARMM All-Atom Protein Force Field Targeting Improved Sampling of the Backbone ϕ, ψ and Side-Chain χ_1 and χ_2 Dihedral Angles.” In: *Journal of Chemical Theory and Computation* 8.9 (2012), pp. 3257–3273. DOI: 10.1021/ct300400x.

[55] J. W. Ponder and D. A. Case. “Force Fields for Protein Simulations.” *Advances in Protein Chemistry*. Vol. 66. Elsevier, 2003, pp. 27–85. DOI: 10.1016/S0065-3233(03)66002-X.

[56] J. E. Jones. “On the Determination of Molecular Fields. — I. From the Variation of the Viscosity of a Gas with Temperature.” In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 106.738 (1924), pp. 441–462. DOI: 10.1098/rspa.1924.0081.

[57] R. A. Buckingham. “The Classical Equation of State of Gaseous Helium, Neon and Argon.” In: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 168.933 (1938), pp. 264–283. DOI: 10.1098/rspa.1938.0173.

[58] M. P. Allen. “Introduction to Molecular Dynamics Simulation.” *Computational Soft Matter: From Synthetic Polymers to Proteins*. Vol. 23. NIC Series. Jülich: N. Attig, K. Binder, H. Grubmüller, K. Kremer, 2004, pp. 1–28. ISBN: 3-00-012641-4.

[59] P. P. Ewald. “Die Berechnung optischer und elektrostatischer Gitterpotentiale.” In: *Annalen der Physik* 369.3 (1921), pp. 253–287. DOI: 10.1002/andp.19213690304.

[60] J. Eastwood, R. Hockney, and D. Lawrence. “P3M3DP—The Three-Dimensional Periodic Particle-Particle/ Particle-Mesh Program.” In: *Computer Physics Communications* 19.2 (1980), pp. 215–261. DOI: 10.1016/0010-4655(80)90052-1.

[61] R. W. Hockney and J. W. Eastwood. *Computer Simulation Using Particles*. Bristol, UK; Philadelphia, PA: CRC Press, 1988. DOI: 10.1201/9780367806934.

[62] H. C. Andersen. “Molecular Dynamics Simulations at Constant Pressure and/or Temperature.” In: *The Journal of Chemical Physics* 72.4 (1980), pp. 2384–2393. DOI: 10.1063/1.439486.

[63] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak. “Molecular Dynamics with Coupling to an External Bath.” In: *The Journal of Chemical Physics* 81.8 (1984), pp. 3684–3690. DOI: 10.1063/1.448118.

[64] S. Nosé. “A Unified Formulation of the Constant Temperature Molecular Dynamics Methods.” In: *The Journal of Chemical Physics* 81.1 (1984), pp. 511–519. DOI: 10.1063/1.447334.

[65] W. G. Hoover. “Canonical Dynamics: Equilibrium Phase-Space Distributions.” In: *Physical Review A* 31.3 (1985), pp. 1695–1697. DOI: 10.1103/PhysRevA.31.1695.

[66] G. Bussi, D. Donadio, and M. Parrinello. “Canonical Sampling through Velocity Rescaling.” In: *The Journal of Chemical Physics* 126.1, 014101 (2007). DOI: 10.1063/1.2408420.

[67] G. J. Martyna, M. L. Klein, and M. Tuckerman. “Nosé–Hoover Chains: The Canonical Ensemble via Continuous Dynamics.” In: *The Journal of Chemical Physics* 97.4 (1992), pp. 2635–2643. DOI: 10.1063/1.463940.

[68] G. J. Martyna, D. J. Tobias, and M. L. Klein. "Constant Pressure Molecular Dynamics Algorithms." In: *The Journal of Chemical Physics* 101.5 (1994), pp. 4177–4189. DOI: 10.1063/1.467468.

[69] G. Maruyama. "Continuous Markov Processes and Stochastic Equations." In: *Rendiconti del Circolo Matematico di Palermo* 4.1 (1955), pp. 48–90. DOI: 10.1007/BF02846028.

[70] P. E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. 3rd ed. Applications of Mathematics 23. Berlin, DE; New York, NY: Springer, 1999. ISBN: 978-3-540-54062-5.

[71] P. V. Coveney and S. Wan. "On the Calculation of Equilibrium Thermodynamic Properties from Molecular Dynamics." In: *Physical Chemistry Chemical Physics* 18.44 (2016), pp. 30236–30240. DOI: 10.1039/C6CP02349E.

[72] M. E. Tuckerman. *Statistical Mechanics: Theory and Molecular Simulation*. Oxford, UK; New York, NY: Oxford University Press, 2010. ISBN: 978-0-19-852526-4.

[73] G. Bussi, A. Laio, and P. Tiwary. "Metadynamics: A Unified Framework for Accelerating Rare Events and Sampling Thermodynamics and Kinetics." *Handbook of Materials Modeling*. Ed. by W. Andreoni and S. Yip. Cham: Springer International Publishing, 2020, pp. 565–595. DOI: 10.1007/978-3-319-44677-6_49.

[74] P. Gkeka et al. "Machine Learning Force Fields and Coarse-Grained Variables in Molecular Dynamics: Application to Materials and Biological Systems." In: *Journal of Chemical Theory and Computation* 16.8 (2020), pp. 4757–4775. DOI: 10.1021/acs.jctc.0c00355.

[75] G. Fiorin, M. L. Klein, and J. Hénin. "Using Collective Variables to Drive Molecular Dynamics Simulations." In: *Molecular Physics* 111.22-23 (2013), pp. 3345–3362. DOI: 10.1080/00268976.2013.813594.

[76] F. Pietrucci. "Strategies for the Exploration of Free Energy Landscapes: Unity in Diversity and Challenges Ahead." In: *Reviews in Physics* 2 (2017), pp. 32–45. DOI: 10.1016/j.revip.2017.05.001.

[77] O. Valsson, P. Tiwary, and M. Parrinello. "Enhancing Important Fluctuations: Rare Events and Metadynamics from a Conceptual Viewpoint." In: *Annual Review of Physical Chemistry* 67.1 (2016), pp. 159–184. DOI: 10.1146/annurev-physchem-040215-112229.

[78] K. Pearson. “LIII. On Lines and Planes of Closest Fit to Systems of Points in Space.” In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572. DOI: `10.1080/14786440109462720`.

[79] G. Pérez-Hernández, F. Paul, T. Giorgino, G. De Fabritiis, and F. Noé. “Identification of Slow Molecular Order Parameters for Markov Model Construction.” In: *The Journal of Chemical Physics* 139.1, 015102 (2013). DOI: `10.1063/1.4811489`.

[80] D. Branduardi, F. L. Gervasio, and M. Parrinello. “From A to B in Free Energy Space.” In: *The Journal of Chemical Physics* 126.5, 054103 (2007). DOI: `10.1063/1.2432340`.

[81] M. Ceriotti, G. A. Tribello, and M. Parrinello. “Simplifying the Representation of Complex Free-Energy Landscapes Using Sketch-Map.” In: *Proceedings of the National Academy of Sciences* 108.32 (2011), pp. 13023–13028. DOI: `10.1073/pnas.1108486108`.

[82] P. Tiwary and B. J. Berne. “Spectral Gap Optimization of Order Parameters for Sampling Complex Molecular Systems.” In: *Proceedings of the National Academy of Sciences* 113.11 (2016), pp. 2839–2844. DOI: `10.1073/pnas.1600917113`.

[83] J. M. L. Ribeiro, P. Bravo, Y. Wang, and P. Tiwary. “Reweighted Autoencoded Variational Bayes for Enhanced Sampling (RAVE).” In: *The Journal of Chemical Physics* 149.7, 072301 (2018). DOI: `10.1063/1.5025487`.

[84] Y. Bozkurt Varolgüneş, T. Bereau, and J. F. Rudzinski. “Interpretable Embeddings from Molecular Simulations Using Gaussian Mixture Variational Autoencoders.” In: *Machine Learning: Science and Technology* 1.1, 015012 (2020). DOI: `10.1088/2632-2153/ab80b7`.

[85] T. Lemke and C. Peter. “EncoderMap: Dimensionality Reduction and Generation of Molecule Conformations.” In: *Journal of Chemical Theory and Computation* 15.2 (2019), pp. 1209–1215. DOI: `10.1021/acs.jctc.8b00975`.

[86] W. Chen and A. L. Ferguson. “Molecular Enhanced Sampling with Autoencoders: On-the-fly Collective Variable Discovery and Accelerated Free Energy Landscape Exploration.” In: *Journal of Computational Chemistry* 39.25 (2018), pp. 2079–2102. DOI: `10.1002/jcc.25520`.

[87] H. Sidky, W. Chen, and A. L. Ferguson. “Machine Learning for Collective Variable Discovery and Enhanced Sampling in Biomolecular Simulation.” In: *Molecular Physics* 118.5, e1737742 (2020). DOI: `10.1080/00268976.2020.1737742`.

[88] C. Hartmann, J. C. Latorre, and G. Ciccotti. “On Two Possible Definitions of the Free Energy for Collective Variables.” In: *The European Physical Journal Special Topics* 200.1 (2011), pp. 73–89. DOI: 10.1140/epjst/e2011-01519-7.

[89] G. M. Torrie and J. P. Valleau. “Nonphysical Sampling Distributions in Monte Carlo Free-Energy Estimation: Umbrella Sampling.” In: *Journal of Computational Physics* 23.2 (1977), pp. 187–199. DOI: 10.1016/0021-9991(77)90121-8.

[90] J. Kästner. “Umbrella Sampling.” In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 1.6 (2011), pp. 932–942. DOI: 10.1002/wcms.66.

[91] R. W. Zwanzig. “High-Temperature Equation of State by a Perturbation Method. I. Nonpolar Gases.” In: *The Journal of Chemical Physics* 22.8 (1954), pp. 1420–1426. DOI: 10.1063/1.1740409.

[92] J. G. Kirkwood. “Statistical Mechanics of Fluid Mixtures.” In: *The Journal of Chemical Physics* 3.5 (1935), pp. 300–313. DOI: 10.1063/1.1749657.

[93] T. Huber, A. E. Torda, and W. F. van Gunsteren. “Local Elevation: A Method for Improving the Searching Properties of Molecular Dynamics Simulation.” In: *Journal of Computer-Aided Molecular Design* 8.6 (1994), pp. 695–708. DOI: 10.1007/BF00124016.

[94] E. Darve and A. Pohorille. “Calculating Free Energies Using Average Force.” In: *The Journal of Chemical Physics* 115.20 (2001), pp. 9169–9183. DOI: 10.1063/1.1410978.

[95] J. Comer, J. C. Gumbart, J. Hénin, T. Lelièvre, A. Pohorille, and C. Chipot. “The Adaptive Biasing Force Method: Everything You Always Wanted to Know but Were Afraid to Ask.” In: *The Journal of Physical Chemistry B* 119.3 (2015), pp. 1129–1151. DOI: 10.1021/jp506633n.

[96] U. H. E. Hansmann and L. T. Wille. “Global Optimization by Energy Landscape Paving.” In: *Physical Review Letters* 88.6, 068105 (2002). DOI: 10.1103/PhysRevLett.88.068105.

[97] H. Grubmüller. “Predicting Slow Structural Transitions in Macromolecular Systems: Conformational Flooding.” In: *Physical Review E* 52.3 (1995), pp. 2893–2906. DOI: 10.1103/PhysRevE.52.2893.

[98] E. M. Müller, A. de Meijere, and H. Grubmüller. “Predicting Unimolecular Chemical Reactions: Chemical Flooding.” In: *The Journal of Chemical Physics* 116.3 (2002), pp. 897–905. DOI: 10.1063/1.1427722.

[99] P. Maragakis, A. van der Vaart, and M. Karplus. “Gaussian-Mixture Umbrella Sampling.” In: *The Journal of Physical Chemistry B* 113.14 (2009), pp. 4664–4673. DOI: 10.1021/jp808381s.

[100] J. K. Whitmer, C.-c. Chiu, A. A. Joshi, and J. J. de Pablo. “Basis Function Sampling: A New Paradigm for Material Property Computation.” In: *Physical Review Letters* 113.19, 190602 (2014). DOI: 10.1103/PhysRevLett.113.190602.

[101] J. K. Whitmer, A. M. Fluitt, L. Antony, J. Qin, M. McGovern, and J. J. de Pablo. “Sculpting Bespoke Mountains: Determining Free Energies with Basis Expansions.” In: *The Journal of Chemical Physics* 143.4, 044101 (2015). DOI: 10.1063/1.4927147.

[102] M. Invernizzi and M. Parrinello. “Rethinking Metadynamics: From Bias Potentials to Probability Distributions.” In: *The Journal of Physical Chemistry Letters* 11.7 (2020), pp. 2731–2736. DOI: 10.1021/acs.jpclett.0c00497.

[103] M. Invernizzi, P. M. Piaggi, and M. Parrinello. “Unified Approach to Enhanced Sampling.” In: *Physical Review X* 10.4, 041034 (2020). DOI: 10.1103/PhysRevX.10.041034.

[104] F. Giberti, G. A. Tribello, and M. Ceriotti. “Global Free-Energy Landscapes as a Smoothly Joined Collection of Local Maps.” In: *Journal of Chemical Theory and Computation* 17.6 (2021), pp. 3292–3308. DOI: 10.1021/acs.jctc.0c01177.

[105] A. Laio and M. Parrinello. “Escaping Free-Energy Minima.” In: *Proceedings of the National Academy of Sciences* 99.20 (2002), pp. 12562–12566. DOI: 10.1073/pnas.202427399.

[106] A. Barducci, G. Bussi, and M. Parrinello. “Well-Tempered Metadynamics: A Smoothly Converging and Tunable Free-Energy Method.” In: *Physical Review Letters* 100.2, 020603 (2008). DOI: 10.1103/PhysRevLett.100.020603.

[107] B. A. Berg and T. Neuhaus. “Multicanonical Ensemble: A New Approach to Simulate First-Order Phase Transitions.” In: *Physical Review Letters* 68.1 (1992), pp. 9–12. DOI: 10.1103/PhysRevLett.68.9.

[108] N. Nakajima, H. Nakamura, and A. Kidera. "Multicanonical Ensemble Generated by Molecular Dynamics Simulation for Enhanced Conformational Sampling of Peptides." In: *The Journal of Physical Chemistry B* 101.5 (1997), pp. 817–824. DOI: `10.1021/jp962142e`.

[109] F. Wang and D. P. Landau. "Efficient, Multiple-Range Random Walk Algorithm to Calculate the Density of States." In: *Physical Review Letters* 86.10 (2001), pp. 2050–2053. DOI: `10.1103/PhysRevLett.86.2050`.

[110] A. F. Voter. "Hyperdynamics: Accelerated Molecular Dynamics of Infrequent Events." In: *Physical Review Letters* 78.20 (1997), pp. 3908–3911. DOI: `10.1103/PhysRevLett.78.3908`.

[111] A. F. Voter. "A Method for Accelerating the Molecular Dynamics Simulation of Infrequent Events." In: *The Journal of Chemical Physics* 106.11 (1997), pp. 4665–4677. DOI: `10.1063/1.473503`.

[112] D. Hamelberg, J. Mongan, and J. A. McCammon. "Accelerated Molecular Dynamics: A Promising and Efficient Simulation Method for Biomolecules." In: *The Journal of Chemical Physics* 120.24 (2004), pp. 11919–11929. DOI: `10.1063/1.1755656`.

[113] M. R. Sørensen and A. F. Voter. "Temperature-Accelerated Dynamics for Simulation of Infrequent Events." In: *The Journal of Chemical Physics* 112.21 (2000), pp. 9599–9606. DOI: `10.1063/1.481576`.

[114] E. Marinari and G. Parisi. "Simulated Tempering: A New Monte Carlo Scheme." In: *Europhysics Letters (EPL)* 19.6 (1992), pp. 451–458. DOI: `10.1209/0295-5075/19/6/002`.

[115] T. Kloek and H. K. van Dijk. "Bayesian Estimates of Equation System Parameters: An Application of Integration by Monte Carlo." In: *Econometrica* 46.1 (1978), pp. 1–19. DOI: `10.2307/1913641`.

[116] H. C. Lie and J. Quer. "Some Connections between Importance Sampling and Enhanced Sampling Methods in Molecular Dynamics." In: *The Journal of Chemical Physics* 147.19, 194107 (2017). DOI: `10.1063/1.4989495`.

[117] S. Piana and A. Laio. "A Bias-Exchange Approach to Protein Folding." In: *The Journal of Physical Chemistry B* 111.17 (2007), pp. 4553–4559. DOI: `10.1021/jp0678731`.

[118] F. Baftizadeh, P. Cossio, F. Pietrucci, and A. Laio. “Protein Folding and Ligand-Enzyme Binding from Bias-Exchange Metadynamics Simulations.” In: *Current Physical Chemistry* 2 (2012), pp. 79–91. DOI: 10.2174/1877947611202010079.

[119] G. Bussi, F. L. Gervasio, A. Laio, and M. Parrinello. “Free-Energy Landscape for β Hairpin Folding from Combined Parallel Tempering and Metadynamics.” In: *Journal of the American Chemical Society* 128.41 (2006), pp. 13435–13441. DOI: 10.1021/ja062463w.

[120] M. Kellermeier, P. Raiteri, J. K. Berg, A. Kempter, J. D. Gale, and D. Gebauer. “Entropy Drives Calcium Carbonate Ion Association.” In: *ChemPhysChem* 17.21 (2016), pp. 3535–3541. DOI: 10.1002/cphc.201600653.

[121] F. Trudu, D. Donadio, and M. Parrinello. “Freezing of a Lennard-Jones Fluid: From Nucleation to Spinodal Regime.” In: *Physical Review Letters* 97.10, 105701 (2006). DOI: 10.1103/PhysRevLett.97.105701.

[122] S. Angioletti-Uberti, M. Ceriotti, P. D. Lee, and M. W. Finnis. “Solid-Liquid Interface Free Energy through Metadynamics Simulations.” In: *Physical Review B* 81.12, 125416 (2010). DOI: 10.1103/PhysRevB.81.125416.

[123] S. Prestipino and P. V. Giaquinta. “Liquid-Solid Coexistence via the Metadynamics Approach.” In: *The Journal of Chemical Physics* 128.11, 114707 (2008). DOI: 10.1063/1.2841081.

[124] K. Meier, T. Laino, and A. Curioni. “Solid-State Electrolytes: Revealing the Mechanisms of Li-ion Conduction in Tetragonal and Cubic LLZO by First-Principles Calculations.” In: *The Journal of Physical Chemistry C* 118.13 (2014), pp. 6668–6679. DOI: 10.1021/jp5002463.

[125] C. Domene, P. Barbini, and S. Furini. “Bias-Exchange Metadynamics Simulations: An Efficient Strategy for the Analysis of Conduction and Selectivity in Ion Channels.” In: *Journal of Chemical Theory and Computation* 11.4 (2015), pp. 1896–1906. DOI: 10.1021/ct501053x.

[126] L. Ceccarini, M. Masetti, A. Cavalli, and M. Recanatini. “Ion Conduction through the hERG Potassium Channel.” In: *PLOS ONE* 7.11, e49017 (2012). DOI: 10.1371/journal.pone.0049017.

[127] M. Alfonso-Prieto, X. Biarnés, P. Vidossich, and C. Rovira. “The Molecular Mechanism of the Catalase Reaction.” In: *Journal of the American Chemical Society* 131.33 (2009), pp. 11751–11761. DOI: 10.1021/ja9018572.

[128] E. Molina-Montes, D. Donadio, A. Hernández-Laguna, C. I. Sainz-Díaz, and M. Parrinello. "DFT Research on the Dehydroxylation Reaction of Pyrophyllite 1. First-principle Molecular Dynamics Simulations." In: *The Journal of Physical Chemistry B* 112.23 (2008), pp. 7051–7060. DOI: `10.1021/jp711278s`.

[129] F. Pietrucci, M. Bernasconi, A. Laio, and M. Parrinello. "Vacancy-Vacancy Interaction and Oxygen Diffusion in Stabilized Cubic ZrO 2 from First Principles." In: *Physical Review B* 78.9, 094301 (2008). DOI: `10.1103/PhysRevB.78.094301`.

[130] U. Aschauer, P. Bowen, and S. Parker. "Oxygen Vacancy Diffusion in Alumina: New Atomistic Simulation Methods Applied to an Old Problem." In: *Acta Materialia* 57.16 (2009), pp. 4765–4772. DOI: `10.1016/j.actamat.2009.06.061`.

[131] Y.-J. Wang, J.-P. Du, S. Shinzato, L.-H. Dai, and S. Ogata. "A Free Energy Landscape Perspective on the Nature of Collective Diffusion in Amorphous Solids." In: *Acta Materialia* 157 (2018), pp. 165–173. DOI: `10.1016/j.actamat.2018.07.029`.

[132] P. Raiteri, A. Laio, F. L. Gervasio, C. Micheletti, and M. Parrinello. "Efficient Reconstruction of Complex Free Energy Landscapes by Multiple Walkers Metadynamics." In: *The Journal of Physical Chemistry B* 110.8 (2006), pp. 3533–3539. DOI: `10.1021/jp054359r`.

[133] D. Branduardi, G. Bussi, and M. Parrinello. "Metadynamics with Adaptive Gaussians." In: *Journal of Chemical Theory and Computation* 8.7 (2012), pp. 2247–2254. DOI: `10.1021/ct3002464`.

[134] P. Tiwary and M. Parrinello. "From Metadynamics to Dynamics." In: *Physical Review Letters* 111.23, 230602 (2013). DOI: `10.1103/PhysRevLett.111.230602`.

[135] K. M. Bal and E. C. Neyts. "Merging Metadynamics into Hyperdynamics: Accelerated Molecular Simulations Reaching Time Scales from Microseconds to Seconds." In: *Journal of Chemical Theory and Computation* 11.10 (2015), pp. 4545–4554. DOI: `10.1021/acs.jctc.5b00597`.

[136] J. Pfaendtner and M. Bonomi. "Efficient Sampling of High-Dimensional Free-Energy Landscapes with Parallel Bias Metadynamics." In: *Journal of Chemical Theory and Computation* 11.11 (2015), pp. 5062–5067. DOI: `10.1021/acs.jctc.5b00846`.

[137] G. Piccini and M. Parrinello. “Accurate Quantum Chemical Free Energies at Affordable Cost.” In: *The Journal of Physical Chemistry Letters* 10.13 (2019), pp. 3727–3731. DOI: 10.1021/acs.jpclett.9b01301.

[138] A. Laio, A. Rodriguez-Fortea, F. L. Gervasio, M. Ceccarelli, and M. Parrinello. “Assessing the Accuracy of Metadynamics.” In: *The Journal of Physical Chemistry B* 109.14 (2005), pp. 6714–6721. DOI: 10.1021/jp045424k.

[139] J. F. Dama, M. Parrinello, and G. A. Voth. “Well-Tempered Metadynamics Converges Asymptotically.” In: *Physical Review Letters* 112.24, 240602 (2014). DOI: 10.1103/PhysRevLett.112.240602.

[140] A. Barducci, M. Bonomi, and M. Parrinello. “Metadynamics.” In: *WIREs Computational Molecular Science* 1.5 (2011), pp. 826–843. DOI: 10.1002/wcms.31.

[141] G. Bussi and A. Laio. “Using Metadynamics to Explore Complex Free-Energy Landscapes.” In: *Nature Reviews Physics* 2.4 (2020), pp. 200–212. DOI: 10.1038/s42254-020-0153-0.

[142] P. Tiwary, J. F. Dama, and M. Parrinello. “A Perturbative Solution to Metadynamics Ordinary Differential Equation.” In: *The Journal of Chemical Physics* 143.23, 234112 (2015). DOI: 10.1063/1.4937945.

[143] P. Tiwary and M. Parrinello. “A Time-Independent Free Energy Estimator for Metadynamics.” In: *The Journal of Physical Chemistry B* 119.3 (2015), pp. 736–742. DOI: 10.1021/jp504920s.

[144] D. W. Scott. “Averaged Shifted Histograms: Effective Nonparametric Density Estimators in Several Dimensions.” In: *The Annals of Statistics* 13.3 (1985), pp. 1024–1040. DOI: 10.1214/aos/1176349654.

[145] D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. 1st ed. Wiley Series in Probability and Statistics. Wiley, 1992. DOI: 10.1002/9780470316849.

[146] D. W. Scott. “Averaged Shifted Histogram.” In: *Wiley Interdisciplinary Reviews: Computational Statistics* 2.2 (2010), pp. 160–164. DOI: 10.1002/wics.54.

[147] H. Niu, L. Bonati, P. M. Piaggi, and M. Parrinello. “Ab Initio Phase Diagram and Nucleation of Gallium.” In: *Nature Communications* 11.1, 2654 (2020). DOI: 10.1038/s41467-020-16372-9.

[148] J. Debnath, M. Invernizzi, and M. Parrinello. “Enhanced Sampling of Transition States.” In: *Journal of Chemical Theory and Computation* 15.4 (2019), pp. 2454–2459. DOI: 10.1021/acs.jctc.8b01283.

[149] F. Palazzesi, O. Valsson, and M. Parrinello. “Conformational Entropy as Collective Variable for Proteins.” In: *The Journal of Physical Chemistry Letters* 8.19 (2017), pp. 4752–4756. DOI: 10.1021/acs.jpclett.7b01770.

[150] J. McCarty, O. Valsson, P. Tiwary, and M. Parrinello. “Variationally Optimized Free-Energy Flooding for Rate Calculation.” In: *Physical Review Letters* 115.7, 070601 (2015). DOI: 10.1103/PhysRevLett.115.070601.

[151] Y. Wu and R. Car. “Variational Approach to Monte Carlo Renormalization Group.” In: *Physical Review Letters* 119.22, 220602 (2017). DOI: 10.1103/PhysRevLett.119.220602.

[152] Y. Wu and R. Car. “Determination of the Critical Manifold Tangent Space and Curvature with Monte Carlo Renormalization Group.” In: *Physical Review E* 100.2, 022138 (2019). DOI: 10.1103/PhysRevE.100.022138.

[153] Y. Wu and R. Car. “Monte Carlo Renormalization Group for Classical Lattice Models with Quenched Disorder.” In: *Physical Review Letters* 125.19, 190601 (2020). DOI: 10.1103/PhysRevLett.125.190601.

[154] P. M. Piaggi, A. Z. Panagiotopoulos, P. G. Debenedetti, and R. Car. “Phase Equilibrium of Water with Hexagonal and Cubic Ice Using the SCAN Functional.” In: *Journal of Chemical Theory and Computation* 17.5 (2021), pp. 3065–3077. DOI: 10.1021/acs.jctc.1c00041.

[155] P. Shaffer, O. Valsson, and M. Parrinello. “Enhanced, Targeted Sampling of High-Dimensional Free-Energy Landscapes Using Variationally Enhanced Sampling, with an Application to Chignolin.” In: *Proceedings of the National Academy of Sciences* 113.5 (2016), pp. 1150–1155. DOI: 10.1073/pnas.1519712113.

[156] J. Rydzewski, K. Walczewska-Szewc, S. Czach, W. Nowak, and K. Kuczera. *Enhancing the Inhomogeneonus Photodynamics of Canonical Bacteriophytochrome*. Preprint. ChemRxiv, 2021. DOI: 10.26434/chemrxiv-2021-n8hlc-v2.

[157] S. Bailleul, K. Dedecker, P. Cnudde, L. Vanduyfhuys, M. Waroquier, and V. Van Speybroeck. “Ab Initio Enhanced Sampling Kinetic Study on MTO Ethene Methylation Reaction.” In: *Journal of Catalysis* 388 (2020), pp. 38–51. DOI: 10.1016/j.jcat.2020.04.015.

[158] C. Perego, O. Valsson, and M. Parrinello. "Chemical Potential Calculations in Non-Homogeneous Liquids." In: *The Journal of Chemical Physics* 149.7, 072305 (2018). DOI: 10.1063/1.5024631.

[159] O. Valsson and M. Parrinello. "Well-Tempered Variational Approach to Enhanced Sampling." In: *Journal of Chemical Theory and Computation* 11.5 (2015), pp. 1996–2002. DOI: 10.1021/acs.jctc.5b00076.

[160] S. Kullback and R. A. Leibler. "On Information and Sufficiency." In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86. DOI: 10.1214/aoms/1177729694.

[161] P. M. Piaggi and M. Parrinello. "Multithermal-Multibaric Molecular Simulations from a Variational Principle." In: *Physical Review Letters* 122.5, 050601 (2019). DOI: 10.1103/PhysRevLett.122.050601.

[162] F. Bach and E. Moulines. "Non-Strongly-Convex Smooth Stochastic Approximation with Convergence Rate O(1/n)." *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger. Curran Associates, Inc., 2013, pp. 773–781.

[163] J. Duchi, E. Hazan, and Y. Singer. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization." In: *Journal of Machine Learning Research* 12.61 (2011), pp. 2121–2159. ISSN: 1533-7928.

[164] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization." *3rd International Conference on Learning Representations*. 2015. arXiv: 1412.6980 [cs.LG].

[165] S. Ruder. "An Overview of Gradient Descent Optimization Algorithms" (2017). arXiv: 1609.04747 [cs.LG].

[166] J. P. Boyd. *Chebyshev and Fourier Spectral Methods*. 2nd ed. Mineola, NY: Dover Publications, 2001. ISBN: 978-0-486-41183-5.

[167] H. S. Cohl and C. MacKenzie. "Generalizations and Specializations of Generating Functions for Jacobi, Gegenbauer, Chebyshev and Legendre Polynomials with Definite Integrals." In: *Journal of Classical Analysis* 1 (2013), pp. 17–33. DOI: 10.7153/jca-03-02.

[168] R. H. Swendsen and J.-S. Wang. "Replica Monte Carlo Simulation of Spin-Glasses." In: *Physical Review Letters* 57.21 (1986), pp. 2607–2609. DOI: 10.1103/PhysRevLett.57.2607.

[169] Y. Sugita and Y. Okamoto. “Replica-Exchange Molecular Dynamics Method for Protein Folding.” In: *Chemical Physics Letters* 314.1-2 (1999), pp. 141–151. DOI: `10.1016/S0009-2614(99)01123-9`.

[170] D. J. Earl and M. W. Deem. “Parallel Tempering: Theory, Applications, and New Perspectives.” In: *Physical Chemistry Chemical Physics* 7.23 (2005), pp. 3910–3916. DOI: `10.1039/b509983h`.

[171] W. R. Gilks, G. O. Roberts, and E. I. George. “Adaptive Direction Sampling.” In: *The Statistician* 43.1 (1994), pp. 179–189. DOI: `10.2307/2348942`.

[172] Q. Liu and D. Wang. “Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm.” *Advances in Neural Information Processing Systems*. Vol. 29. Curran Associates, Inc., 2016.

[173] C. Camilloni, D. Provasi, G. Tiana, and R. A. Broglia. “Exploring the Protein G Helix Free-Energy Surface by Solute Tempering Metadynamics.” In: *Proteins: Structure, Function, and Bioinformatics* 71.4 (2007), pp. 1647–1654. DOI: `10.1002/prot.21852`.

[174] K. Minoukadeh, C. Chipot, and T. Lelièvre. “Potential of Mean Force Calculations: A Multiple-Walker Adaptive Biasing Force Approach.” In: *Journal of Chemical Theory and Computation* 6.4 (2010), pp. 1008–1017. DOI: `10.1021/ct900524t`.

[175] A. P. Bhati, S. Wan, D. W. Wright, and P. V. Coveney. “Rapid, Accurate, Precise, and Reliable Relative Free Energy Prediction Using Ensemble Based Thermodynamic Integration.” In: *Journal of Chemical Theory and Computation* 13.1 (2017), pp. 210–222. DOI: `10.1021/acs.jctc.6b00979`.

[176] S. Bruckner and S. Boresch. “Efficiency of Alchemical Free Energy Simulations. I. A Practical Comparison of the Exponential Formula, Thermodynamic Integration, and Bennett’s Acceptance Ratio Method.” In: *Journal of Computational Chemistry* 32.7 (2010), pp. 1303–1319. DOI: `10.1002/jcc.21713`.

[177] A. B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Series in Statistics. New York, NY: Springer, 2009. DOI: `10.1007/b13794`.

[178] I. Daubechies. “Wavelets and Other Phase Space Localization Methods.” *Proceedings of the International Congress of Mathematicians*. Ed. by S. D. Chatterji. Basel: Birkhäuser, 1995, pp. 57–74. DOI: `10.1007/978-3-0348-9078-6_8`.

[179] M. Rosenblatt. “Remarks on Some Nonparametric Estimates of a Density Function.” In: *The Annals of Mathematical Statistics* 27.3 (1956), pp. 832–837. DOI: `10.1214/aoms/1177728190`.

[180] E. Parzen. “On Estimation of a Probability Density Function and Mode.” In: *The Annals of Mathematical Statistics* 33.3 (1962), pp. 1065–1076. DOI: `10.1214/aoms/1177704472`.

[181] P. Hohenberg and W. Kohn. “Inhomogeneous Electron Gas.” In: *Physical Review* 136.3B (1964), pp. 864–871. DOI: `10.1103/PhysRev.136.B864`.

[182] W. Kohn and L. J. Sham. “Self-Consistent Equations Including Exchange and Correlation Effects.” In: *Physical Review* 140.4A (1965), pp. 1133–1138. DOI: `10.1103/PhysRev.140.A1133`.

[183] R. O. Jones. “Density Functional Theory: Its Origins, Rise to Prominence, and Future.” In: *Reviews of Modern Physics* 87.3 (2015), pp. 897–923. DOI: `10.1103/RevModPhys.87.897`.

[184] M. Valiev et al. “NWChem: A Comprehensive and Scalable Open-Source Solution for Large Scale Molecular Simulations.” In: *Computer Physics Communications* 181.9 (2010), pp. 1477–1489. DOI: `10.1016/j.cpc.2010.04.018`.

[185] C. Habermann and F. Kindermann. “Multidimensional Spline Interpolation: Theory and Applications.” In: *Computational Economics* 30.2 (2007), pp. 153–169. DOI: `10.1007/s10614-007-9092-4`.

[186] I. Daubechies. *Ten Lectures on Wavelets*. CBMS-NSF Regional Conference Series in Applied Mathematics 61. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1992. DOI: `10.1137/1.9781611970104`.

[187] V. Michel. *Lectures on Constructive Approximation: Fourier, Spline, and Wavelet Methods on the Real Line, the Sphere, and the Ball*. Applied and Numerical Harmonic Analysis. New York, NY: Birkhäuser, 2013. DOI: `10.1007/978-0-8176-8403-7`.

[188] W. Rudin. *Functional Analysis*. 2nd ed. International Series in Pure and Applied Mathematics. New York: McGraw-Hill, 1991. ISBN: 978-0-07-054236-5.

[189] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. 2nd ed. Wellesley, MA: Wellesley-Cambridge Press, 1997. ISBN: 978-0-9614088-7-9.

[190] J.-P. Kahane and P. G. Lemarié-Rieusset. *Fourier Series and Wavelets*. Studies in the Development of Modern Mathematics 3. Luxembourg: Gordon and Breach, 1995. ISBN: 978-2-88124-993-8.

[191] S. Goedecker. *Wavelets and Their Application: For the Solution of Partial Differential Equations in Physics*. Lausanne, CH: Presses Polytechniques et Universitaires Romandes, 1998. ISBN: 978-2-88074-398-7.

[192] S. Mallat. *A Wavelet Tour of Signal Processing*. 3rd ed. San Diego, CA: Academic Press, 2009. ISBN: 978-0-12-466606-1.

[193] A. Cohen. *Numerical Analysis of Wavelet Methods*. 1st ed. Studies in Mathematics and Its Applications 32. Amsterdam, NL; Boston, MA: Elsevier, 2003. ISBN: 978-0-444-51124-9.

[194] J. Gomes and L. Velho. *From Fourier Analysis to Wavelets*. Cham, CH: Springer International Publishing, 2015. DOI: `10.1007/978-3-319-22075-8`.

[195] M. Pollicott and H. Weiss. "How Smooth Is Your Wavelet? Wavelet Regularity via Thermodynamic Formalism." In: *Communications in Mathematical Physics* 281.1 (2008), pp. 1–21. DOI: `10.1007/s00220-008-0457-x`.

[196] Y. Crespo, F. Marinelli, F. Pietrucci, and A. Laio. "Metadynamics Convergence Law in a Multidimensional System." In: *Physical Review E* 81.5, 055701(R) (2010). DOI: `10.1103/PhysRevE.81.055701`.

[197] M. McGovern and J. de Pablo. "A Boundary Correction Algorithm for Metadynamics in Multiple Dimensions." In: *The Journal of Chemical Physics* 139.8, 084102 (2013). DOI: `10.1063/1.4818153`.

[198] B. Pampel and O. Valsson. *Dataset for "Enhanced Sampling with Wavelet-Based Bias Potentials"*. Zenodo, 2022. DOI: `10.5281/zenodo.5851773`.

[199] B. Pampel and O. Valsson. "Enhanced Sampling with Wavelet-Based Bias Potentials" (2022, under preparation).

[200] S. Wolfe, H. B. Schlegel, I. G. Csizmadia, and F. Bernardi. "Chemical Dynamics of Symmetric and Asymmetric Reaction Coordinates." In: *Journal of the American Chemical Society* 97.8 (1975), pp. 2020–2024. DOI: `10.1021/ja00841a005`.

[201] W. Quapp. "A Growing String Method for the Reaction Pathway Defined by a Newton Trajectory." In: *The Journal of Chemical Physics* 122.17, 174106 (2005). DOI: `10.1063/1.1885467`.

[202] S. Plimpton. “Fast Parallel Algorithms for Short-Range Molecular Dynamics.” In: *Journal of Computational Physics* 117.1 (1995), pp. 1–19. DOI: `10.1006/jcph.1995.1039`.

[203] R. Demichelis, P. Raiteri, J. D. Gale, D. Quigley, and D. Gebauer. “Stable Prenucleation Mineral Clusters Are Liquid-like Ionic Polymers.” In: *Nature Communications* 2.1, 590 (2011). DOI: `10.1038/ncomms1604`.

[204] P. Raiteri, R. Demichelis, and J. D. Gale. “Thermodynamically Consistent Force Field for Molecular Dynamics Simulations of Alkaline-Earth Carbonates and Their Aqueous Speciation.” In: *The Journal of Physical Chemistry C* 119.43 (2015), pp. 24447–24458. DOI: `10.1021/acs.jpcc.5b07532`.

[205] Y. Wu, H. L. Tepper, and G. A. Voth. “Flexible Simple Point-Charge Water Model with Improved Liquid-State Properties.” In: *The Journal of Chemical Physics* 124.2, 024503 (2006). DOI: `10.1063/1.2136877`.

[206] H. Sun, Z. Jin, C. Yang, R. L. C. Akkermans, S. H. Robertson, N. A. Spenley, S. Miller, and S. M. Todd. “COMPASS II: Extended Coverage for Polymer and Drug-like Molecule Databases.” In: *Journal of Molecular Modeling* 22.2, 47 (2016). DOI: `10.1007/s00894-016-2909-0`.

[207] J. Mei, J. W. Davenport, and G. W. Fernando. “Analytic Embedded-Atom Potentials for Fcc Metals: Application to Liquid and Solid Copper.” In: *Physical Review B* 43.6 (1991), pp. 4653–4658. DOI: `10.1103/PhysRevB.43.4653`.

[208] M. E. Tuckerman, J. Alejandre, R. López-Rendón, A. L. Jochim, and G. J. Martyna. “A Liouville-operator Derived Measure-Preserving Integrator for Molecular Dynamics Simulations in the Isothermal–Isobaric Ensemble.” In: *Journal of Physics A: Mathematical and General* 39.19 (2006), pp. 5629–5651. DOI: `10.1088/0305-4470/39/19/S18`.

[209] A. Grossfield, P. N. Patrone, D. R. Roe, A. J. Schultz, D. Siderius, and D. M. Zuckerman. “Best Practices for Quantification of Uncertainty and Sampling Quality in Molecular Simulations.” In: *Living Journal of Computational Molecular Science* 1.1, 5067 (2019). DOI: `10.33011/livecoms.1.1.5067`.

[210] L. Martino, V. Elvira, and F. Louzada. “Effective Sample Size for Importance Sampling Based on Discrepancy Measures.” In: *Signal Processing* 131 (2017), pp. 386–401. DOI: `10.1016/j.sigpro.2016.08.025`.

[211] S. Bertoluzza and S. Falletta. "Building Wavelets on]0,1[at Large Scales." In: *Journal of Fourier Analysis and Applications* 9.3 (2003), pp. 261–288. DOI: 10.1007/s00041-003-0014-0.

[212] M. Bachmayr. "Integration of Products of Gaussians and Wavelets with Applications to Electronic Structure Calculations." In: *SIAM Journal on Numerical Analysis* 51.5 (2013), pp. 2491–2513. DOI: 10.1137/120874424.

[213] B. Alpert, G. Beylkin, D. Gines, and L. Vozovoi. "Adaptive Solution of Partial Differential Equations in Multiwavelet Bases." In: *Journal of Computational Physics* 182.1 (2002), pp. 149–190. DOI: 10.1006/jcph.2002.7160.

[214] G. C. Donovan, J. S. Geronimo, and D. P. Hardin. "Orthogonal Polynomials and the Construction of Piecewise Polynomial Smooth Wavelets." In: *SIAM Journal on Mathematical Analysis* 30.5 (1999), pp. 1029–1056. DOI: 10.1137/S0036141096313112.

[215] G. C. Donovan, J. S. Geronimo, and D. P. Hardin. "Intertwining Multiresolution Analyses and the Construction of Piecewise-Polynomial Wavelets." In: *SIAM Journal on Mathematical Analysis* 27.6 (1996), pp. 1791–1815. DOI: 10.1137/S0036141094276160.

[216] Bad Religion. *Markovian Process*. Lyrics written by G. Graffin. In: *Stranger Than Fiction*. CD, Atlantic Records. 1994.

[217] B. Dünweg. "Langevin Methods." *Computer Simulations of Surfaces and Interfaces*. Ed. by B. Dünweg, D. P. Landau, and A. I. Milchev. Dordrecht: Springer Netherlands, 2003, pp. 77–92. DOI: 10.1007/978-94-010-0173-1_4.

[218] H. Risken. *The Fokker-Planck Equation*. Ed. by H. Haken. Vol. 18. Springer Series in Synergetics. Springer Berlin Heidelberg, 1984. DOI: 10.1007/978-3-642-96807-5.

[219] B. Pampel, S. Holbach, L. Hartung, and O. Valsson. "Sampling Rare-Event Energy Landscapes via a Birth-Death Process" (2022, under preparation).

[220] B. Pampel. *Birth-Death Augmented Langevin Dynamics Sampler [Software]*. Version v0.3.1, 2022. DOI: 10.5281/zenodo.5874773.

[221] B. Pampel, S. Holbach, L. Hartung, and O. Valsson. *Dataset for "Sampling Rare-Event Energy Landscapes via a Birth-Death Process"*. Zenodo, 2022. DOI: 10.5281/zenodo.5873265.

[222] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Monographs on Statistics and Applied Probability 26. Boca Raton, FL: Chapman & Hall, 1998. ISBN: 978-0-412-24620-3.

[223] D. Chandler. *Introduction to Modern Statistical Mechanics*. New York, NY: Oxford University Press, 1987. ISBN: 978-0-19-504276-4.

[224] Ampere. *Remain Unadapted*. In: *All Our Tomorrows End Today*. 10" Vinyl, Ebullition Records. 2004.

INDEX

A

APPENDIX

A.1 METADYNAMICS ALGORITHMS IN PSEUDOCODE

Here, exemplary steps of the metadynamics method are given in commented pseudocode. Algorithm 2 gives the basic algorithm that was introduced by Laio and Parinello [105] and described in Section 3.1.1 of this work. Algorithm 3 presents the well-tempered variant, where the height of the Gaussians is reduced during the course of the simulation. It was introduced by Barducci et al. [106] and described in Section 3.1.2.

Algorithm 2: Metadynamics
Exemplary steps of a metadynamics simulation as introduced in Ref. 105

```
Input: $N_T$: Number of time steps
       $N_G$: Stride (in time steps) between depositing Gaussians
       $\vec{\boldsymbol{r}} = \{\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N\}$: Coordinates of particles
       $U(\vec{\boldsymbol{r}})$: Potential energy function
       $D(\vec{\boldsymbol{r}}, \vec{\mathbf{f}})$: Propagator of coordinates (MD, LD)
       $\boldsymbol{s} = \{s_1(\vec{\boldsymbol{r}}), s_2(\vec{\boldsymbol{r}}), \dots, s_n(\vec{\boldsymbol{r}})\}$: Collective variables
       $h$: Fixed height of Gaussians
       $\boldsymbol{\sigma} = \{\sigma_1, \sigma_2, \dots, \sigma_n\}$: Width of Gaussians per collective variable

$t \leftarrow 1$                                              // Current time step
$t_G \leftarrow 1$                             // Time steps since last added Gaussian
$n_G \leftarrow 0$                                     // Counter of added Gaussians
while $t \leq N_T$ do                                         // Loop over time steps
    $u \leftarrow U(\vec{\boldsymbol{r}})$                     // Calculate current potential energy
    $\vec{f}_i \leftarrow -\nabla_{\vec{r}_i} u$             // Calculate forces from potential energy
    $\boldsymbol{s} = \{s_1(\vec{\boldsymbol{r}}), s_2(\vec{\boldsymbol{r}}), \dots, s_n(\vec{\boldsymbol{r}})\}$      // Update CV values
    $v_G \leftarrow \sum_{k=1}^{n_G} h \exp\left(-\sum_{j=1}^{n} \frac{(s_j - \tilde{s}_{k,j})^2}{2\sigma_j^2}\right)$      // Calculate Metad bias
    $\vec{f}_{G,i} \leftarrow -\nabla_{\vec{r}_i} v_G$                // Calculate Metad biasing force
    if $t_G = N_G$ then              // Metad stride is matched: add Gaussian
        $n_G \leftarrow n_G + 1$                       // Update counter of Gaussians
        $\tilde{\mathbf{s}}_{n_G} \leftarrow \mathbf{s}$                   // New Gaussian at current CV value
        $t_G \leftarrow 0$             // Reset counter since last Gaussian added
    end if
    $\vec{\mathbf{x}} \leftarrow D(\vec{\mathbf{x}}, \vec{\mathbf{f}} + \vec{\mathbf{f}}_G)$                     // Propagate coordinates
    $t \leftarrow t + 1$
    $t_G \leftarrow t_G + 1$
end while
```

Algorithm 3: Well-tempered metadynamics
Exemplary steps of a well-tempered metadynamics simulation as introduced in Ref. 106

```
Input: $N_T$: Number of time steps
       $N_G$: Stride (in time steps) between depositing Gaussians
       $\vec{\boldsymbol{r}} = \{\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N\}$: Coordinates of particles
       $U(\vec{\boldsymbol{r}})$: Potential energy function
       $D(\vec{\boldsymbol{r}}, \vec{\mathbf{f}})$: Propagator of coordinates (MD, LD)
       $\boldsymbol{s} = \{s_1(\vec{\boldsymbol{r}}), s_2(\vec{\boldsymbol{r}}), \dots, s_n(\vec{\boldsymbol{r}})\}$: Collective variables
       $h$: Fixed height of Gaussians
       $\boldsymbol{\sigma} = \{\sigma_1, \sigma_2, \dots, \sigma_n\}$: Width of Gaussians per collective variable
       $h_0$: Initial height of Gaussians
       $\gamma$: Biasfactor
       $\beta$: Inverse thermal energy of system ($\beta^{-1} = k_{\mathrm{B}}T$)

$t \leftarrow 1$                                                // Current time steps
$t_G \leftarrow 1$                                 // Time steps since last added Gaussian
$n_G \leftarrow 0$                                       // Counter of added Gaussians
while $t \leq N_T$ do                                           // Loop over time steps
    $U(\vec{\mathbf{x}})$                                  // Calculate current potential energy
    $\vec{F} \leftarrow -\nabla_{\vec{\mathbf{x}}} U(\vec{\mathbf{x}})$                     // Calculate force from potential energy
    $\mathbf{s} \leftarrow \{s_1(\vec{\mathbf{x}}), s_2(\vec{\mathbf{x}}), \dots, s_n(\vec{\mathbf{x}})\}$                    // Update CV values
    $V_G(\mathbf{s}) \leftarrow \sum_{k=1}^{n_G} h_k \exp\left(-\sum_{j=1}^{n} \frac{(s_j - \tilde{s}_{k,j})^2}{2\sigma_j^2}\right)$     // Calculate Metad bias
    $\vec{F}_G \leftarrow -\nabla_{\vec{\mathbf{x}}} V_G(\mathbf{s})$                       // Calculate Metad biasing force
    if $t_G = N_G$ then                     // Metad stride is matched: add Gaussian
        $n_G \leftarrow n_G + 1$                           // Update counter of Gaussians
        $\tilde{\mathbf{s}}_{n_G} \leftarrow \mathbf{s}$                            // New Gaussian at current CV value
        $h_{n_G} \leftarrow h_0 \cdot \exp\left(-\frac{1}{\gamma-1}\beta V_G(\mathbf{s})\right)$         // Height of new Gaussian
        $t_G \leftarrow 0$                     // Reset counter since last Gaussian added
    end if
    $\vec{\mathbf{x}} \leftarrow D(\vec{\mathbf{x}}, U(\vec{\mathbf{x}}) + V_G(\mathbf{s}), \vec{F} + \vec{F}_G)$          // Propagate coordinates
    $t \leftarrow t + 1$
    $t_G \leftarrow t_G + 1$
end while
```

A.2 SOURCE CODE FOR THE GENERATION OF DAUBECHIES WAVELETS

```python
#!/usr/bin/env python3
"""
    Calculate the scaling function and its derivative for Daubechies Wavelets
    Can also be used to print out the filter coefficients

    Copyright (C) 2021 Benjamin Pampel

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.
"""

import argparse
import numpy as np
from scipy.special import comb


def parse_args():
    """Get cli args"""
    parser = argparse.ArgumentParser()
    parser.add_argument("-N", "--moments",
                        required=True, type=int,
                        help="Number of vanishing moments of the wavelet",
    )
    parser.add_argument("-r", "--recursion",
                        type=int, default=6,
                        help="Depth of recursion,\
                              i.e. 2**d points will be calculated per integer.\n\
                              Defaults to 6 (64 points per int)."
    )
    parser.add_argument("-d", "--derivs",
                        type=int, default=0,
                        help="Number of derivatives to also calculate.\n\
                              Defaults to 0 (no derivatives)."
    )
    parser.add_argument("-f", "--filename",
                        help="Name of the output file.\n\
                              Defaults to DbN.data if wavelets are calculated.",
    )
    parser.add_argument("-c", "--coeffs",
                        action="store_true",
                        help="Only calculate the filter coefficients.\n\
                              Output is by default to screen\
                              but can be to file if the -f flag is given.",
    )
```

```
    parser.add_argument("-n", "--coeffsnorm",
                        type=float, default=np.sqrt(2),
                        help="Normalization of the filter coefficients to be printed.\n\
                              Will be ignored if the -c flag is not specified.\n\
                              Defaults to sqrt(2).",
    )
    return parser.parse_args()


def filter_coeffs(N, norm=np.sqrt(2)):
    """Calculate the filter coefficients for Daubechies Wavelets

    see Strang & Nguyen - "Wavelets and Filters", 1997, ch. 5.5
    :param p: number of vanishing moments
    :param norm: specifies the sum of the final coefficients
    """
    if N < 1:
        raise ValueError("The Wavelets must have at least 1 vanishing moment")
    if N > 34:
        raise ValueError("Sorry, the implementation does not work for N > 34")
    if norm <= 0:
        raise ValueError("The norm must be larger than 0")
    poly = np.polynomial.polynomial

    # find roots of the defining polynomial B(y)
    B_y = [comb(N + i - 1, i, exact=True) for i in range(N)]
    y = poly.polyroots(B_y)
    # calculate the roots of C(z) from the roots y via a quadratic formula
    roots = [poly.polyroots([1, 4 * yi - 2, 1]) for yi in y]
    # take the ones inside the unit circle and add roots at -1
    z = [root for pair in roots for root in pair if np.abs(root) < 1]
    z += [-1] * N
    # put together the polynomial C(z) and normalize the coefficients
    C_z = poly.polyfromroots(z)
    C_z = np.real(C_z)  # imaginary part may be non-zero because of rounding errors
    C_z *= norm / sum(C_z)
    return C_z[::-1]


def highpass_from_lowpass(h):
    """Get lowpass filter coefficients g from the highpass coefficients h"""
    alternating_sign_list = [(1 if i % 2 == 0 else -1) for i, _ in enumerate(h)]
    g = h[::-1] * alternating_sign_list
    return g


def m_matrices(h):
    """Calculate the matrices M0 and M1 for the vector cascade

    :param h: filter coefficients, with norm 1
    :return [M0, M1]: matrices
    """
    # initialize matrices to zero
    n = len(h) - 1
```

```python
    M0 = np.zeros((n, n), dtype=np.float64)
    M1 = np.zeros((n, n), dtype=np.float64)
    # 'c style' construction
    for i in range(n):
        for j in range(n):
            if 0 <= 2 * i - j <= n:
                M0[i, j] = 2 * h[2 * i - j]
            if -1 <= 2 * i - j <= n - 1:
                M1[i, j] = 2 * h[2 * i - j + 1]
    return [M0, M1]


def normalize_eigenvector(eigvec, derivnum):
    """Normalize eigenvector to obtain values at the integers

    :param eigvec: eigenvector of the M0 matrix
    :param derivnum: derivative currently calculated
    :return: phi at the integer values
    """
    weighted_sum = np.sum(eigvec * ((-np.arange(len(eigvec))) ** derivnum))
    norm = np.math.factorial(derivnum) / weighted_sum
    return eigvec * norm


def wavelet(N, d=6, derivs=0):
    """Calculate the scaling and wavelet function for Daubechies Wavelets

    :param N: number of vanishing moments N
    :param d: recursion number. returned array will have 2**d values per integer
    :param derivs: number of derivatives to also calculate
    :return (x, phi, psi): phi is the scaling and psi the wavelet function
                           both are lists containing the function and derivatives
    """
    if derivs > N - 1:
        raise ValueError(f"Only {N-1} derivatives exist but {derivs} were requested")

    h = filter_coeffs(N, 1)
    g = highpass_from_lowpass(h)

    H = m_matrices(h)
    G = m_matrices(g)

    step = 1 << d  # number of values between integers
    # set up arrays of values to be calculated
    phi = np.empty((N, (2 * N - 1) * step), dtype=np.float64)
    psi = np.empty((N, (2 * N - 1) * step), dtype=np.float64)

    # get eigenvalues and vectors of matrix
    H0_eigvals, H0_eigvecs = np.linalg.eig(H[0])

    # identify the indices corresponding to the required derivatives in H0_eigvals
    #   and H0_eigvecs
    # the eigenvalues are 2**(-deriv) starting with 0
    # prefix 'dy' is for the dyadic values
```

```
    dy_eigval_indices = [
        np.argmin(np.absolute(H0_eigvals - 2 ** (-j))) for j in range(derivs + 1)
    ]

    values_at_int = np.empty((N, 2 * N - 1), dtype=np.float64)

    for j, k in enumerate(dy_eigval_indices):
        # j is order of derivative (0:derivs), k the position of the corresponding
        #   eigenvector
        values_at_int[j] = normalize_eigenvector(H0_eigvecs[:, k], j)

        # multiply matrices with factor (less flops)
        factor = 1 << j
        H_temp = [factor * H[0], factor * H[1]]
        G_temp = [factor * G[0], factor * G[1]]

        # fill first two datasets by hand
        binarydict = {"0": values_at_int[j]}
        binarydict["1"] = H_temp[1] @ values_at_int[j]
        phi[j][::step] = binarydict["0"]
        phi[j][step >> 1 :: step] = binarydict["1"]
        psi[j][::step] = G_temp[0] @ values_at_int[j]
        psi[j][step >> 1 :: step] = G_temp[1] @ values_at_int[j]

        # do the recursion
        oldbits = ["1"]
        for depth in range(2, d + 1):
            newbits = ["%d%s" % (new, old) for new in [0, 1] for old in oldbits]
            for binary in newbits:
                start = int(binary, 2) * step >> depth
                firstbit = int(binary[0])
                binarydict[binary] = H_temp[firstbit] @ binarydict[binary[1:]]
                phi[j][start::step] = binarydict[binary]
                psi[j][start::step] = G_temp[firstbit] @ binarydict[binary[1:]]
            oldbits = newbits

    # corresponding x values
    x = np.arange(0, 2 * N - 1, 1 / step)

    return (x, phi, psi)


if __name__ == "__main__":
    args = parse_args()

    if args.coeffs:  # print only file coeffs
        coeffs = filter_coeffs(args.moments, args.coeffsnorm)
        if args.filename is None:  # print to screen
            [print("{:.32e}".format(c)) for c in coeffs][0]
        else:
            header = "#! FIELDS coeff"
            header += "\n#! SET type Db" + str(args.moments)
            np.savetxt(args.filename, coeffs.T, header=header)

```

```
    else:  # do the main program, i.e. calculate and print the wavelet functions
        x, scaling_func, wavelet = wavelet(args.moments, args.recursion, args.derivs)
        if args.filename is None:  # default names
            args.filename = f"Db{args.moments}.data"
        header = "#! FIELDS x Phi Psi"
        header += "".join([f" Phi_d{i} Psi_d{i}" for i in range(1, args.derivs + 1)])
        header += "\n#! SET type Db" + str(args.moments)

        # collect all data into one array
        data = np.empty((3 + args.derivs * 2, len(x)))
        data[0] = x
        for deriv in range(args.derivs + 1):
            data[2 * deriv + 1] = scaling_func[deriv]
            data[2 * deriv + 2] = wavelet[deriv]
        np.savetxt(args.filename, data.T, header=header)
```