Mean Force Integration: A Unified Framework for Merging Independent Simulations subject to Various Bias Potentials

Antoniu Bjola

A dissertation submitted in compliance
with the requirements for the degree of
Doctor of Philosophy
of
University College London.

Department of Chemical Engineering
University College London

Under the supervision of
Dr. Matteo Salvalaglio

July 22, 2025

I, Antoniu Bjola, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Molecular dynamics (MD) simulation has become a powerful tool for studying and predicting molecular properties due to significant algorithm advances and the explosive growth of computational capabilities. Moreover, molecular dynamics enables the direct evaluation of free energy surfaces (FESs), offering atomistic insight that complements experimental studies and enables the prediction of numerous thermodynamic properties.

Yet MD remains constrained by system size and accessible time scales. Furthermore, many processes of interest, such as nucleation or protein folding, are characterised by *rare events*, where considerable energy barriers impede transitions between stable states. Nevertheless, transitions must be sampled multiple times for statistically significant predictions of free energies, rendering brute force simulations unfeasible.

To address this issue, numerous methods have been proposed to enhance the sampling of rare events. Two widely used methods are umbrella sampling [1] and metadynamics [2]. The former makes use of parallel simulations that sample local predefined regions of configuration space, while the latter continually constructs a bias potential that facilitates the sampling of high-energy configurations. A new method called mean force integration (MFI) [3], which works on the basis of metadynamics, computes the mean force rather than the FES directly, thereby simplifying reweighting and accelerating convergence. Additionally, it can be used to combine independent simulations, turning a serial problem into a parallel one, which increases computational efficiency.

This thesis advances MFI to a versatile framework: A general formulation is presented, accommodating the combination of arbitrary static and history-dependent biases. This is complemented by an *on-the-fly* uncertainty metric that estimates the convergence of the mean force, and a bootstrap analysis that provides a quantitative assessment of the error of the FES. These advances are validated with complex chemical systems, including the nucleation of supersaturated argon vapour, the two-step crystallisation of a colloidal system, and the β-scission reaction of butyl acrylate. It is shown how the computational cost of excessively expensive simulations can be reduced by employing several shorter simulations subject to diverse biasing parameters. The resulting under-converged trajectories were analysed and combined with MFI, resulting in converged FESs. For the β-scission reaction, the FES was used to predict reaction rates, which agreed with experimental rates.

Additionally, novel reinitialisation protocols are introduced, dividing simulations into diverse biasing stages and recycling interim FES estimates as starting static biases, thereby consistently enhancing convergence of the FES. This was further developed into a framework where simulations are analysed in real time, terminated and reinitialised automatically, as biasing parameters are optimised iteratively.

To encourage a wider adoption of MFI, all the Python code used in this work is made openly accessible at `github.com/mme-ucl/MFI`. By unifying data from independent biased trajectories, enabling an iterative improvement of biasing parameters, and providing reliable convergence metrics, MFI broadens the range of phenomena that researchers can tackle.

Impact Statement

Free energy surfaces entail a quantitative landscape from which thermodynamic properties, such as molecular stability, transition states, reaction pathways, and rates, are derived. Accordingly, determining the FES accurately is essential for the *in silico* design of fine chemicals. In principle, MD simulations allow such surfaces to be extracted directly from an atomistic model. However, their reliable computation is often hampered by large energy barriers, slow molecular diffusion, and the limited timescales accessible in conventional MD.

Enhanced sampling methods partially alleviate these obstacles, but offer limited possibilities for combining independent trajectories subject to arbitrary bias potentials. Preliminary trajectories, generated with sub-optimal biasing parameters, are routinely discarded despite containing valuable information. Moreover, biasing parameters can not be adjusted dynamically, resulting in slower convergence. Furthermore, most approaches provide only coarse or *a posteriori* error estimates, so computational efforts are commonly overspent long after convergence has been reached. Conversely, insufficient sampling can remain undetected, requiring researchers to spend additional resources to prolong simulations.

This thesis advances MFI into a practical alternative that resolves those shortcomings: Multiple under-converged trajectories that employ various biasing methods can be combined in a self-consistent manner to yield a converged estimate of the FES. On the one hand, this enables a parallel execution of simulations, where computational resources are utilised more opti-

mally without the inter-process communication that constrains conventional replica methods. On the other hand, it allows for a serial workflow, where biasing parameters can be refined iteratively, and earlier simulations sampled with inadequate biasing parameters can be merged with the other data. Also, it permits a more flexible structuring of large simulation campaigns, allowing researchers to run simulations when computational allocations are available. Additionally, reliable uncertainty estimators grant more confidence in the final results and inform researchers when results converge, preventing the unnecessary use of further computational resources.

Collectively, these advances afford greater flexibility in designing and executing biased MD simulations, thereby facilitating the investigation of complex systems that were otherwise out of reach. Conversely, the computational cost of expensive systems can be reduced by employing optimised biasing strategies, thus reducing the carbon footprint for large-scale simulation campaigns.

All algorithms developed in this work are openly accessible at `github.com/mme-ucl/MFI`, encouraging a wider adoption of the MFI method and fostering broader collaboration within the enhanced sampling community. Furthermore, several examples and tutorials regarding the use of MFI are provided as Jupyter notebooks, lowering the barrier to entry for researchers wishing to try out the MFI library to run simple simulation campaigns or post-process their own data.

UCL Research Paper Declaration Form

Referencing the doctoral candidate's own published works.

1. **Research manuscript that has already been published:**

   (a) **Title of manuscript:** Estimating Free-Energy Surfaces and Their Convergence from Multiple, Independent Static and History-Dependent Biased Molecular-Dynamics Simulations with Mean Force Integration

   (b) **DOI (URL):** 10.1021/acs.jctc.4c00091 (`https://pubs.acs.org/doi/10.1021/acs.jctc.4c00091`)

   (c) **Journal:** Journal of Chemical Theory and Computation

   (d) **Publisher:** American Chemical Society

   (e) **Date of publication:** 24th of June 2024

   (f) **Authors of manuscript:** Antoniu Bjola, Matteo Salvalaglio

   (g) **Was the work peer reviewed?:** Yes

   (h) **Has the copyright been retained?:** No

   (i) **Earlier form of the manuscript uploaded to a preprint server:** DOI: 10.26434/chemrxiv-2024-83h5q-v2
   URL: https://doi.org/10.26434/chemrxiv-2024-83h5q-v2

   ☒ *I acknowledge permission of the publisher named under 1d to include in this thesis portions of the publication named as included in 1c.*

2. **Statement of contribution of all authors:** Antoniu Bjola conducted all research, performed the simulations, analysed the data, and wrote the manuscript. Matteo Salvalaglio provided supervision throughout the project, offering critical insights and guidance, and contributed to improving the manuscript through detailed feedback and revision.

3. **Material can be found in the following chapters of the thesis:** Chapters 3, 4, 6

1. **Research manuscript that has already been published:**

   (a) **Title of manuscript:** Unveiling Solvent Effects on β-Scissions through Metadynamics and Mean Force Integration

   (b) **DOI (URL):** 10.1021/acs.jctc.4c00383 (`https://pubs.acs.org/doi/10.1021/acs.jctc.4c00383`)

   (c) **Journal:** Journal of Chemical Theory and Computation

   (d) **Publisher:** American Chemical Society

   (e) **Date of publication:** 3rd of July 2024

   (f) **Authors of manuscript:** Francesco Serse, Antoniu Bjola, Matteo Salvalaglio, Matteo Pelucchi,

   (g) **Was the work peer reviewed?:** Yes

   (h) **Has the copyright been retained?:** No

   (i) **Earlier form of the manuscript uploaded to a preprint server:** DOI: 10.26434/chemrxiv-2024-rn2rz
   URL: https://doi.org/10.26434/chemrxiv-2024-rn2rz

   ☒ *I acknowledge permission of the publisher named under 1d to include in this thesis portions of the publication named as included in 1c.*

2. **Statement of contribution of all authors:** Francesco Serse designed and carried out the molecular dynamics simulations, including preparing the force fields and simulation environments. Antoniu Bjola contributed to the post-processing and analysis of the simulation data using Mean Force Integration (MFI) and assisted in optimising the biasing parameters. Matteo Salvalaglio and Matteo Pelucchi supervised the work, providing critical insights and guidance throughout the project, and contributed to improving the manuscript through feedback and revision.

3. **Material can be found in the following chapter of the thesis:** Chapter 6

1. **Research manuscript that is prepared for publication but that has not yet been published:**

   (a) **Current title of the manuscript:** Initialising Short Independent Simulations using Mean Force Integration

   (b) **Has the manuscript been uploaded to a preprint server:** No

   (c) **Where is the work intended to be published?:** Journal of Chemical Theory and Computation

   (d) **Manuscript's authors in the intended authorship order:** Antoniu Bjola, Matteo Salvalaglio

   (e) **Stage of publication:** Results of Chapter 5 need to be included in the manuscript. Further revisions may follow.

2. **Statement of contribution of all authors:** Antoniu Bjola developed the methods, performed the simulations, analysed the data, and is writing the manuscript. Matteo Salvalaglio provides supervision throughout the project, offering critical insights and guidance, and contributes to improving the manuscript through detailed feedback and revision.

3. **Material can be found in the following chapters of the thesis:** Chapter 5

e-Signatures confirming that the information above is accurate:

Candidate signature: Antoniu Bjola
Date: 29th of May 2025

Supervisor signature: Matteo Salvalaglio
Date: 29th of May 2025

Publications and Presentations

Peer-reviewed publications:

- **A. Bjola**, M. Salvalaglio, ”Estimating Free-Energy Surfaces and Their Convergence from Multiple, Independent Static and History-Dependent Biased Molecular-Dynamics Simulations with Mean Force Integration”, Journal of Chemical Theory and Computation 20 (14), 6253-6262. Available at: `https://doi.org/10.1021/acs.jctc.4c00091`.

- F. Serse, **A. Bjola**, M. Salvalaglio, M. Pelucchi, ”Unveiling solvent effects on β-scissions through metadynamics and mean force integration”, Journal of Chemical Theory and Computation 20 (14), 6253-6262. Available at: `https://doi.org/10.1021/acs.jctc.4c00383`.

Publications in progress:

- **A. Bjola**, M. Salvalaglio, ”Reinitialising Short Independent Simulations using Mean Force Integration”.

Twenty-minute oral presentations at conferences:

- Materials and Molecular Modelling (MMM) Hub Conference & User meeting, London (UK), 8-9 November 2023.
- American Chemical Society (ACS) Spring Meeting 2024, New Orleans (USA), 17-21 March 2024.
- Thomas Young Centre (TYC) Postgraduate Student Day 2024, London (UK), 22 May 2024.

Flash presentations:

- Three-Minute Thesis (3MT) Competition, Department of Chemical Engineering, UCL, London (UK), 13 March 2024.
- Chemical Engineering Day UK 2024 (ChemEngDay24), London (UK), 25-26 April 2024.

Poster presentations at conferences:

- Thermodynamics 2022 Conference, Bath (UK), 7-9 September 2022.
- Thomas Young Centre (TYC) Postgraduate Student Day 2023, London (UK), 22 May 2023.
- Sargent Centre Annual Industrial Consortium Meeting 2023, London (UK), 8 December 2023.
- Chemical Engineering Day UK 2024 (ChemEngDay24), London (UK), 25-26 April 2024.

Acknowledgements

I would like to extend my deepest gratitude to Matteo Salvalaglio, who has not only been a reliable and inspiring academic guide but also showed empathy during a time when people moved apart.

I would also like to thank my colleagues, especially Aaron Finney and Edgar Olehnovics, for the spontaneous discussions we had about molecular simulations and other aspects of life. It gave me a sense of community, which I was often missing during a long period of imposed isolation.

I would also like to thank University College London and its department of Chemical Engineering for providing the opportunity to undergo my doctoral studies at a world-class institution and providing me with countless resources and plenty of academic and non-academic support.

Last but not least, I want to express my sincerest gratitude to my friends and family, who gave me moral support, without which none of this would have happened.

Contents

List of Figures

List of Tables

List of Abbreviations

AAD	Average Absolute Deviation
ABF	Adaptive Biasing Force
AIMD	*Ab initio* Molecular Dynamics
AMBER	Assisted Model Building with Energy Refinement
BA	Butyl Acrylate
CV	Collective Variable
DFT	Density Functional Theory
EoS	Equation of State
FES	Free Energy Surface
FFT	Fast Fourier Transform
GAFF	Generalised AMBER Force Field
HPC	High-Performance Computing
InvF	Inverse Free Energy Bias
MD	Molecular Dynamics
MetaD	Metadynamics
MFI	Mean Force Integration
NVE	Microcanonical Ensemble
NVT	Canonical Ensemble
OPES	On-the-Fly Probability Enhanced Sampling
PBC	Periodic Boundary Conditions
PRTR	Parallel Real-Time Reinitialisation
SRTR	Serial Real-Time Reinitialisation
UI	Umbrella Integration
US	Umbrella Sampling
VES	Variational Enhanced Sampling
WHAM	Weighted Histogram Analysis Method
WT-MetaD	Well-Tempered Metadynamics

List of Symbols

E	Internal energy (kJ mol^{-1})
F	Helmholtz free energy (kJ mol^{-1})
H	Hamiltonian (kJ mol^{-1})
K	Kinetic energy (kJ mol^{-1})
N	Number of atoms / particles (–)
P	Pressure (Pa)
$P(\xi)$	Probability density at CV value ξ (–)
Q	Canonical partition function (–)
S	Entropy (kJ mol^{-1} K^{-1})
T	Temperature (K)
U	Potential energy (kJ mol^{-1})
V	Volume (m^3)
Z	Configurational integral (–)
Z_{NVT}	Canonical partition function at constant N, V, T (–)
h	Planck constant (J s)
k_B	Boltzmann constant (J K^{-1})
m	Atomic / particle mass (kg)
r_N	Vector of atomic positions (Å or nm)
p_N	Vector of atomic momenta (kg m s^{-1})
d	Inter-atomic distance (Å or nm)
β	Inverse thermal energy, $(k_B T)^{-1}$ (J^{-1})
ΔF	Free-energy difference between states A and B (kJ mol^{-1})
δ	Gaussian width used in MetaD (units of ξ)
γ	Bias factor in WT-MetaD (–)
κ	Force constant of harmonic restraint (kJ mol^{-1} unit^{-2})
λ	Switching parameter in TI (–)
Λ	Thermal de Broglie wavelength (m)
μ	Chemical potential (kJ mol^{-1})

$\omega(\xi)$	Harmonic bias potential (kJ mol^{-1})
ω_0	Initial height of Gaussian hills (kJ mol^{-1})
σ	Standard deviation (units of ξ)
σ_i	Standard deviation in window i (units of ξ)
N_i	Number of MD steps in window i (–)
$p_i(\xi)$	Weighted probability of window i at ξ (–)
ξ	Collective variable (system-dependent units)
ϕ	Torsional dihedral angle (rad)
ψ	Torsional dihedral angle (rad)

Chapter 1

Introduction

The prediction of molecular properties has long been a central challenge in physics, chemistry, and materials science. Early theoretical frameworks [6, 7, 8, 9] provided a foundation for understanding molecular behaviour, yet the complexity of these equations made them computationally infeasible to solve in practice. As a result, scientists primarily relied on experimental observations, while predictive methods were limited to simpler, approximate models. Solid-state properties were approximated using the foundational principles of lattice dynamics [10, 11, 12, 13] by approximating interatomic interactions as harmonic. Fluid properties were often predicted using equations of state (EoS), which modelled the relationships between pressure, volume, and temperature by fitting a small number of parameters to empirical data. These equations, such as the Van der Waals equation [14, 15] and its successors [16, 17, 18, 19, 20, 21], allowed for approximations of thermodynamic properties and phase behaviour. These approaches were particularly valuable for petrochemical companies, enabling them to model and optimise their processes. However, their inability to accurately describe complex chemical systems highlighted the need for more advanced methods that could capture the underlying physics in greater detail.

With the rapid development of computational resources, more rigorous theoretical models could be applied to real-world problems. Rather than relying on empirical approximations, researchers began implementing atom-

istic descriptions of molecular systems, employing complex potential energy functions that account for the underlying electronic structure, chemical bonds, van der Waals forces, and polarisation interactions. This shift enabled the prediction of material properties directly from first principles or highly detailed force fields, thereby aligning theoretical models with experimental observations. Whenever discrepancies appeared between the simulation and experiment, the model parameters were iteratively refined to achieve better agreement. Thanks to ever-increasing computational power, these simulations can now provide reliable predictions even under extreme conditions [22], circumventing the need for expensive or impractical experimental setups. Moreover, if the governing theory and molecular representation are carefully chosen, computer simulations offer unique insights into atomistic mechanisms, providing a level of detail inaccessible to experiments.

Over the past few decades, such molecular simulations have impacted the development of diverse fields. In biology, new insights emerged regarding the structure and conformational transitions of macromolecules [23] and understanding ligand-receptor docking mechanisms [24]. In materials science, computational methods have led to a deeper understanding of the crystallisation mechanisms [25, 26], high-energy radiation damage in nuclear applications [27], stability of perovskite structures [28, 29], and many other insights [30, 31, 32]. As hardware and algorithms continue to evolve, further breakthroughs are anticipated. Nevertheless, these methods face nontrivial challenges in capturing complex processes over relevant timescales and length scales. The following section explores limitations of molecular simulations and highlights strategies to overcome them.

Despite their proven utility, molecular simulations face inherent system size and simulation timescale limitations. The timescale limitation arises from the stringent temporal resolution required. To reliably predict the dynamics of a molecular system, the positions and forces of each atom must

be evaluated on a timescale of femtoseconds (10^{-15} seconds), such that the simulation of slow chemical processes that occur over milliseconds or longer becomes prohibitively expensive. Furthermore, as system size increases, the number of required force evaluations grows, significantly driving up the computational cost of large chemical systems.

For example, capturing repeated folding and unfolding events for the NTL9 protein, containing approximately 10,000 atoms, requires about 3 ms of simulation time [33]. Using a time-step of 2.5 fs, it would require roughly 10^{12} force evaluation steps, which would take months to years to simulate on state-of-the-art supercomputers [34].

Protein folding [23] exemplifies a broader problem in molecular simulations often referred to as simulating *rare events* [35]. Rare events are low-probability transitions between long-lived stable states. In the protein-folding case, substantial simulation time is needed to capture either the forward or backwards transition, making direct brute-force approaches impractical. Other examples of rare events include nucleation processes [36, 25] and conformational changes in bio- and macromolecules [37, 38, 28, 29, 32].

1.1 Overcoming Timescale Limitations

The practical challenge of timescales has spurred the development of methods aimed at improving both the speed and accuracy of molecular simulations, rather than conducting brute-force (conventional) simulations, *enhanced sampling techniques* are utilised, which increase the probability of sampling rare events. Although a wide range of approaches exists [39, 40, 41, 42, 43, 44, 45, 46], this work focuses on methods that perturb the system's Hamiltonian, resulting in a biased potential energy surface that enables a more advantageous sampling.

Umbrella sampling (US) [1, 37, 47, 48] is a method that perturbs the system's Hamiltonian via harmonic bias potentials. They act as an attractive bias that confines the simulation to a particular region of phase space,

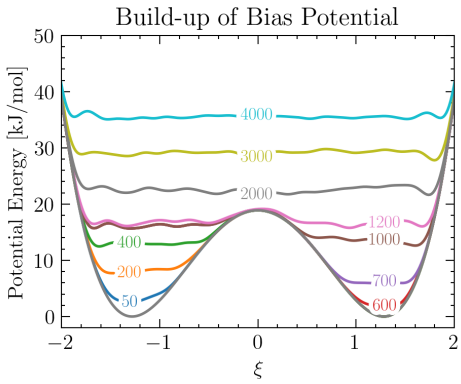


Figure 1.1: Figure illustrates the evolution of the history-dependent repulsive bias. The lines represent free energy plus the bias potential as a function of the number of Gaussian hills deposited.

also referred to as an *umbrella*. The phase space sampling is broken into multiple umbrellas, which are then sampled individually while ensuring a significant overlap. Once each region is sufficiently sampled, its free energy surface can be patched together to construct the overall free energy surface. One key advantage of this window-based approach is that simulations can be run independently of each other and completed with later simulations if needed. However, a drawback lies in selecting where and how to place the umbrellas effectively, which usually requires some *a priori* knowledge of the free energy landscape. This method is discussed in detail in section 2.4.1.

Metadynamics (MetaD) [2, 49, 50] applies a history-dependent *repulsive* bias to the system's Hamiltonian. Throughout the simulation, Gaussian potentials, also referred to as *hills*, are periodically deposited in previously visited regions, effectively discouraging the system from resampling those regions and facilitating the sampling of new configurations. By progressively filling the free energy basins, MetaD enables the simulation to overcome considerable energy barriers and explore transition paths more readily (see Fig. 1.1 for an illustration). This method is discussed in greater detail in section 2.4.3 and 2.4.4.

Adaptive Biasing Force (ABF) [51, 52, 53] is an enhanced sampling technique that aims to flatten free energy barriers along a chosen reac-

tion coordinate. Unlike MetaD, ABF directly operates on forces to iteratively counteract the free energy gradient. By continuously measuring the average force acting on the system as it explores a given coordinate, ABF constructs a compensating bias potential that counteracts this mean force. As a result, the underlying free energy landscape becomes more uniform, enabling the simulation to traverse high-barrier regions more readily. This iterative process proceeds without prior knowledge of the shape of the free-energy profile, allowing for efficient and systematic exploration of complex configurations. Even though this is an effective tool for sampling rare events, it lies outside the scope of this work and will not be discussed in more detail.

Parallelisation in MD simulations is not limited to distributing computational workload [54, 55] but can also involve parallel sampling methods. A practical approach is Replica Exchange, where multiple replicas of the same system are simulated in parallel under different conditions. Periodically, exchanges between replicas are attempted based on a Metropolis criterion, allowing configurations to swap conditions and facilitating the exploration of phase space. A widely used variant is *Parallel Tempering* [56], in which replicas differ only in temperature. Higher-temperature replicas sample a broader range of configurations, while lower-temperature replicas capture more detail. This method is particularly effective for overcoming large energy barriers and can be combined with enhanced sampling techniques such as Metadynamics, Umbrella Sampling, ABF, or variationally enhanced sampling (VES).

Another parallel approach is the *Multiple Walkers* method, where several independent simulations, also referred to as “walkers”, run in parallel. Each walker explores a different portion of phase space and periodically communicates its progress to the other walkers. When used alongside Metadynamics, for instance, all walkers contribute simultaneously to constructing the bias potential, ensuring rapid exploration of relevant configurations.

While parallelisable simulation methods are widely used for their effectiveness, they require communication or synchronisation among parallel processes, confining them to a single computing environment. This constraint can become problematic for researchers dealing with large-scale or *ab initio* molecular dynamics (AIMD) simulations who may lack access to suitable high-performance computing (HPC) resources.

In such scenarios, a straightforward alternative: Rather than exchanging information in real time, each simulation runs independently, and the results are combined only during post-processing. Methods like Umbrella Sampling (US) are well suited to this approach, as each “window” can be simulated in isolation without additional communication overhead. Once all windows have been sampled sufficiently, their free energy surfaces can be merged to obtain a global free energy surface, allowing large-scale problems to be tackled with minimal infrastructure requirements.

A known challenge in umbrella sampling arises during the post-processing step, where individual windows are patched together to reconstruct a global free-energy surface (FES). When the overlap between neighbouring windows is weak, standard techniques often perform poorly. To address this issue, numerical optimisation approaches such as the weighted histogram analysis method (WHAM) [57, 58] or the (multi-scale) Bennett acceptance ratio [59, 60, 46]. Kästner introduced an alternative method: *Umbrella Integration* (UI) [47, 61, 62], which computes the mean force of each umbrella window rather than the free energy directly. After all simulations are completed and analysed, their local mean forces are combined through a weighted-average scheme and then integrated to recover the global FES. This approach enhances robustness in cases of sparse overlap between neighbouring windows and yields more reliable results [63, 64]. A detailed examination of UI is provided in Section 2.4.2.

1.2 Aim of this Project

Building on Kästner's ideas, Marinova and Salvalaglio developed *Mean Force Integration* (MFI) [3], a post-processing technique that similarly computes mean forces rather than free energies. Their work showed that MFI can combine mean forces from independent MetaD simulations through a weighted-average approach. Unlike other parallel MetaD methods, MFI offers a self-consistent way to merge results from independent simulations, thus facilitating the parallelisation of MetaD simulations.

The primary goal of this project is to develop further and optimise the mean force integration (MFI) method, ultimately creating a versatile post-processing framework for a wide range of biased MD simulations. To this end, the work focuses on the following objectives:

- **Generalised Framework:** Leverage the commonalities among Metadynamics (MetaD), Umbrella Sampling (US), and other biased techniques to build a generalised framework capable of analysing data from each method, and any arbitrary bias potential, separately or combined.

- **Convergence Estimation:** Develop a robust methodology for assessing local and global convergence by examining the statistical variance in the mean force and the free-energy surface. When exact solutions are available, compare the estimated convergence to the actual deviation to validate the approach.

- **Flexible Simulation Campaigns:** Explore and compare different strategies for running simulations. Simulations either run concurrently to exploit parallel efficiencies, serially to reuse information from prior runs and iteratively improve biasing parameters, or in a hybrid parallel-serial approach.

- **Real-Time Adaptation:** Integrate the above components into an adaptive framework where simulations are monitored in real time.

Convergence metrics are continually evaluated, allowing simulations to be automatically terminated and new ones initiated in their place.

- **Validation against Complex Chemical Systems:** Test the methods introduced in this work on three advanced systems, which would be exceedingly expensive to simulate with conventional enhanced sampling methods.

To promote broader adoption, all MFI methods employed in this work are publicly accessible via the pyMFI Python library at `github.com/mme-ucl/MFI`, together with a stable version that contains the core functionalities at `github.com/mme-ucl/pyMFI`. The library enables researchers to post-process and combine biased simulation data. In addition, various use cases, simple examples, and many of the applications presented in this work are accessible as Jupiter notebooks within the repository and are further described in Appendix B.

1.3 Outline of Report

In the subsequent Chapter 2, the theoretical background of this work is presented, starting with core concepts of statistical mechanics and molecular dynamics simulations, followed by methods to calculate the free energy and biased sampling techniques.
Thereafter, Chapter 3 outlines the new developments of MFI, which are tests on analytical models using Langevin dynamics and on alanine dipeptide.
In Chapter 4, the new developments of MFI are applied to analyse two nucleating systems: The condensation of supersaturated argon vapour, and the two-step crystallisation of a colloidal system.
In Chapter 5, the MFI methods are further extended by presenting techniques to analyse simulations in real-time and re-initialise simulations in an effective manner. These techniques are used to conduct simulation campaigns that employ multiple short simulations, and are tested on analytical models and alanine dipeptide.

Next, Chapter 6 shows how MFI is used to analyse and combine multiple simulations of the β-scission of butyl acrylate. The combined FESs are used to predict reaction rates, which are compared to experimentally measured rates.

Lastly, concluding remarks are given and an outlook on future developments is provided.

Chapter 2

Theoretical Background

This chapter starts by presenting the fundamental principles of statistical mechanics, bridging microscopic atomic behaviour with macroscopic thermodynamic properties. Next, the core concepts of molecular dynamics simulations are discussed, covering initialisation, evaluation of forces calculation, and control mechanisms. Following this, two pioneering theories for calculating free energy differences are introduced. The discussion then transitions to the challenges posed by large energy barriers that hamper accurate sampling, setting the stage for enhanced sampling techniques, including umbrella sampling and metadynamics. Finally, the mean force integration method is presented as a flexible approach to reconstructing free energy landscapes.

2.1 Statistical Mechanics

When investigating molecular systems in computer simulations, it is possible to visualise the evolution of all atoms. However, researchers are usually more interested in macroscopic properties, such as temperature or free energy, which cannot be directly extracted from raw simulation data. In an experimental setup, for instance, the temperature of a system is measured by averaging the interactions of many molecules with a thermometer. Analogously, in molecular simulations, one obtains the temperature from the average kinetic energy of the atoms. This section introduces how such

macroscopic properties can be derived through the framework of statistical mechanics.

2.1.1 Equations of Motion

A molecular system is defined by the position and velocity of all atoms. The position, defined in a three-dimensional Cartesian space, can be expressed as:

$$\mathbf{r}_i(t) = \big(x_i(t), y_i(t), z_i(t)\big) \tag{2.1}$$

where $\mathbf{r}_i(t)$ is the position of the atom i as a function of time, t and x_i, y_i, z_i are the respective coordinates in Cartesian space. The velocity, $\mathbf{v}_i(t)$, is defined as the time derivative of the position:

$$\mathbf{v}_i(t) = \frac{d\mathbf{r}_i(t)}{dt} = \left(\frac{dx_i(t)}{dt}, \frac{dy_i(t)}{dt}, \frac{dz_i(t)}{dt}\right), \tag{2.2}$$

and the acceleration, $\mathbf{a}_i(t)$, as the second time derivative of the position:

$$\mathbf{a}_i(t) = \frac{d\mathbf{v}_i(t)}{dt} = \frac{d^2\mathbf{r}_i(t)}{dt^2} = \left(\frac{d^2x_i(t)}{dt^2}, \frac{d^2y_i(t)}{dt^2}, \frac{d^2z_i(t)}{dt^2}\right). \tag{2.3}$$

With this information, the evolution of the system is determined using classical Mechanics, particularly Newton's Second Law of Motion: The force acting on an object is equal to the object's mass times the acceleration it undergoes [65]:

$$\mathbf{F}_i(t) = m_i\mathbf{a}_i(t), \tag{2.4}$$

where $\mathbf{F}_i(t)$ is the force acting on the atom i, and m_i is the mass of the atom i. Equations 2.3 and 2.4 are the Equations of Motion and can be used to calculate the evolution of the system. However, in the context of molecular systems, the equations of motion are usually expressed with the Hamiltonian, $\mathcal{H}(\mathbf{r}, \mathbf{p})$ [48]:

$$\mathcal{H}(\mathbf{r},\mathbf{p}) = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i} + U(\mathbf{r}), \tag{2.5}$$

where N is the total number of atoms and $\mathbf{r}$ and $\mathbf{p}$ are short notations indicating the position of all atoms $\{\mathbf{r}_1(t), \mathbf{r}_2(t), \ldots, \mathbf{r}_N(t)\}$ and their conjugate momenta $\{\mathbf{p}_1(t), \mathbf{p}_2(t), \ldots, \mathbf{p}_N(t)\}$, respectively. The second term, $U(\mathbf{r})$, represents the potential energy of the system. In contrast, the first term describes the kinetic energy, which is expressed as a function of the conjugate momenta to ensure compatibility with generalised coordinate systems. In a Cartesian coordinate system, for example, it would be defined as $\mathbf{p}_i = m_i\mathbf{v}_i$. Expressing the equation of motion in this manner is practical because it conserves the total energy of a closed system and naturally accommodates coordinate transformations. Moreover, taking the partial derivatives of equation 2.5 leads to Hamiltonian's Equations of Motion:

$$\frac{\partial \mathcal{H}}{\partial \mathbf{p}_i} = \frac{\mathbf{p}_i}{m_i} = \frac{d\mathbf{r}_i}{dt}, \tag{2.6}$$

$$-\frac{\partial \mathcal{H}}{\partial \mathbf{r}_i} = -\frac{\partial U(\mathbf{r})}{\partial \mathbf{r}_i} = \mathbf{F}_i(\mathbf{r}) = \frac{d\mathbf{p}_i}{dt} = m_i\mathbf{a}_i. \tag{2.7}$$

Using the Hamiltonian equations, the time evolution of the system can be determined by finding its state at time t, characterised by a specific set of position and momenta $\{\mathbf{r}_1(t), \mathbf{r}_2(t), ..., \mathbf{r}_N(t), \mathbf{p}_1(t), \mathbf{p}_2(t), ..., \mathbf{p}_N(t)\}$ for a system with N atoms. This gives rise to the notion of phase space, which encompasses all possible states of the system. In this space, each state is fully characterised by the positions and momenta (each contributing D coordinates) of the N atoms, yielding a total of $2 \times D \times N$ dimensions. Even for relatively small systems, the high dimensionality of phase space is challenging to conceptualise, underscoring the necessity for a statistical treatment of molecular systems.

2.1.2 Ensembles

In statistical mechanics, a *statistical ensemble* is defined as the collection of all possible microstates of a system that share specified macroscopic constraints, such as a fixed number of atoms N, temperature T, pressure P or volume V. Each microstate in the ensemble corresponds to a unique assignment of coordinates and momenta and may exhibit distinct values of various properties. To capture the overall behaviour of the system, *ensemble averages* are introduced, which express a given property b as an average over all microstates. If p_i is the probability of occupying microstate i, and b_i is the value of b in that microstate, the ensemble average $\langle b \rangle$ is given by

$$\langle b \rangle \;=\; \sum_{i=1}^{N} p_i \, b_i, \tag{2.8}$$

where N represents the number of microstates. The ensemble average is related to quantities measured in simulations through the *ergodic principle*. According to the ergodic hypothesis, if a system is observed sufficiently long, each microstate will be visited in proportion to its probability p_i. Consequently, the time average of a property b over a long enough simulation converges to the ensemble average:

$$\overline{b} \;=\; \lim_{M \to \infty} \frac{1}{M} \sum_{i=1}^{M} b_i \;\approx\; \langle b \rangle, \tag{2.9}$$

where M denotes the number of time steps or samples in the simulation trajectory.

Although the ergodic principle provides a way to estimate ensemble averages, explicitly knowing the probabilities p_i remains essential for determining other macroscopic properties that cannot be measured directly. One route to deriving an expression for p_i is to consider how entropy (or information) depends on the probability distribution $\mathbf{P} = \{P_1, \ldots, P_N\}$. Such a function must satisfy four axioms:

1. It should be a continuous function depending only on $\mathbf{P}$.
2. It should be maximised by a uniform distribution.
3. It should be additive when combining independent ensembles.
4. It should remain unchanged by adding inaccessible states ($P_i = 0$).

The unique function that meets these requirements is the *Gibbs entropy*:

$$S(\mathbf{P}) \;=\; -\,k_B \sum_{i=1}^{N} P_i \,\ln\!\big(P_i\big), \tag{2.10}$$

where the factor k_B is a constant, which will be defined later. This formula is fundamental in statistical mechanics as it describes the entropy of a generalised ensemble. However, it is also widely adopted in other fields, such as communication and information theory, where Shannon reached an equivalent expression. Another essential requirement is the normalisation of the probabilities:

$$\sum_{i=1}^{N} P_i \;=\; 1. \tag{2.11}$$

With equations 2.8, 2.10 and 2.11 in place, one can derive the *general partition function*.

2.1.3 General Partition Function and Change in Entropy

The *partition function* is one of the central quantities in statistical mechanics. It encapsulates the statistical properties of a system in equilibrium and enables the calculation of thermodynamic quantities such as free energy, entropy, and average energy. The precise form of the partition function depends on the ensemble of interest. In pursuit of a general description, two guiding rules are adopted for constructing a “general” ensemble. The first rule states that some extensive quantities, denoted by $\alpha_i^{(k)}$, are *fixed*. Only microstates with the specified value $\alpha^{(k)}$ are allowed, where k indexes the constrained quantity. In discrete form, this restriction can be written using the discrete delta function, which is analogous to the Dirac delta for discrete

functions:

$$\delta(\alpha_i^{(k)} - \alpha^{(k)}) > 0 \tag{2.12}$$

The second rule states that the average values of extensive variables, denoted by $b^{(l)}$, where l indexes the variable, remain finite. This can be expressed with the ensemble average relation given in equation 2.8.

To determine p_i, the *principle of maximum entropy* is applied, which states that the most likely probability distribution is the one maximising the Gibbs entropy (equation 2.10), subject to the normalisation of probabilities (equation 2.11) and finite ensemble averages (equation 2.8). This leads to the following constrained optimisation problem:

$$\begin{aligned} \max_{\mathbf{P}} \quad & S(\mathbf{p}) = -\,k_B \sum_i P_i \ln(P_i) \\ \text{s.t.} \quad & -\,k_B \left(\sum_i P_i \;-\; 1 \right) = 0 \\ & -\,k_B \left(\sum_i P_i b_i^{(l)} \;-\; \langle b^{(l)} \rangle \right) = 0 \end{aligned} \tag{2.13}$$

where the factor $-k_B$ has been added in the second and third lines for convenience, and the summation over i is simplified for better readability. To solve this maximisation, the method of Lagrange multipliers is applied, giving rise to the Lagrange function:

$$\begin{aligned} \mathcal{L}\big(\mathbf{P}, \lambda_0, \{\lambda^{(l)}\}\big) = & -\,k_B \sum_i P_i \ln(P_i) - k_B \lambda_0 \left(\sum_i P_i - 1 \right) \\ & - k_B \sum_l \lambda^{(l)} \left(\sum_i P_i b_i^{(l)} - \langle b^{(l)} \rangle \right), \end{aligned} \tag{2.14}$$

where λ_0 and $\lambda^{(l)}$ are Lagrange multipliers, and l indexes the multiplier and the quantity $b^{(l)}$. Taking the partial derivative of $\mathcal{L}$ with respect to P_i and setting it to zero yields:

$$\frac{\partial \mathcal{L}}{\partial P_i} = -k_B \ln P_i - k_B - k_B\lambda_0 - k_B \sum_l \lambda^{(l)} b_i^{(l)} = 0 \,, \tag{2.15}$$

which can be simplified by defining $\Psi = \lambda_0 + 1$, and rearranged to isolate p_i:

$$P_i = \frac{e^{-\sum_l \lambda^{(l)} b_i^{(l)}}}{e^{\Psi}}, \tag{2.16}$$

Summing both sides over all microstates i and recalling that $\sum_i P_i = 1$ leads to:

$$Z = e^{\Psi} = \sum_i^N e^{-\sum_l \lambda^{(l)} b_i^{(l)}}, \tag{2.17}$$

where Z is the *generalised partition function*, which provides a summation of all possible microstates in a given ensemble.

Combining equations 2.16 and 2.17 results in the general probability distribution, which holds for a wide range of ensembles:

$$P_i = \frac{e^{-\sum_l \lambda^{(l)} b_i^{(l)}}}{Z} = \frac{e^{-\sum_l \lambda^{(l)} b_i^{(l)}}}{\sum_{i=1}^N e^{-\sum_l \lambda^{(l)} b_i^{(l)}}} \tag{2.18}$$

Having determined the probability distribution P_i, the Gibbs entropy can be expressed in terms of Z and the ensemble averages. Inserting equation 2.18 into equation 2.10, and making use of the normalisation of probabilities (equation 2.11) and ensemble averages (equation 2.8), following relation is found:

$$\frac{S}{k_B} = \sum_i^N P_i \Big(\ln(Z) + \sum_l \lambda^{(l)} b_i^{(l)} \Big) = \ln(Z) + \sum_l \lambda^{(l)} \langle b^{(l)} \rangle \,. \tag{2.19}$$

Taking the total differential of S yields:

$$\frac{dS}{k_B} = d\ln(Z) + \sum_l \lambda^{(l)} d\langle b^{(l)} \rangle + \sum_l \langle b^{(l)} \rangle d\lambda^{(l)} \,. \tag{2.20}$$

In the pursuit of an expression for $d\ln(Z)$, $\ln(Z)$ is differentiated w.r.t.

$\alpha^{(k)}$:

$$\frac{\partial \ln(Z)}{\partial \alpha^{(k)}} = \frac{\partial \left(\ln \sum_i e^{-\sum_l \lambda^{(l)} b_i^{(l)}} \right)}{\partial \alpha^{(k)}} \tag{2.21}$$

Using the chain rule and the product rule as well as equation 2.18, this expression can be rewritten as:

$$d \ln(Z) = -\sum_l \langle b^{(l)} \rangle d\lambda^{(l)} - \sum_l \lambda^{(l)} \left\langle \frac{\partial b^{(l)}}{\partial \alpha^{(k)}} \right\rangle d\alpha^{(k)} \tag{2.22}$$

and when combining equations 2.20 and 2.22 one finds:

$$\frac{dS}{k_B} = -\sum_l \lambda^{(l)} \left\langle \frac{\partial b^{(l)}}{\partial \alpha^{(k)}} \right\rangle d\alpha^{(k)} + \sum_l \lambda^{(l)} d\langle b_i^{(l)} \rangle \ . \tag{2.23}$$

This result shows explicitly how variations in the “fixed” extensive variables $\alpha^{(k)}$ and in “finite” extensive variables $b^{(l)}$ affect the entropy. With the concepts introduced so far, it is possible to describe specific ensembles by choosing the appropriate set of variables and constraints.

2.1.4 Microcanonical Ensemble (NVE)

Arguably, the simplest ensemble in statistical mechanics is the *microcanonical ensemble*, which describes an isolated system. Such a system has a fixed number of atoms, N', confined to a fixed volume, V', and a fixed total energy, E', commonly abbreviated as the *NVE* ensemble. Since the total energy is fixed, all microstates in this ensemble must have the same energy e_i, which is calculated by the system’s Hamiltonian (equation 2.5):

$$e_i = \mathcal{H}(\mathbf{r}_i, \mathbf{p}_i) \tag{2.24}$$

The fixed variables N', V' and E' relate to the fixed variable $\alpha^{(k)}$ and are enforced by equation 2.12:

$$\delta(e_i(\mathbf{r}_i, \mathbf{p}_i) - E') > 0 \quad \Leftrightarrow \quad \delta(\mathcal{H}(\mathbf{r}_i, \mathbf{p}_i) - E') > 0 \ , \tag{2.25}$$

$$\delta(n_i - N') > 0 \,, \tag{2.26}$$

$$\delta(v_i(\mathbf{r}_i, \mathbf{p}_i) - V') > 0 \,, \tag{2.27}$$

where n_i is the number of atoms in microstate i and v_i is the volume of the microstate. The partition function for the microcanonical ensemble, Ω, is defined as the sum over all microstates that satisfy the constraints from equations 2.25, 2.26 and 2.27:

$$\Omega = \sum_i \delta\big(n_i - N'\big)\delta\big(v_i(\mathbf{r}_i, \mathbf{p}_i) - V'\big)\delta\big(\mathcal{H}(\mathbf{r}_i, \mathbf{p}_i) - E'\big) \,. \tag{2.28}$$

Using equation 2.18, the probability of being in some microstate i can be found:

$$P_i = \frac{1}{\Omega} \tag{2.29}$$

This result reflects the *principle of equal a priori probabilities*, which states that every microstate in an isolated system is equally likely to occur. Moreover, when equation 2.29 is combined with the Gibbs entropy (equation 2.10), Boltzmann's famous equation, $S = k_B \ln \Omega$, is found.

2.1.5 Canonical Ensemble (NVT)

The microcanonical ensemble is an idealised system that is isolated from its surroundings. However, most physical systems exchange energy with their surroundings, such that other ensembles are required to describe them. A commonly used ensemble to imitate experimental conditions is the canonical ensemble. It represents a closed system with a fixed number of atoms, N', a fixed volume, V', submerged in a heat bath at a constant temperature, T, so that the temperature of the system is maintained at T. The fixed extensive variables N' and V' relate to the fixed variable $\alpha^{(k)}$ and are described using equation 2.12:

$$\delta(n_i - N') > 0 \tag{2.30}$$

$$\delta(v_i - V') > 0 \tag{2.31}$$

Additionally, the ensemble average of the energy, E, is required to be finite, thus relating to the property b introduced in subsection 2.1.3. This condition can be described as:

$$E = \langle e_i \rangle = \sum_i \mathcal{H}(\mathbf{r}_i, \mathbf{p}_i) P_i \ . \tag{2.32}$$

In section 2.1.3, a general expression for the change in entropy was derived (equation 2.23). This expression can be altered by enforcing the constraints of the canonical ensemble (equations 2.30, 2.31 and 2.32), resulting in a more tangible formulation for the change in entropy:

$$\frac{dS}{k_B} = -\lambda \left\langle \frac{\partial \mathcal{H}(\mathbf{r}_i, \mathbf{p}_i)}{\partial N} \right\rangle dN - \lambda \left\langle \frac{\partial \mathcal{H}(\mathbf{r}_i, \mathbf{p}_i)}{\partial V} \right\rangle dV + \lambda d\langle e_i \rangle \ . \tag{2.33}$$

Comparing this equation with the change in entropy as defined in classical thermodynamics (which is derived from the first and second law of thermodynamics) [48], a direct correspondence of terms is exposed:

$$dS = -\frac{\mu}{T} dN + \frac{P}{T} dV + \frac{1}{T} dE \ , \tag{2.34}$$

Here, μ is the chemical potential and P the pressure (different from the probability distribution $\mathbf{P}$, that is written **bold**). One can equate the three terms of each of the equations 2.33 and 2.34 as follows:

$$\mu = \left\langle \frac{\partial \mathcal{H}(\mathbf{r}_i, \mathbf{p}_i)}{\partial N} \right\rangle \quad ; \quad p = -\left\langle \frac{\partial \mathcal{H}(\mathbf{r}_i, \mathbf{p}_i)}{\partial V} \right\rangle \quad ; \quad \langle e_i \rangle = E \tag{2.35}$$

Furthermore, an expression for λ is found:

$$\lambda = \frac{1}{k_B T} \tag{2.36}$$

It is conventional in statistical mechanics to replace the λ with a β, such that:

$$\beta = \frac{1}{k_B T} \tag{2.37}$$

The probability of finding state i, given in equation 2.18, will then have the following form in the canonical ensemble:

$$p_i = \frac{e^{-\beta \mathcal{H}(\mathbf{r}_i, \mathbf{p}_i)}}{Z_{NVT}} \tag{2.38}$$

and Z_{NVT} is the canonical partition function. Since it will be used repeatedly in the following sections, Z_{NVT} will, from now on, be written as Q to distinguish it from other ensembles. By applying the property $\sum_i P_i = 1$ on both sides of equation 2.38, Q can be expressed as:

$$Q = \sum_i e^{-\beta \mathcal{H}(\mathbf{r}_i, \mathbf{p}_i)} \tag{2.39}$$

Finally, using the expression of the entropy in equation 2.19, as well as the expression for λ and $\langle b^{(l)} \rangle$ shown above, the relation emerges:

$$k_B T \ln(Q) = -(E - TS) \ , \tag{2.40}$$

where the expression provides a link to the thermodynamic definition of the Helmholtz free energy F:

$$F = E - TS \tag{2.41}$$

When combining equations 2.40 and 2.41, one finds a relation between the Helmholtz free energy F and the canonical partition function Q:

$$F = -k_B T \ln(Q) \tag{2.42}$$

This equation is of central importance: it shows how to compute the free energy of a system directly from its canonical partition function Q. The Helmholtz free energy F measures the portion of a system's energy that is available to do work at constant T and V. Consequently, a system with

high F can spontaneously evolve to a state of lower F, whereas the reverse transition is not thermodynamically favoured.

Note to the reader: As the Helmholtz free energy is predominantly used in this work, it will be referred to as the free energy.

2.1.6 Continuous Phase Space

Up to this point, it was assumed that ensembles are *discrete* collections of microstates. In reality, however, positions and momenta of atoms span a *continuous* phase space, which should be reflected in the equations. The underlying principles introduced so far are still valid, but the discrete sums must be transformed into integrals. As such, the partition function of the canonical ensemble for a system with N atoms in D-dimensional space can be written as:

$$Q = \frac{1}{h^{DN}N!}\int e^{-\beta\mathcal{H}(\mathbf{r}^N,\mathbf{p}^N)}d\mathbf{p}^N d\mathbf{r}^N \tag{2.43}$$

where the bold $\mathbf{r}^N$ and $\mathbf{p}^N$ denote vectors of position and momenta. Since the partition function should be a dimensionless quantity, it is divided by Planck's constant h to the power of $D \times N$ (the number of dimensions times the number of atoms). Furthermore, the expression is also divided by $N!$ to avoid over-counting microstates that differ only by permutations of identical atoms.

Equation 2.43 can be further simplified by splitting the Hamiltonian (described in equation 2.5) into two terms representing the potential energy $U(\mathbf{r}^N)$ and the kinetic energy $K(\mathbf{p}^N)$. Since $K(\mathbf{p}^N)$ only depends on momenta and $U(\mathbf{r}^N)$ only on positions, the kinetic term can be integrated out:

$$Q = \frac{1}{\Lambda^{DN}N!}\int e^{-\beta U(\mathbf{r}^N)}d\mathbf{r}^N \tag{2.44}$$

where Λ is the Thermal de Broglie wavelength, defined as:

$$\Lambda = \sqrt{\frac{\beta h^2}{2\pi m}} \tag{2.45}$$

The remaining integral of the equation 2.44 is commonly referred to as the configurational integral Z_N:

$$Z_N = \int e^{-\beta U(\mathbf{r}^N)} d\mathbf{r}^N \tag{2.46}$$

Consequently, the probability of finding the system in configuration $\mathbf{r}^N$ is:

$$P(\mathbf{r}^N) = \frac{e^{-\beta U(\mathbf{r}^N)}}{Z_N} \tag{2.47}$$

Hence, the ensemble average of a property A that depends only on positions becomes:

$$A = \left\langle a(\mathbf{r}^N) \right\rangle = \int P(\mathbf{r}^N) a(\mathbf{r}^N) d\mathbf{r}^N = \frac{\int a(\mathbf{r}^N) e^{-\beta U(\mathbf{r}^N)} d\mathbf{r}^N}{Z_N} \tag{2.48}$$

Using the concepts introduced in this section, the average properties of the data produced by molecular dynamics simulations can be found. The following section presents the core principles of molecular dynamics simulations.

2.2 Classical Molecular Dynamics

Molecular Dynamics (MD) simulations have become increasingly effective and user-friendly due to the availability of efficient methods, robust software, and accurate parameters. Nonetheless, careful setup and thoughtful parameter selection are essential to ensure reliable and physically meaningful simulation results. For this reason, it is necessary to have an insight into the fundamental principles of molecular dynamics.

2.2.1 Initialisation

To start any simulation, one must provide initial conditions for the system, namely the positions and momenta of all N atoms:

$$\{\mathbf{r}_1(0), \mathbf{r}_2(0), \ldots, \mathbf{r}_N(0), \mathbf{p}_1(0), \mathbf{p}_2(0), \ldots, \mathbf{p}_N(0)\}.$$

Because equilibrium properties are independent of the initial positions, these can be chosen arbitrarily, provided they do not lead to non-physical configurations, such as overlapping nuclei. Common approaches include positioning atoms on a regular lattice or generating random positions subject to some physical constraints. The initial velocities are typically assigned by sampling from a Boltzmann distribution at the desired initial temperature [48], ensuring that the system's kinetic energy is consistent with thermal equilibrium.

2.2.2 Simulation Box

In addition to the initial conditions, a finite simulation box must be defined, which confines the system and prevents atoms from escaping. To mimic an infinite system and eliminate surface effects, periodic boundary conditions (PBC) are implemented. Under PBC, when an atom exits one side of the simulation box, it re-enters from the opposite side with unchanged momentum, ensuring continuity of motion and preserving the bulk properties of the system. Figure 2.1 schematically illustrates periodic boundary conditions in two dimensions, where identical replicas surround the central simulation box.

Figure 2.1 shows that the central simulation box is surrounded by periodic images, which collectively represent an infinite lattice. This arrangement ensures that atoms do not experience artificial boundaries or wall effects, thereby allowing the simulation to capture the bulk properties of the system.

2.2.3 Calculation of Forces

Once the system is initialised, the system's dynamics can be simulated by determining the forces that drive the atoms. The total force acting on each atom is the sum of contributions from intramolecular interactions (forces

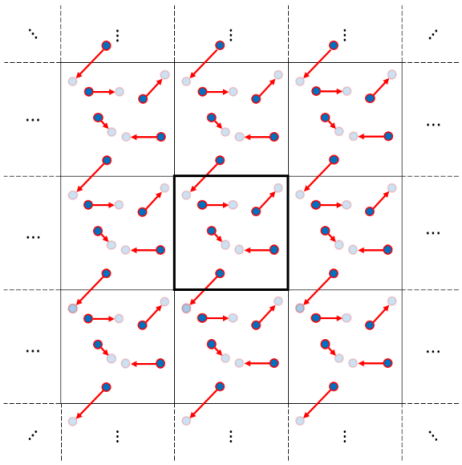


Figure 2.1: Visualisation of periodic boundary conditions in 2 dimensions: The simulated box is in the centre, surrounded by copies of it. The dark-blue dots represent the initial atoms' positions, the light-blue dots represent the evolved atoms' positions, and the red line represents the path of the atoms.

among atoms within the same molecule), intermolecular interactions (forces between atoms in different molecules), and any external forces applied to the system:

$$\mathbf{F}_i(\mathbf{r},t) = \mathbf{F}_i^{\text{intra}}(\mathbf{r},t) + \mathbf{F}_i^{\text{inter}}(\mathbf{r},t) + \mathbf{F}_i^{\text{ext}}(\mathbf{r},t). \tag{2.49}$$

Intramolecular forces arise from interactions between atoms that are covalently bonded within the same molecule. These forces account for the molecule's internal degrees of freedom, which include bond stretching, angle bending, and torsional (dihedral) rotations. As molecules undergo vibrations and conformational changes, the corresponding intramolecular forces also vary.

Intermolecular forces result primarily from Van der Waals and electrostatic interactions between atoms of different molecules. These interactions are typically modelled using distance-dependent functions. A Lennard-Jones or Buckingham potential often represents the Van der Waals forces, while Coulomb's law describes the electrostatic interactions. In many cases, additional effects, such as hydrogen bonding or induced polarisation, may

also be included depending on the complexity of the force field.

The complete set of potential energy functions that describe all the intramolecular and intermolecular interactions in a system is collectively known as a *force field*. In molecular dynamics simulations, an appropriate force field is selected from available databases, each optimised for different types of systems and interactions. In Chapters 3 and 5, the *AMBER* (Assisted Model Building with Energy Refinement) force field [66, 67, 68, 69] is employed to simulate alanine dipeptide, because it has been extensively validated for accurately reproducing the behaviour of proteins [68]. In contrast, Chapter 6 focuses on free radical polymerisation, particularly the β-scission reaction in butyl acrylate dimer, and investigates the effects of various solvents. For this purpose, the Generalised AMBER Force Field (GAFF) [68, 69] is used because it is designed to handle a broader range of organic molecules and solvents. Choosing an appropriate force field is essential for any simulation, and once this is done, the simulation environment can be set up.

2.2.4 Cutoff Schemes and Neighbour Lists

In molecular dynamics simulations, evaluating intramolecular interactions can be computationally demanding if every pair of atoms is considered. To make these calculations tractable, a cutoff distance is introduced so that interactions are only computed for atom pairs separated by a distance less than the cutoff. This approximation significantly reduces the number of force evaluations while maintaining accuracy.

A crucial aspect of employing cutoffs efficiently is the use of neighbour lists, which keep track of all atoms within the cutoff distance for each atom. Instead of recalculating the distances for every atom pair at each time step, the simulation periodically updates these lists, thereby accelerating the computation of intramolecular interactions.

In this work, the simulation utilises the Verlet scheme, which offers an efficient balance between accuracy and performance. By maintaining a

buffer zone around the cutoff distance, the Verlet scheme minimises the frequency of neighbour list updates while ensuring no interacting pairs are missed as the system evolves. This approach has proven effective in many MD simulations by providing a robust method to handle the computational demands of non-bonded force calculations.

2.2.5 Integration of Equations of Motion

Except for highly simplified cases, it is not possible to solve the Equations of Motion analytically, which is why an iterative numerical approach is used. Time is discretised into small increments, Δt, and the evolution of the system is determined iteratively using an integration scheme that accurately captures the dynamics of the system. A widely adopted method is the *Velocity Verlet* algorithm [70], which updates both positions and velocities in a coupled manner:

$$\mathbf{r}(t+\Delta t) = \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \tfrac{1}{2}\mathbf{a}(t)\Delta t^2, \tag{2.50}$$

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t) + \tfrac{1}{2}\left[\mathbf{a}(t) + \mathbf{a}(t+\Delta t)\right]\Delta t, \tag{2.51}$$

where the acceleration at time t is computed as $\mathbf{a}(t) = \mathbf{F}(t)/m$, and $\mathbf{a}(t+\Delta t)$ is evaluated after updating the positions.

Another widespread algorithm is the *Leap Frog Verlet* scheme [70], which is the default integrator in popular MD software such as GROMACS [71]. In contrast to the *Velocity Verlet* algorithm, the velocities are evaluated with a half-step offset from the positions:

$$\mathbf{r}(t+\Delta t) = \mathbf{r}(t) + \mathbf{v}\left(t + \frac{\Delta t}{2}\right)\Delta t, \tag{2.52}$$

$$\mathbf{v}\left(t + \frac{\Delta t}{2}\right) = \mathbf{v}\left(t - \frac{\Delta t}{2}\right) + \mathbf{a}(t)\Delta t, \tag{2.53}$$

While the advantage of the *Leap Frog Verlet* is a lower memory overhead, *Velocity Verlet* provides synchronised positions and velocities, simplifying

the calculation of kinetic energies and temperature control. Nevertheless, both algorithms are time reversible and symplectic in nature (preservation of energy and phase space volume over long simulation times). These properties are essential for accurately reproducing the dynamical behaviour of molecular systems, rendering both approaches reliable integration algorithms.

An essential aspect of the integration process is selecting an appropriate time step. A time step that is too large can result in unreliable trajectories, while a time step that is too small increases computational cost without substantial gains in accuracy. Therefore, careful calibration is critical to balancing computational efficiency with the desired level of precision.

The trajectory of the molecular system is constructed over time by iteratively updating positions and velocities using an appropriate integration scheme. However, additional mechanisms are required to maintain the desired simulation conditions.

2.2.6 Thermostats and Barostats

Controlling simulation conditions, such as temperature and pressure, is essential to ensure the system behaves as intended during an MD simulation.

Temperature control is achieved through thermostats, which periodically rescale or modify the velocities of atoms to maintain a desired temperature. This mimics the behaviour of a system in thermal contact with an infinite heat bath at a constant temperature.

Similarly, pressure control involves barostats that regulate the pressure by altering the volume or shape of the simulation box in response to deviations from the target pressure. This mimics the effect of coupling the system to a pressure reservoir, comparable to being connected to a constant-pressure piston.

The choice of thermostats and barostats depends on the specific simulation conditions and objectives, and the choice is stated in the simulation details.

By incorporating effective thermostats and barostats, the simulation maintains stable temperature and pressure, thereby ensuring that the computed properties reflect realistic conditions throughout the simulation. However, when the system is first initialised, it is typically not in equilibrium, requiring an equilibration process to ensure that the outputs represent the steady-state behaviour.

2.2.7 Equilibration Run

After the system is initialised, it is typically still far from equilibrium. The initial configuration may contain non-physical artefacts, necessitating an equilibration process to allow the system to adjust to the imposed conditions.

In this process, the system often first undergoes an energy minimisation step, in which high-energy contacts are resolved, and the potential energy is reduced. Next, a thermal equilibration gradually adjusts the velocities of the atoms to bring the system to the target temperature, followed by pressure equilibration if maintaining a specific pressure is required. Note that the protocol can vary depending on the simulation conditions. Throughout this process, key properties such as total energy, temperature, and pressure are monitored, indicating when the system has reached an equilibrated state.

After equilibration, the simulation enters the production phase. In this phase, the system's configuration and other properties are recorded at regular intervals for subsequent analysis.

2.2.8 Output

The primary output of a molecular dynamics simulation is the trajectory, i.e., the time evolution of the system's coordinates. Although these raw data can be visualised to observe the system's evolution, an individual snapshot is highly sensitive to initial conditions. Instead, the trajectory is used to compute averages of properties such as pressure, free energy or distance between atoms. By sampling the system repeatedly over time, these averages converge to values that accurately represent the system's behaviour under

the imposed conditions.

The following section describes two pioneering methods for evaluating the free energy difference between two states from simulation data. It is followed by a section that presents modern techniques for enhancing the sampling of rare events.

2.3 Free Energy Calculations

Even though equation 2.42 that is used to calculate the free energy seems straightforward, it requires evaluating the partition function, which is only feasible for elementary systems (e.g., a single atom in a box). Nevertheless, in most applications, researchers are more interested in the free energy *difference* between two states, since this quantity reveals which state is thermodynamically more favourable.

2.3.1 Free Energy Perturbation Theory

Free energy perturbation theory (FEP) [8, 48] was one of the earliest methods for determining the free energy difference between two states, A and B. For example, in a molecular adsorption process, state A might describe a freely moving molecule in solution, whereas state B corresponds to the same molecule adsorbed on a surface. The potential energy of state A is $U_A(\mathbf{r}^N)$ (no surface interactions), while in state B it is $U_B(\mathbf{r}^N)$ (including the adsorption interactions).

Using equations 2.42 and 2.44, the free energy difference between the two states is:

$$\Delta F_{AB} = F_B - F_A = -k_B T \ln\left(\frac{Q_B}{Q_A}\right) = -k_B T \ln\left(\frac{\int e^{-\beta U_B(\mathbf{r}^N)} d\mathbf{r}^N}{\int e^{-\beta U_A(\mathbf{r}^N)} d\mathbf{r}^N}\right) \tag{2.54}$$

To recast the ratio of partition functions as a canonical ensemble average over state A, the integrand of the numerator is multiplied by $1 = \exp[\beta U_A(\mathbf{r}^N) - \beta U_A(\mathbf{r}^N)]$, yielding:

$$\Delta F_{AB} = -k_B T \ln \left(\frac{\int e^{-\beta\left(U_B(\mathbf{r}^N) - U_A(\mathbf{r}^N)\right)} e^{-\beta U_A(\mathbf{r}^N)} d\mathbf{r}^N}{\int e^{-\beta U_A(\mathbf{r}^N)} d\mathbf{r}^N} \right). \tag{2.55}$$

Recognising the canonical ensemble average from equation 2.48, it can be simplified to:

$$\Delta F_{AB} = -k_B T \ln \left\langle e^{-\beta\left(U_B(\mathbf{r}^N) - U_A(\mathbf{r}^N)\right)} \right\rangle_A, \tag{2.56}$$

which is commonly referred to as the FEP equation [8]. In practice, one samples a set of configurations $\{\mathbf{r}^N\}$ from the canonical distribution of state A, and re-evaluates their energies employing the potential U_B. By averaging $e^{-\beta\left(U_B(\mathbf{r}^N) - U_A(\mathbf{r}^N)\right)}$ over these sampled configurations, the free energy difference ΔF_{AB} can be estimated.

However, if the overlap between the configurational spaces of A and B is not significant, the free energy difference may have large uncertainties. A possible solution to this problem is to introduce a set of $M-2$ intermediate states with potentials $U_\alpha(\mathbf{r}^N)$, where $\alpha = \{1, ..., M\}$, $\alpha = 1$ corresponds to state A and $\alpha = M$ to state B. The *free energy perturbation* equation (2.56) can be expanded for this approach:

$$\Delta F_{AB} = -k_B T \sum_{\alpha=1}^{M-1} \ln \left\langle e^{-\beta\left(U_{\alpha+1} - U_\alpha\right)} \right\rangle_\alpha. \tag{2.57}$$

By ensuring that consecutive intermediate states α and α+1 have sufficiently overlapping configurations, the free energy change is evaluated in smaller, more tractable steps, thereby improving reliability.

2.3.2 Thermodynamic Integration

Thermodynamic Integration (TI) [7, 48] is similar to FEP, in that it also involves two potential energy functions, $U_A(\mathbf{r}^N)$ and $U_B(\mathbf{r}^N)$, corresponding to

the initial state A and the final state B. However, TI introduces an *external* switching parameter, $\lambda \in [0, 1]$, which defines a *metapotential*:

$$U(\mathbf{r}^N, \lambda) = f(\lambda)U_A(\mathbf{r}^N) + g(\lambda)U_B(\mathbf{r}^N) \,, \tag{2.58}$$

where $f(\lambda)$ and $g(\lambda)$ are chosen so that $U(\mathbf{r}^N, \lambda = 0) = U_A(\mathbf{r}^N)$ and $U(\mathbf{r}^N, \lambda = 1) = U_B(\mathbf{r}^N)$. Ideally, f and g are defined so the system transitions smoothly from U_A to U_B as λ goes from 0 to 1.

Once the metapotential is defined, simulations are performed at several fixed values of λ and the derivative of the free energy with respect to λ is evaluated. This is accomplished by combining equations 2.42 and 2.46, and substituting $U(\mathbf{r}^N, \lambda)$ as the potential energy, one obtains:

$$\begin{aligned}
\frac{\partial F}{\partial \lambda} &= -k_B T \frac{d}{d\lambda} \ln(Q) = -\frac{k_B T}{Q} \frac{\partial Q}{\partial \lambda} = -\frac{k_B T}{Z} \frac{\partial Z}{\partial \lambda} \\
&= -\frac{k_B T}{Z} \int \Big(-\beta \frac{\partial U(\mathbf{r}^N, \lambda)}{\partial \lambda} \Big) e^{-\beta U(\mathbf{r}^N, \lambda)} d\mathbf{r}^N \\
&= \frac{\int \frac{\partial U(\mathbf{r}^N, \lambda)}{\partial \lambda} e^{-\beta U(\mathbf{r}^N, \lambda)} d\mathbf{r}^N}{Z} \\
&= \left\langle \frac{\partial U(\mathbf{r}^N, \lambda)}{\partial \lambda} \right\rangle_\lambda
\end{aligned} \tag{2.59}$$

Lastly, with the result from equation 2.59, ΔF_{AB} can be found by numerical integration:

$$\Delta F_{AB} = \int_0^1 \frac{\partial F}{\partial \lambda} d\lambda = \int_0^1 \left\langle \frac{\partial U}{\partial \lambda} \right\rangle_\lambda d\lambda \tag{2.60}$$

Thus, by incrementally transforming state A into state B, evaluating the free energy derivative at every step and integrating the derivatives along the path, ΔF_{AB} can be computed accurately.

2.3.3 Collective Variables

Another way to calculate the change in the free energy is by expressing it as a function of some *collective variable* (CV). For instance, if two molecules X

and Y react to form a product XY, it would be helpful to understand how the free energy changes as a function of the distance between the molecules, $d = |\mathbf{r}_X - \mathbf{r}_Y|$, where $\mathbf{r}_X$ and $\mathbf{r}_Y$ denote the respective molecular positions. In this example, d serves as the CV and the free energy is computed as a function of d. More generally, CVs are denoted by ξ, which is a function of the configuration $\mathbf{r}^N$, i.e., $\xi = \xi(\mathbf{r}^N)$.

To accommodate a CV in the partition function, the notation $q(\xi^{\text{ref}})$ is introduced, which denotes the partition function evaluated at a specific value ξ^{ref}. The equation for $q(\xi^{\text{ref}})$ is similar to Q (equation 2.44), however, the integrand is multiplied by a Dirac delta function $\delta\big(\xi^{\text{ref}} - \xi(\mathbf{r}^N)\big)$, ensuring that only configurations for which $\xi(\mathbf{r}^N) = \xi^{\text{ref}}$ contribute:

$$q(\xi^{ref}) = \frac{1}{h^{DN}N!}\int e^{-\beta\mathcal{H}(\mathbf{r}^N,\mathbf{p}^N)}\delta\big(\xi^{ref} - \xi(\mathbf{r}^N)\big)d\mathbf{p}^N d\mathbf{r}^N \tag{2.61}$$

The free energy associated with configurations that satisfy $\xi(\mathbf{r}^N) = \xi^{\text{ref}}$ is then defined analogously to equation 2.42:

$$F(\xi^{ref}) = -k_B T \ln q(\xi^{ref}) \tag{2.62}$$

To determine the probability of observing configurations with a particular CV value, the ratio of those configurations to all possible configurations is employed:

$$P(\xi) = \frac{q(\xi^{ref})}{Q} = \frac{\int exp\big(-\beta U(\mathbf{r}^N)\big)\delta\big(\xi^{ref} - \xi(\mathbf{r}^N)\big)d\mathbf{r}^N}{Z} \tag{2.63}$$

Combining equation 2.62 and 2.63 leads to an expression for $F(\xi^{\text{ref}})$ in

terms of $P(\xi^{\text{ref}})$:

$$\begin{aligned} F(\xi^{ref}) &= -k_B T \ln q(\xi^{ref}) \\ &= -k_B T \ln \left(P(\xi^{ref})Q\right) \\ &= -k_B T \ln(P(\xi^{ref})) - k_B T \ln Q \ , \end{aligned} \tag{2.64}$$

where the term $k_B T \ln Q$ is independent of ξ and hence acts like a constant offset. It is common practice to ignore this constant offset and use $F(\xi)$ only as a *relative* free energy:

$$F(\xi) = -k_B T \ln(P(\xi)) \tag{2.65}$$

Equation 2.65 applies to systems with a single CV. If multiple CVs $\xi_1, \ldots, \xi_M$ are introduced, one replaces the delta function in equation 2.63 with the product $\prod_{k=1}^{M} \delta\left(\xi_k^{\text{ref}} - \xi_k(\mathbf{r}^N)\right)$. This yields a corresponding expression for the free energy as a function of several CVs:

$$F(\xi_1, ..., \xi_M) = -k_B T \ln(P(\xi_1, ..., \xi_M)) \tag{2.66}$$

Although there is no strict upper limit on the number of CVs, increasing their count dramatically enlarges the CV space and leads to sampling inefficiencies, more expensive computations, and cumbersome visualisation. In this work, however, most systems are restricted to at most two CVs.

2.3.4 Ergodic Principle

Finally, to connect the molecular simulation trajectory with macroscopic properties, one employs the *ergodic principle*. The ergodic principle asserts that over sufficiently long simulation times, the fraction of time spent in a particular region of phase space is proportional to the probability of the system being in that region. In other words, the time average of any property converges to its ensemble average.

For instance, consider a collective variable $\xi(\mathbf{r}^N)$ that is recorded at

discrete time intervals throughout a simulation. If a histogram $H\big[\xi(t)\big]$ is constructed to count how often ξ takes certain values, then the probability $P(\xi)$ of observing a particular value ξ is approximated by the long-time limit of the histogram:

$$P(\xi) \;\cong\; \lim_{t\to\infty} \frac{1}{t} \int_0^t H\big[\xi(t')\big]\, dt'. \tag{2.67}$$

Although this is an approximation, it improves as the total simulation time increases, provided the simulation is ergodic in nature.

With the concepts introduced so far, a molecular simulation can be performed to sample the system's phase space and thus estimate its free energy as a function of a chosen CV.

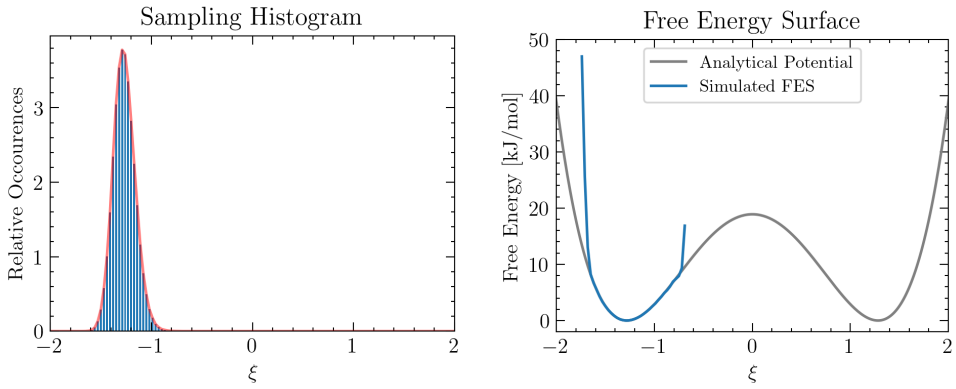
2.3.5 Langevin Dynamics Simulations

Langevin dynamics simulations (LDS) are a computationally efficient method for modelling the behaviour of systems described by a potential energy surface [72]. Unlike conventional MD, which follows deterministic Newtonian equations, LDS incorporates frictional (dissipative) forces and stochastic thermal noise to mimic the effect of a heat bath. Although LDS can be applied to complex chemical systems, in this work it is used to simulate simplified systems, characterised by a predefined analytical potential energy surface, $U(\xi)$, which depends on a CV, ξ. This approach is particularly beneficial for developing and testing various simulation methods, since simulating an analytic potential is considerably faster than simulating a chemical system. The evolution of the Langevin system is governed by the Langevin equation:

$$\mathbf{ma}(t) = -\nabla U(\xi) - \mathbf{m}\gamma\mathbf{v}(t) + \sqrt{2\mathbf{m}\gamma k_B T}\eta(t)\ , \tag{2.68}$$

where $\mathbf{m}$ is a mass matrix, γ is the friction (or damping) coefficient, and the last term presents random thermal fluctuations.

To demonstrate how $F(\xi)$ can be computed from simulation data, a one-dimensional analytic potential is employed to run an LDS. The potential is



(a) (b)

Figure 2.2: Results of a Langevin dynamics simulation of the analytical potential energy surface $U(\xi) = 7\xi^4 - 23\xi^2$. (a) Histogram of sampled ξ values (blue bars), with a continuous fit (red line). (b) Underlying potential (grey line) compared with the free energy surface calculated from simulation data (blue line)

defined as $U(\xi) = 7\xi^4 - 23\xi^2$, which mimics a system with two stable states separated by a large energy barrier (see figure 2.2). During the simulation, values of ξ are periodically recorded to construct a probability distribution, and from that, $F(\xi)$ is determined using equation 2.65. The results are depicted in Figure 2.2.

Note: The LDS are carried out with the pesmd tool of the PLUMED software. With this tool, input parameters such as temperature, time-step, the potential energy surface and its CVs are given in reduced units. In this work, it is presumed that the potential energy has units of $[kJ/mol]$*, the CV has units of* $[nm]$*, and for consistency, the time unit is* $[ps]$*.*

Throughout the simulation, ξ values between roughly -1.8 and -0.6 are sampled. Consequently, the free energy is well-defined and consistent with the analytic potential in that interval, but effectively infinite for values of ξ that have not been sampled. The reason is that the simulation was initialised in the left basin and was never able to jump to the other basin, because the energy barrier separating those two states is too high. In the next section, methods that enhance the sampling and facilitate the crossing of the energy barrier will be discussed.

2.4 Biased Sampling

With the methods introduced so far, it is possible to define a system, run an MD simulation, determine the probability density as a function of some CV, and ultimately find the free energy surface (FES). Although these steps are straightforward in principle, in practice, the simulation often revisits the same configurations, neglecting others, because they are energetically difficult to access.

In Section 2.3.5, a simplified model with two basins, separated by a large energy barrier, was considered. When the simulation was initialised in one basin, it remained trapped there, unable to cross the barrier, even after a long simulation time. This example illustrates the fundamental challenge of effectively exploring phase space in classical (brute-force) simulations.

2.4.1 Umbrella Sampling

Umbrella sampling is a method developed by Torrie and Valleau [1] and introduces a perturbation into the potential energy profile. This perturbation usually takes the form of a harmonic restraint:

$$\omega(\xi) = \frac{\kappa}{2}(\xi - \xi^{ref})^2 \ , \tag{2.69}$$

where κ is the force constant, ξ is a collective variable, and ξ^{ref} is the restraint centre. This harmonic term is often called an (attractive) bias potential, since it confines the simulation to the vicinity of ξ^{ref}, thus biasing the sampling in phase space. Due to the altered energy landscape, the equations presented previously must be modified. The biased system has the potential energy:

$$U^b = U^u + \omega(\xi) \ , \tag{2.70}$$

where the superscripts b and u denote biased and unbiased quantities, respectively. Although U depends on the configuration $\mathbf{r}^N$, that dependence has been omitted for better clarity. The probability of observing a configuration with CV value ξ^{ref} is determined using equation 2.63, however, using

the bias potential energy U^b instead of U^u results in the bias probability $P^b(\xi)$:

$$P^b(\xi) = \frac{\int e^{-\beta\left(U^u+\omega(\xi)\right)}\delta\left(\xi^{ref}-\xi\right)d\mathbf{r}^N}{\int e^{-\beta\left(U^u+\omega(\xi)\right)}d\mathbf{r}^N} \ . \tag{2.71}$$

The bias term $\omega(\xi)$ in equation 2.71 can be factored out of the integral, since it is independent of the configuration. Combining equation 2.71 with the unbiased probability from equation 2.63 yields:

$$\frac{P^u(\xi)}{P^b(\xi)} = e^{\beta\omega(\xi)}\frac{\int e^{-\beta U^u}e^{-\beta\omega(\xi)}d\mathbf{r}^N}{\int e^{-\beta U^u}d\mathbf{r}^N} \ . \tag{2.72}$$

Using the definition of the ensemble average from equation 2.8, it can be shown that:

$$P^u(\xi) = P^b(\xi)e^{\beta\omega(\xi)}\left\langle e^{-\beta\omega(\xi)}\right\rangle \tag{2.73}$$

Substituting equation 2.73 into the free energy expression from equation2.42 yields:

$$F(\xi) = -k_BT\ln\left(P^b(\xi)\right) - \omega(\xi) - k_BT\ln\left\langle e^{-\beta\omega(\xi)}\right\rangle \ , \tag{2.74}$$

where the first term, $-k_BT\ln\left(P^b(\xi)\right)$, quantifies the contribution arising from the biased probability distribution, the second term, $-\omega(\xi)$, provides the unbiasing correction to remove the influence of the bias potential, and the third term, $-k_BT\ln\left\langle e^{-\beta\omega(\xi)}\right\rangle$, corresponds to the work arising due to the use of a bias potential in an unperturbed ensemble. The expression in equation 2.74 is valid if the simulation explores all relevant regions of ξ. In practice, umbrella sampling is usually performed through multiple simulations, each with a different bias centre, to achieve thorough sampling across the CV range. This technique gives rise to a more general equation:

$$F_i(\xi) = -k_BT\ln\left(P_i^b(\xi)\right) - \omega_i(\xi) + c_i \ , \tag{2.75}$$

where the subscript i denotes the properties of simulation i and c_i is a constant that replaces the ensemble average term in the equation 2.74. This constant shifts the free energy surface vertically, so its correct value must be determined to reconstruct the entire free energy landscape. Furthermore, there must be sufficient overlap between neighbouring simulation windows to align their respective free energy segments consistently.

Several methods exist to calculate c_i, and they are all iterative in nature. A popular approach is the weighted histogram analysis method (WHAM) [57, 58], which minimises the statistical error in the total unbiased probability:

$$P^u = \sum_{i}^{N_{windows}} w_i P_i^u(\xi) \tag{2.76}$$

where w_i is the weight of each window. These weights are optimised to minimise the variance $\sigma^2\big(P^u(\xi)\big)$.

When this method is used to enhance sampling for the problem discussed in section 2.3.5, a bias would be imposed to confine the simulation around a specific value of ξ. After a short simulation under this bias, a second simulation is conducted using a bias shifted to a different ξ value. This protocol is repeated until all regions have been adequately sampled. The essential requirement of this approach is that the sampling of neighbouring regions overlap significantly, so that the data can be merged reliably to produce a continuous FES. This concept is illustrated in section 2.4.2 on umbrella integration in Figure 2.3.

2.4.2 Umbrella Integration

Umbrella integration (UI) [47, 61, 62] was introduced by Kästner and is based on the principles of umbrella sampling and thermodynamic integration. As in umbrella sampling, multiple simulations are conducted in separate windows, each constrained by a harmonic potential. The key distinction is that, rather than calculating the free energy in each window, the mean force, $\partial F/\partial \xi$, is computed. Equation 2.75, previously used to obtain the

free energy of window i, is rewritten to define the mean force:

$$\frac{\partial F_i(\xi)}{\partial \xi} = -k_B T \frac{\ln\left(P_i^b(\xi)\right)}{\partial \xi} - \frac{\omega_i(\xi)}{\partial \xi} - k_B T \frac{\ln\left\langle e^{-\beta\omega(\xi)}\right\rangle}{\partial \xi} , \tag{2.77}$$

where $\omega_i(\xi) = \frac{\kappa}{2}(\xi - \xi^{ref})^2$. The last term is an ensemble average that does not depend on ξ, so its partial derivative vanishes. Thus, the expression simplifies to:

$$\frac{\partial F_i(\xi)}{\partial \xi} = -k_B T \frac{\ln\left(P_i^b(\xi)\right)}{\partial \xi} - \kappa(\xi - \xi^{ref}) . \tag{2.78}$$

Furthermore, the biased probability distribution of window i is approximated using a Gaussian function:

$$P_i^b = \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{(\xi - \overline{\xi_i})^2}{2\sigma_i^2}} \tag{2.79}$$

where $\overline{\xi_i}$ is the mean of ξ sampled in window i, and σ_i^2 denotes the statistical variance of $\overline{\xi_i}$. The mean forces for all windows are then combined using a weighted average:

$$\frac{\partial F(\xi)}{\partial \xi} = \sum_i p_i(\xi) \, \frac{\partial F_i(\xi)}{\partial \xi} , \tag{2.80}$$

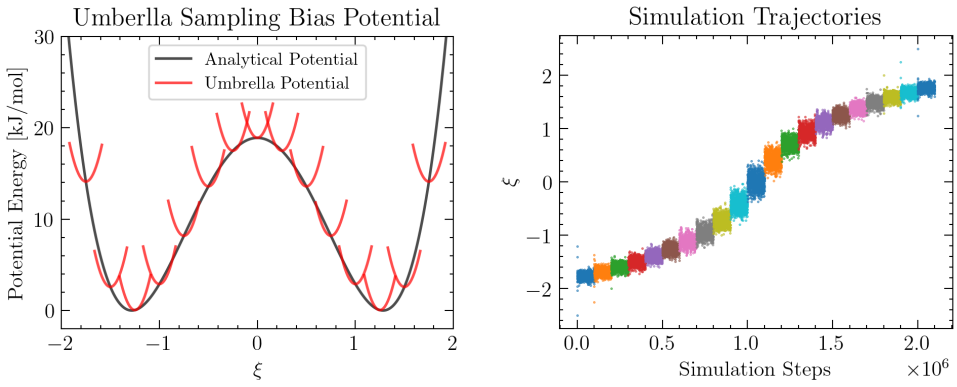
where $p_i(\xi)$ is the weighted probability distribution of window i, which is defined by:

$$p_i(\xi) = \frac{N_i P_i^b(\xi)}{\sum_i N_i P_i^b(\xi)} , \tag{2.81}$$

where N_i is the total number of steps in window i. Equations 2.81 and 2.80 ensure that windows sampled for longer durations, or regions of CV space that have been sampled more thoroughly, have a larger weight ($p_i(\xi)$) and exert a greater influence on the overall FES.

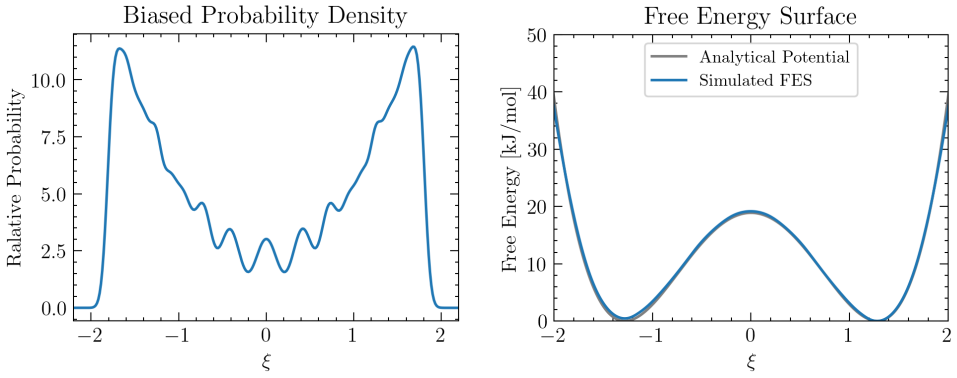
Once the total mean force is obtained via 2.80, the free energy is recovered by integrating this mean force. Such an integration is non-trivial in higher-dimensional CV spaces, and will be described in Section 3.3.

To illustrate this approach, the potential surface $U(\xi) = 7\xi^4 - 23\xi^2$ (em-



(a) (b)

Figure 2.3: (a) Harmonic constraints (red lines) are distributed along the potential energy surface (black line). (b) Simulation trajectories of the individual simulations with different colours.



(a) (b)

Figure 2.4: (a) Combined biased probability density of all simulations. (b) FES (blue line) after all simulations have been patched together with the potential energy surface (grey line).

ployed in the previous sections) can be simulated using UI. Twenty-one windows, each lasting 0.5 ns, were simulated using a harmonic restraint whose centre values were evenly spaced from $\xi = -2.5$ to $\xi = 2.5$, as illustrated in figure 2.3 (a). The resulting trajectories are plotted as a function of time and their CV in Figure 2.3 (b).

The individual mean forces are then patched together using equation 2.80, and finally integrated to obtain the FES, which is shown in figure 2.4.

The biased probability density depicted in Figure 2.4 (a) contains fluctu-

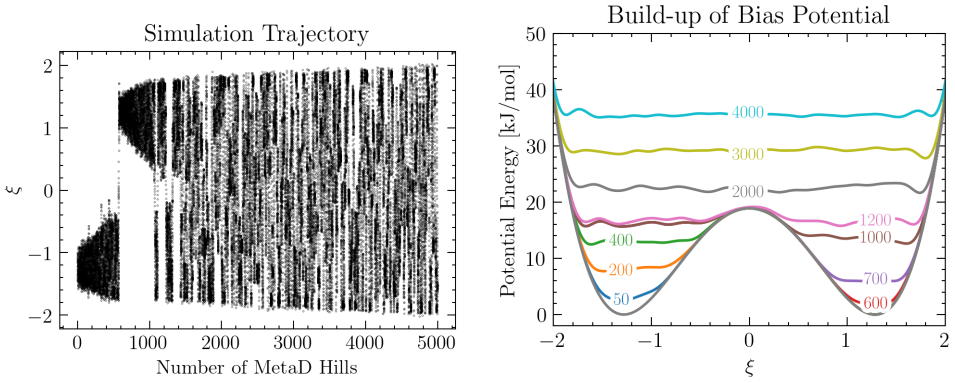
ations, which are more pronounced around $\xi = 0$, where the energy barrier is located. These fluctuations result from non-uniform sampling that arises because of the harmonic constraints. Yet, these fluctuations can not be detected in the FES in figure 2.4 (b), since they are removed by the unbiasing correction given in equation 2.78

2.4.3 Metadynamics

Metadynamics (MetaD) [2, 50] was developed by Parrinello and co-workers as an enhanced sampling method that operates in a manner analogous to umbrella sampling, except that a repulsive bias is employed rather than an attractive one. The repulsive bias is deposited at a constant rate in the region of the CV-space visited by the simulation. The individual repulsive potentials typically take the form of a Gaussian function (also referred to as *hills*), and the sum of these hills makes up the bias potential. This setup is referred to as a *history-dependent bias* because the bias experienced by the system evolves over time. Consequently, regions of CV space that are sampled frequently accumulate more repulsive bias, thereby encouraging exploration of less-visited regions. Mathematically, the MetaD bias potential, $V_t(\xi)$, at time t, for an arbitrary number of CVs ($\xi = \xi_1, ..., \xi_{N_{CV}}$), is expressed as:

$$V_t(\xi) = \sum_{t'=1}^{t} \omega \; e^{-\sum_{i=1}^{N_{CV}} \frac{\left(\xi_i - \xi_i(t')\right)^2}{2\sigma_{H,i}^2}} \quad , \tag{2.82}$$

where i indexes the CV, ξ_i is the i-th CV and $\xi_i(t)$ it the value that ξ_i takes at time t. N_{CV} is the total number of CVs, ω is the height of the Gaussian hills and $\sigma_{H,i}$ is the width of the Gaussian for the i-th CV. Because repeatedly sampled regions acquire large repulsive contributions, the simulation is directed away from those regions and eventually attains more uniform coverage of phase space. After sufficient sampling, the FES plus the bias



(a) (b)

Figure 2.5: (a) depicts the evolution of the trajectory as CV coordinate, as a function of the number of MetaD hills deposited. (b) depicts the evolution of MetaD bias potential. The lines represent potential $U(\xi)$ plus the bias potential $V_t(\xi)$, where the number of Gaussian hills deposited at time t is labelled on the line.

potential becomes nearly flat:

$$V_{t\to\infty}(\xi) \approx -F_{t\to\infty}(\xi) + C \ , \tag{2.83}$$

where C is a constant representing a vertical offset. Equation 2.83 is an approximation valid only at large simulation times. A more precise derivation is obtained by introducing a perturbation into the potential energy surface, similarly to the approach in section 2.4.1 on umbrella sampling. Taking equation 2.74, but using the history-dependent bias from metadynamics, one obtains:

$$F_t(\xi) = -k_B T \ln\left(P_t^b(\xi)\right) - V_t(\xi) - F_c \ , \tag{2.84}$$

where F_c is the ensemble average term $k_B T \ln\left\langle e^{-\beta V_t(\xi)}\right\rangle$ that accounts for the work performed in the system due to the introduction of the MetaD potential, and needs to be calculated numerically [73].

Figure2.5 illustrates the evolution of the history-dependent bias using the simplified potential surface $U(\xi) = 7\xi^4 - 23\xi^2$ (introduced in section2.3.5).

Figure 2.5 shows how the simulation starts in the left basin, where the bias potential accumulates. After about 600 hills, it escapes the left basin

and starts sampling the other basin, until about 1200 hills, when it samples most of CV space almost uniformly. Compared to the example in Section 2.3.5, the exploration of phase space was enhanced considerably.

The underlying principle of MetaD is that, even if no *a priori* information is available about the free-energy landscape, the adaptive bias increases the accessibility of higher-energy states. Consequently, new regions of CV space are explored rapidly, and the sampling of rare events improves greatly. A limitation of this method is that the bias continues to accumulate if the simulation is not terminated, which can lead to unbounded growth of the bias potential.

2.4.4 Well-Tempered Metadynamics

Even though *conventional* (non-well-tempered) MetaD is an effective tool for enhancing the sampling of CV space, after sufficient time, all states of interest might be accessible. Yet, the growing potential is impeding the convergence of the FES. Well-tempered metadynamics (WT-MetaD) [49, 74] addresses this issue by scaling the height of the Gaussian hill by a factor ω_t (as opposed to ω in metadynamics), which is defined as follows:

$$\omega_t = \omega_0 \, \tau_G \, e^{-\frac{V_t(\xi)}{\triangle T}} \, , \tag{2.85}$$

where ω_0 is the initial Gaussian height, τ_G is the rate at which the Gaussians are deposited, and $\triangle T$ is a constant that controls the rate at which the Gaussian height decreases. Because $-V_t(\xi)$ appears in the numerator of the exponential, each new Gaussian added around ξ becomes progressively smaller as more hills accumulate in that region. In the long-time limit, the bias potential converges to:

$$V_{t\to\infty}(\xi) \approx -\frac{\triangle T}{\triangle T + T} F_t(\xi) + C \tag{2.86}$$

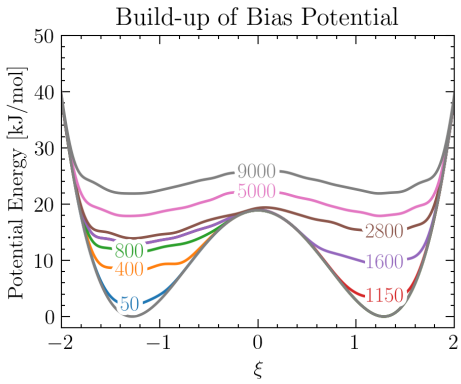


Figure 2.6: Accumulation of the bias potential. The lines represent potential $U(\xi)$ plus the bias potential $V_t(\xi)$, where the number of Gaussian hills deposited at time t is labelled on the line.

where T is the temperature of the system, and C is a constant offset. To understand the role of $\triangle T$, the extreme cases are presented:

- $\triangle T \rightarrow 0$: The Gaussian height immediately approaches zero, so the bias effectively vanishes and the simulation behaves as if no bias potential were present.
- $\triangle T \rightarrow \infty$: The Gaussian height remains constant, and the simulation behaves like the *conventional* MetaD scheme discussed in section 2.4.3.

A simulation equivalent to that in Section 2.4.3 can be performed to visualise the evolution of the bias potential. Figure 2.6 shows how the total potential ($U(\xi) + V_t(\xi)$) develops over time, with the number of deposited Gaussian hills indicated on each line.

Initially, the accumulation of the potential resembles the conventional MetaD example, but growth is curtailed in regions where the bias potential is already large. This feature is advantageous because smaller perturbations to the potential energy surface promote faster convergence of the FES.

2.4.5 Further Metadynamics Approaches

Variational Enhanced Sampling (VES) [75] is an advanced extension of the metadynamics concept, which aims to create an optimal bias potential. The

bias potential is constructed by minimising a loss function, quantifying how far the biased probability distribution deviates from a chosen target distribution, which is often uniform or well-tempered. The bias V_{VES} is expressed as a series expansion (e.g. Fourier Series) with basis functions, $\phi_k(\xi)$, indexed by k:

$$V_{\mathrm{VES}}(\xi) = \sum_k \alpha_k, \phi_k(\xi) \ , \tag{2.87}$$

where α_k are expansion coefficients that are iteratively refined during the simulation, thereby optimising V_{VES}. This approach allows the system to explore the free energy landscape efficiently while aiming to reduce the mismatch between the biased and target distributions.

On-the-Fly Probability Enhanced Sampling (OPES) [76] is another recent method that builds on metadynamics but adopts a different strategy for bias construction. With this approach, the bias potential, $V_{\mathrm{OPES}}(\xi)$, is adaptively adjusted so that the biased probability density, $P^b(\xi)$, converges towards a specified target distribution, $P^*(\xi)$. The OPES bias potential can be expressed as:

$$V_{\mathrm{OPES}}(\xi) = -\frac{1}{\beta} \ln \left(\frac{P^*(\xi)}{P^b(\xi)} \right) . \tag{2.88}$$

As with VES, the target distribution is typically chosen to resemble a well-tempered MetaD distribution or, if information about the free energy landscape is available, a custom distribution.

The advantage of VES and OPES is that these approaches offer better control of the sampling distribution compared to conventional metadynamics and it has been reported by Invernizzi *et al.* that OPES converges faster than WT-MetaD [76]. Even though these methods are effective, they lie outside the scope of this work and will not be discussed in more detail.

2.5 Mean Force Integration

Vaselina Marinova *et al.* first published the *Mean Force Integration* (MFI) method [3] in 2019. This chapter shows their initial contributions, which

serve as the foundation for the development of MFI, presented in Chapters 3 and 5.

The MFI method was inspired by Kästner's Umbrella Integration (UI) method (described in section 2.4.2), which is closely related to Umbrella Sampling (US) (described in section 2.4.1). However, rather than evaluating the free energy of each window directly, with UI, the mean force of each window is determined first. Later, the individual mean forces are patched, using a weighted average approach, and then integrated to find the free energy. This procedure removes the need to estimate the alignment constant, c_i, in equation 2.75. When Salvalaglio developed MFI, the goal was to adapt the UI approach to a history-dependent enhanced sampling scheme, such as MetaD.

2.5.1 One-Dimensional Formulation

To derive the MFI equations that allow for the calculation of the FES, equation 2.84 is differentiated with respect to ξ:

$$\frac{\partial F_t(\xi)}{\partial \xi} = -\frac{1}{\beta}\frac{\partial \ln\left(P_t^b(\xi)\right)}{\partial \xi} - \frac{\partial V_t(\xi)}{\partial \xi} - \frac{1}{\beta}\frac{\partial \ln\left\langle e^{-V_t(\xi)}\right\rangle_u}{\partial \xi}\,, \tag{2.89}$$

where term $\ln\left\langle e^{-V_t(\xi)}\right\rangle_u$ accounts for the reversible work arising from the use of the bias potential in the unperturbed ensemble. Since this quantity is constant with respect to ξ, its derivative vanishes. Otherwise, it would need to be evaluated numerically through an implicit expression [73]:

$$\left\langle e^{-V_t(\xi)}\right\rangle_u = \frac{\int e^{-\beta F(\xi)-\beta V_t(\xi)}}{\int e^{-\beta F(\xi)}}\,, \tag{2.90}$$

The MetaD bias potential $V_t(\xi)$ is a sum of Gaussians (defined in sections 2.4.4 and 2.4.4), and its derivative with respect to ξ is:

$$\frac{\partial V_t(\xi)}{\partial \xi} = -\sum_{t'=1}^{t}\frac{w_{t'}(\xi-\xi_{t'})}{\sigma_H^2}\exp\left(-\frac{(\xi-\xi_{t'})^2}{2\sigma_H^2}\right) \tag{2.91}$$

Because $V_t(\xi)$ evolves continuously during MetaD simulations, the biased probability density, $P_t^b(\xi)$, is evaluated over *time windows* of duration τ, during which the bias is constant. Instead of constructing $P_t^b(\xi)$ as a histogram with discrete bins, a summation of continuous Gaussian kernels is used:

$$P_t^b(\xi) = \frac{n_{cv}}{h\sqrt{2\pi}} \sum_{t'=t}^{t+\tau} \exp\left(- \frac{(\xi - \xi_{t'})^2}{2h^2} \right), \tag{2.92}$$

where h denotes the width of the Gaussian and n_{cv} is the sampling stride of the CV, corresponding to the Gaussian height. The placement of h and n_{cv} ensures that simulations with different values of h and n_{cv} can be combined in a consistent manner. Given that n_H is the stride at which the MetaD potential is updated, the ratio n_{cv}/n_H should be between ten and twenty for a good trade-off between computational resources and accuracy.

By differentiating equation 2.92 with respect to ξ, the first term of equation 2.89 is obtained:

$$\frac{1}{\beta} \frac{\partial \ln\left(P_t^b(\xi)\right)}{\partial \xi} = \frac{\displaystyle\sum_{t'=t}^{t+\tau} -\frac{(\xi - \xi_{t'})}{\beta h^2} \exp\left(- \frac{(\xi - \xi_{t'})^2}{2h^2} \right)}{\displaystyle\sum_{t'=t}^{t+\tau} \exp\left(- \frac{(\xi - \xi_{t'})^2}{2h^2} \right)} . \tag{2.93}$$

Combining equations 2.89, 2.91, and 2.93 enables the calculation of the mean force for the time window $[t, t+\tau]$:

$$\frac{\partial F_t(\xi)}{\partial \xi} = \frac{\displaystyle\sum_{t'=t}^{t+\tau} \frac{(\xi - \xi_{t'})}{\beta h^2} \exp\left(- \frac{(\xi - \xi_{t'})^2}{2h^2} \right)}{\displaystyle\sum_{t'=t}^{t+\tau} \exp\left(- \frac{(\xi - \xi_{t'})^2}{2h^2} \right)} + \sum_{t'=1}^{t} \frac{w_{t'}(\xi - \xi_{t'})}{\sigma_H^2} \exp\left(- \frac{(\xi - \xi_{t'})^2}{2\sigma_H^2} \right) \tag{2.94}$$

Each window's mean force is then combined (or "patched") via a weighted average, where the weight is the window's biased probability distribution:

$$\left\langle \frac{\partial F(\xi)}{\partial \xi} \right\rangle_t = \frac{\sum_{t'=0}^{t} \left(P_t^b(\xi) \frac{dF_t(\xi)}{d\xi} \right)}{\sum_{t'=0}^{t} P_t^b(\xi)} , \tag{2.95}$$

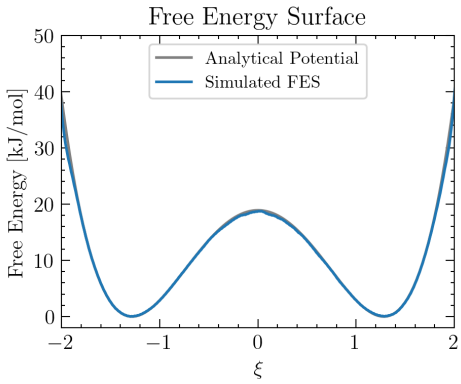


Figure 2.7: FES (blue line) constructed from a metadynamics simulation and analysed with MFI, and the analytical potential surface (grey line).

where $\left\langle \frac{\partial F(\xi)}{\partial \xi} \right\rangle_t$ is the average mean force. Lastly, the average mean force is integrated over ξ to find the free energy:

$$F_t(\xi) = \int \left\langle \frac{\partial F(\xi)}{\partial \xi} \right\rangle_t d\xi \tag{2.96}$$

The efficacy of MFI can be validated by applying the above equations to the simulation data from the example in section 2.4.4, which yields the FES shown in figure 2.7.

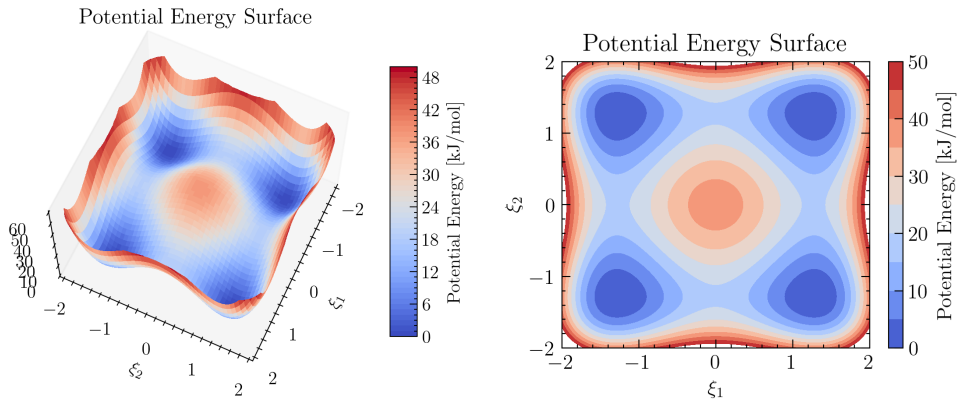
2.5.2 N-Dimensional Extension

Although the derivation above is for a one-dimensional CV, the procedure extends to N dimensions. In that case, the mean force is calculated with:

$$\nabla F_t(\boldsymbol{\xi}) = -\beta^{-1} \nabla \ln \left(P_t^b(\boldsymbol{\xi}) \right) - \nabla V_t(\boldsymbol{\xi}) \, , \tag{2.97}$$

where ∇ denotes the gradient with respect to all N collective variables, and $\boldsymbol{\xi}$ is a vector of CVs (i.e., $\boldsymbol{\xi} = \{\xi_1, ..., \xi_N\}$). The MetaD potential for an N-dimensional space is given in equation 2.82, and its gradient is defined as:

$$\nabla V_t(\boldsymbol{\xi}) = -\sum_{t'=1}^{t} w_{t'} \sigma_{H,N}^{-1} (\boldsymbol{\xi} - \boldsymbol{\xi}_{t'}) \exp \left(-\frac{1}{2} (\boldsymbol{\xi} - \boldsymbol{\xi}_{t'})^{\mathrm{T}} \sigma_{H,N}^{-1} (\boldsymbol{\xi} - \boldsymbol{\xi}_{t'}) \right) , \tag{2.98}$$



(a) (b)

Figure 2.8: (a) The analytical potential energy surface visualised in 3 dimensions and (b) the analytical potential energy surface visualised in 2 dimensions using contour lines

where $\sigma_{H,N}$ denotes a $N \times N$ covariance matrix of the Gaussian hills. The biased probability density of an N-dimensional space is:

$$P_t^b(\boldsymbol{\xi}) = \frac{n_{cv}}{\sqrt{|h_N|}(2\pi)^{N/2}} \sum_{t'=t}^{t+\tau} \exp\left(-\frac{1}{2}(\boldsymbol{\xi} - \boldsymbol{\xi}_{t'})^{\mathrm{T}} h_N^{-1} (\boldsymbol{\xi} - \boldsymbol{\xi}_{t'}) \right), \tag{2.99}$$

where h_N denotes the covariance matrix and $|h_N|$ its determinant. The gradient of the logarithm of the probability density is defined as:

$$\frac{1}{\beta}\nabla \ln\left(P_t^b(\boldsymbol{\xi})\right) = -\frac{\sum_{t'=t}^{t+\tau} \frac{h_N^{-1}}{\beta}(\boldsymbol{\xi} - \boldsymbol{\xi}_{t'}) \exp\left(-\frac{1}{2}(\boldsymbol{\xi} - \boldsymbol{\xi}_{t'})^{\mathrm{T}} h_N^{-1} (\boldsymbol{\xi} - \boldsymbol{\xi}_{t'}) \right)}{\sum_{t'=t}^{t+\tau} \exp\left(-\frac{1}{2}(\boldsymbol{\xi} - \boldsymbol{\xi}_{t'})^{\mathrm{T}} h_N^{-1} (\boldsymbol{\xi} - \boldsymbol{\xi}_{t'}) \right)}. \tag{2.100}$$

Equations 2.98 and 2.100 can be substituted into equation 2.97 to find the mean force in an N-dimensional CV space.

Although the computational cost grows exponentially with increasing dimensionality, the simulation of a two-dimensional demonstration is feasible. This is demonstrated through the analytical potential surface $U(\boldsymbol{\xi}) = 7\xi_1^4 - 23\xi_1^2 + 7\xi_2^4 - 23\xi_2^2$, illustrated in figure 2.8.

Figure 2.9 shows the calculated FES and its absolute deviation from the

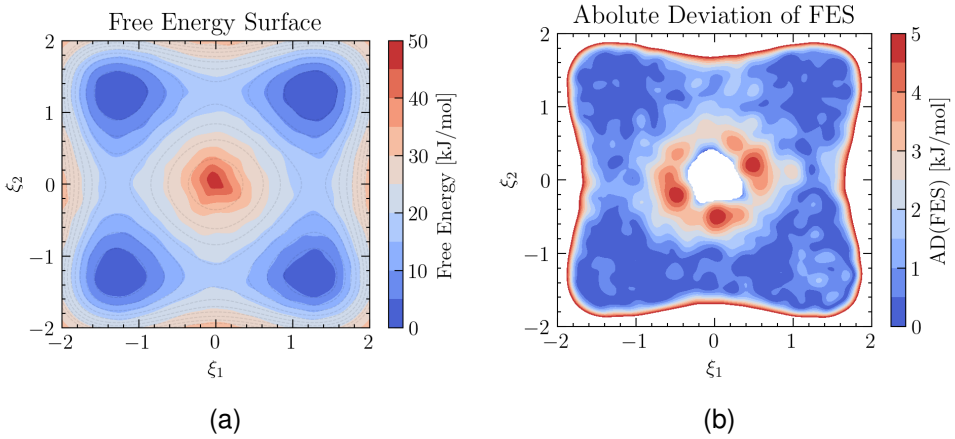


Figure 2.9: (a) The computed FES and (b) the average deviation of the FES from the analytical surface.

analytical surface. The basin regions are reproduced accurately, whereas the transition regions show larger deviations, consistent with a more limited sampling in those areas.

2.5.3 Patching Independent Simulations

A significant strength of MFI is the ability to combine independent simulations by patching their average mean forces. To accomplish this, equation 2.95 can be generalised to account for mean forces from N simulations in-

dexed by j:

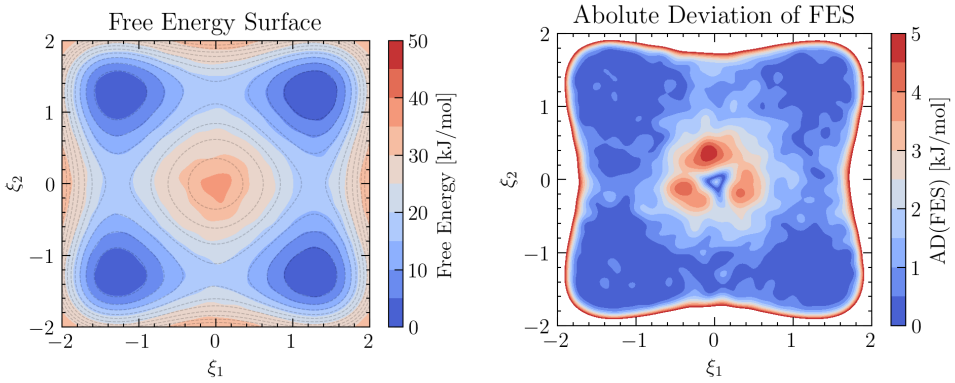
$$\begin{aligned}\left\langle \frac{\partial F(\xi)}{\partial \xi} \right\rangle_{patch} &= \frac{\sum_{t'=0}^{t_1} \left(P^b_{1,t}(\xi) \frac{\partial F_{1,t}(\xi)}{\partial \xi} \right) + \ldots + \sum_{t'=0}^{t_N} \left(P^b_{N,t}(\xi) \frac{\partial F_{N,t}(\xi)}{\partial \xi} \right)}{\sum_{t'=0}^{t_1} \left(P^b_{1,t}(\xi) \right) + \ldots + \sum_{t'=0}^{t_N} \left(P^b_{N,t}(\xi) \right)} \\ &= \frac{\sum_{j=1}^{N} \sum_{t'=0}^{t_1} \left(P^b_{j,t}(\xi) \frac{\partial F_{j,t}(\xi)}{\partial \xi} \right)}{\sum_{j=1}^{N} \sum_{t'=0}^{t_1} \left(P^b_{j,t}(\xi) \right)} \\ &= \frac{\sum_{j=1}^{N} \left(P^b_j(\xi) \left\langle \frac{\partial F(\xi)}{\partial \xi} \right\rangle_j \right)}{\sum_{j=1}^{N} P^b_j(\xi)}\end{aligned} \tag{2.101}$$

where $P^b_{j,t}(\xi)$ and $\frac{\partial F_{j,t}(\xi)}{\partial \xi}$ are, respectively, the biased probability density and the mean force of simulation j in window t. In the last line, $P^b_j(\xi)$ and $\left\langle \frac{\partial F(\xi)}{\partial \xi} \right\rangle_j$ represent the total biased probability density and average mean force of simulation j, while $\left\langle \frac{\partial F(\xi)}{\partial \xi} \right\rangle_{patch}$ is the patched average mean force.

This result is significant because no other established methodology can combine independent MetaD simulations in a fully self-consistent way.

When a second simulation with settings similar to the first is performed, and both are patched together, the resulting FES and its absolute deviation (AD) from the analytical reference are presented in Figure 2.10:

The overall appearance of the patched FES and its AD closely resembles the single-simulation result in figure2.9, which is expected because the simulation setup and the resulting sampling were similar in both cases. Chapter 3 will discuss how these methods can be further refined to yield more accurate surfaces.



(a) (b)

Figure 2.10: (a) FES of the patched simulations. (b) Average deviation of the FES from the analytical surface.

Chapter 3

Extending Mean Force Integration

The MFI method introduced in Section 2.5 provides a novel strategy for analysing MetaD simulations and offers a self-consistent procedure for merging results from independent MetaD simulations. There is, however, considerable scope for further extending MFI methods. This chapter presents the advancements realised by the author:

- The use of multiple bias potentials across several independent simulations, within a single simulation, or both (Section 3.1).
- The real-time evaluation of the convergence of the mean force, which is compared to the error of the FES (Section 3.2).
- The implementation of a fast and accurate integration scheme based on Fourier Transforms (Section 3.3).

The effectiveness of these approaches is then demonstrated through the simulation of analytical potentials and the conformational changes of alanine dipeptide.

The contents of this chapter were published in the *Journal of Chemical Theory and Computation* [77].

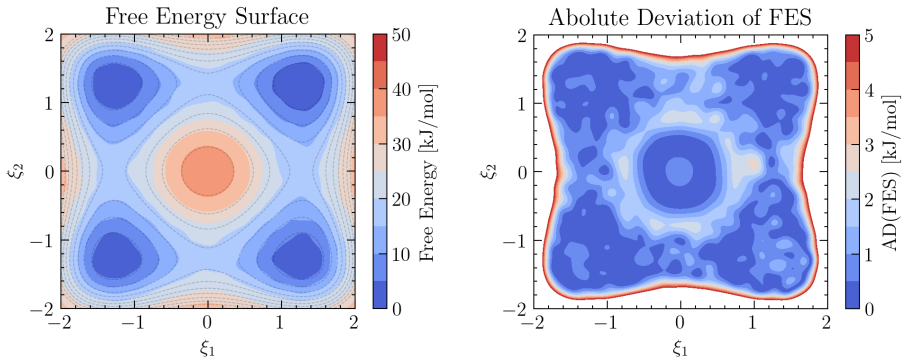
3.1 Enhanced Sampling Subject to Various Bias Potentials

MetaD, Umbrella Sampling (US) and Umbrella Integration (UI) share the same fundamental principle: the introduction of a perturbation to the potential energy. Considering equations 2.77 and 2.89, the distinction among these methods is the form of the bias potential and the constant term appearing at the end of the equations. The constant term cancels out when evaluating the mean force rather than the free energy, thereby simplifying the unbiasing process. The bias potential, on the other hand, is a history-dependent repulsive potential in MetaD, whereas in US it is a static attractive potential. The two types of potentials are typically not combined. A notable exception is the "well-sliced" approach of Awasthi *et al.*, which applies a fixed harmonic umbrella to one CV while biasing an orthogonal CV with WT-MetaD, demonstrably speeding up convergence on flat, high-dimensional free-energy landscapes [78].

3.1.1 Combining Simulations that use Different Biases

The MFI equations introduced in Section 2.5 assume that the mean force is evaluated using a MetaD bias potential. However, the bias potential, $V_t(\xi)$, in equation 2.89 is not restricted to MetaD and can be replaced with a harmonic restraint. The resulting expression then resembles the equation for calculating the mean force in UI 2.78. Because both approaches evaluate the mean force, they can be combined using the weighted average method given in equation 2.101, which effectively allows the combination of MetaD and US simulations. This could be advantageous in situations where most of the FES is converged with an initial MetaD simulation, which is complemented by a US simulation that focuses sampling in the less converged region.

To demonstrate this ability, the protocol from Section 2.5.3 (two similar MetaD were patched) was repeated, but the second simulation was replaced



(a) (b)

Figure 3.1: Figure (a) shows the free energy surface, and Figure (b) shows the average deviation of the FES to the analytical surface.

with a US simulation. Since the first simulation exhibited a large error around $\xi = (0, 0)$ (see Figure 2.9), a harmonic restraint was applied at that location. The resulting patched FES and its average deviation from the analytical surface (AD) are shown in Figure 3.1. Comparison with Figure 2.10 from Section 2.5.3 shows that the patched simulation with the combined bias displays a lower error, particularly around the centre of the restraint.

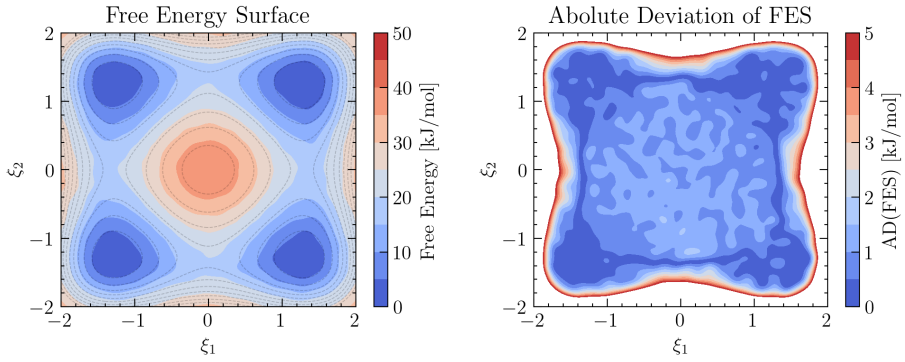
3.1.2 Unbiasing Simulations Subject to Multiple Biases

As outlined in Section 3.1.1, the bias potential, $V_t(\xi)$, in equation 2.89 is not limited to MetaD. In fact, it can include multiple bias terms, for example, a MetaD bias combined with a harmonic restraint of the form $\frac{\kappa}{2}(\xi - \xi^{ref})^2$. The unbiased mean force of such a composite bias is expressed as:

$$\frac{\partial F_t(\xi)}{\partial \xi} = -k_B T \frac{\partial \ln\left(P_t^b(\xi)\right)}{\partial \xi} - \left[\frac{\partial V_t(\xi)}{\partial \xi} + \kappa(\xi - \xi^{ref})\right]. \tag{3.1}$$

Including this additional harmonic term restricts the system to remain near ξ^{ref} while still enabling sampling of higher-energy regions. This approach can be particularly advantageous when exploring a region characterised by steep potential gradients. Conventional umbrella sampling typically requires careful selection of κ and ξ^{ref}, whereas the history-dependent component in MetaD offers more flexibility in parameter choices.

To showcase this approach, the simulation protocol from Section 3.1 (MetaD simulation patched with US simulation) was repeated, but the US simulation was replaced with a simulation that employs a MetaD bias combined with a harmonic constraint centred at $(0, 0)$. The resulting FES and its AD are shown in Figure 3.2. Comparison with Figure 3.1 from Section 3.1.1 indicates that the patched simulation using the combined bias exhibits reduced error.



(a) (b)

Figure 3.2: Figure (a) shows the free energy surface, and Figure (b) shows the average deviation of the FES to the analytical surface.

Equation 3.1 can be generalised to include an arbitrary number N_{Bias} of bias potentials:

$$\frac{\partial F_t(\xi)}{\partial \xi} = -k_B T \frac{\partial \ln\left(P_t^b(\xi)\right)}{\partial \xi} - \sum_i^{N_{Bias}} \frac{\partial V_{i,t}(\xi)}{\partial \xi} \,, \tag{3.2}$$

where $V_{i,t}(\xi)$ is the bias potential that can take any form, whether it is history-dependent or static.

In summary, it is possible to combine any number of simulations, each employing different biasing strategies. Nonetheless, every simulation must be set up carefully because even a single erroneous simulation can compromise the combined results. For this reason, it would be beneficial to track the convergence of the simulations to detect this issue if it arises. The results for this section were obtained with the Python code in Appendix B.1.3.

3.2 Convergence Evaluation

When quantifying the accuracy of the evaluated FES, the exact solution is typically not known, and it is, therefore, not possible to confirm that the correct FES was obtained. In some situations, a relatively short simulation may suffice to find the correct FES, whereas other situations require longer simulations. Even if the accuracy cannot be definitively established, it is possible to measure the convergence, which in turn provides credibility to the resulting FES.

The convergence is commonly assessed using the statistical variance, which quantifies the spread between a set of samples s_i and their average $\langle s \rangle = \frac{1}{n} \sum_i^n s_i$. The variance is given by:

$$\sigma^2 = \frac{\sum_{i=1}^{n} (s_i - \langle s \rangle)^2}{n - 1} \tag{3.3}$$

where σ^2 is the variance, and the "-1" is the Bessel correction that compensates for the bias introduced by using a subset of samples from the entire distribution. As n increases, the influence of the Bessel correction diminishes.

This section presents several methods for evaluating the variance of the FES and the mean force. The first method is block averaging, a common technique for assessing the convergence of MD simulations. This is followed by an *on-the-fly* method for estimating the convergence of the average mean force, and a *volume-normalised* error that accounts for the error increase when exploring new regions. Next, a bootstrapping approach is introduced to determine the error of the FES.

All methods are applied to evaluate the error of a Langevin dynamic simulation on the analytical potential $U(\xi) = \xi^8 - 50e^{-\frac{(\xi-1)^2}{0.1}} + 50e^{-\frac{(\xi-0.5)^2}{0.03}} - 70e^{-\frac{\xi^2}{0.03}} - 93e^{-\frac{(\xi+0.5)^2}{0.05}} - 103e^{-\frac{(\xi+1.5)^2}{0.07}}$, which is depicted in Figure 3.3. Only values below an energy cutoff of 180 kJ/mol, indicated by the red-dashed line in Figure 3.3, will contribute to the overall convergence.

3.2.1 Block Averaging

When the variance of some quantity s is calculated using equation 3.3, it is essential to ensure that the samples s_i are random. In statistics, this implies that the samples are **independent** and **identically distributed** (i.i.d.). However, in MD simulations, consecutive samples are typically correlated because they are drawn from a time-dependent trajectory. Consequently, the samples are not random, and their variance is underestimated if calculated directly.

Block averaging mitigates this problem by grouping n_b consecutive samples into N_b blocks. Each block i is assigned a block value s'_i, which is the average of all samples within that block. For example, using $n_b = 3$, each block average is obtained by combining three consecutive samples, as shown in Table 3.1).

In the general case, each block value s'_i is determined with the following expression:

$$s'_i = \sum_{j=(i-1)\cdot n_b+1}^{i\cdot n_b} \frac{s_j}{n_b} \,, \tag{3.4}$$

where j runs over the samples within block i. The total average $\langle s \rangle$ remains constant irrespective of n_b. The block variance, σ_b^2, is then determined us-

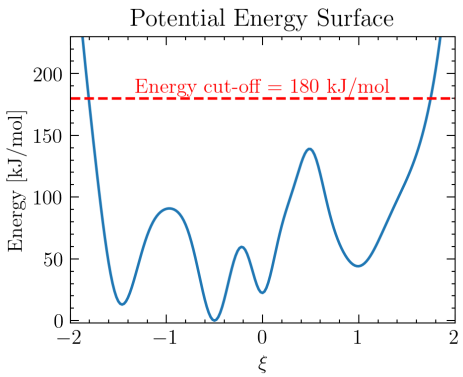


Figure 3.3: Potential energy surface (blue line) used for demonstrating convergence estimation methods. Only values below the energy cutoff (dashed line) are used for the convergence estimation.

	Data				Avr.
Correlated:	s_1, s_2, s_3	s_4, s_5, s_6	...	s_{n-2}, s_{n-1}, s_n	$\langle s \rangle$
Uncorrelated:	s'_1	s'_2	...	s_{N_b}	$\langle s \rangle$
Calculation:	$\frac{s_1 + s_2 + s_3}{3}$	$\frac{s_4 + s_5 + s_6}{3}$	...	$\frac{s_{n-2} + s_{n-1} + s_n}{3}$	$\langle s \rangle$

Table 3.1: Table illustrates the calculation of the block averages using 3 samples per block. The first row shows the raw samples, the second row shows the collective samples, and the third row shows the calculation of the collective samples.

ing:

$$\sigma_b^2 = \frac{1}{N_b - 1} \sum_{i=1}^{N_b} \left(s'_i - \langle s \rangle \right) \tag{3.5}$$

and the corresponding standard error is $\sigma_b = \sqrt{\sigma_b^2 / N_b}$.

To determine a suitable block size n_b, the variance is calculated for increasing n_b. Once the variance stabilises, the block size is deemed sufficient to decorrelate the data, reflecting the uncorrelated variance. In this work, the primary interest lies in evaluating $F(\xi)$ and the mean force, $\partial F(\xi)/\partial \xi$, both defined on a discrete grid in the collective variable (CV) space. Accordingly, their standard errors $\sigma(\xi)$ are also computed for each grid point separately.

As an example, the standard error of the mean force was computed with two different block sizes, using the simulation described at the start of Section 3.2, and the results are shown in Figure 3.4.

In Figure 3.4, it can be observed that the overall error computed with a block size of 1,000 simulation steps is slightly smaller than that obtained with a block size of 300,000 simulation steps, although they are very similar. Both results show qualitatively similar trends, namely an increased error around the largest energy barrier (around $\xi = 0.5$) and at the edges.

To facilitate direct comparison, it is helpful to average the error over the CV range (global average), producing the average standard error $\overline{\sigma_b}$. The calculation is then repeated for multiple block sizes and $\overline{\sigma_b}$ is plotted against

n_b, as shown in Figure 3.5.

As the variance plateaus at around 1 kJ/mol, it indicates that this value corresponds to the uncorrelated standard error. Nevertheless, the block size cannot be increased indefinitely because the number of samples is finite and N_b must remain large enough to yield a reliable variance estimate. Consequently, block averaging is generally feasible only for sufficiently large datasets and is typically performed after the simulation has concluded.

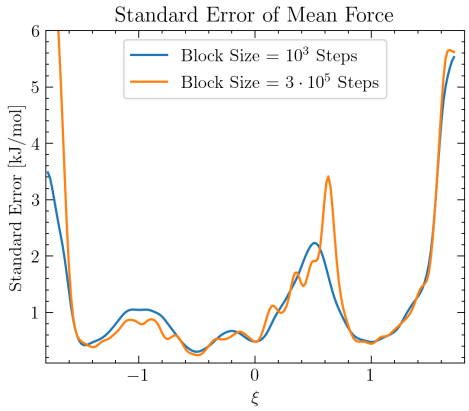
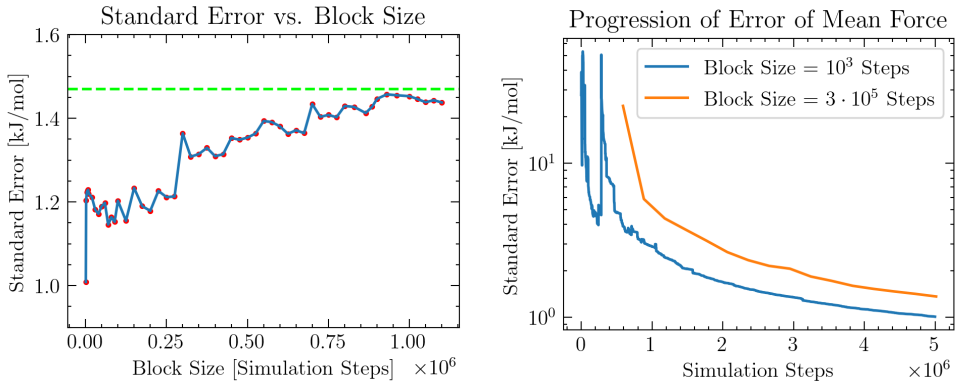


Figure 3.4: Standard deviation of the mean force as a function of ξ, calculated using block sizes of 100 simulation steps (blue line) and 10,000 simulation steps (orange line).



(a) (b)

Figure 3.5: (a) Standard error obtained via block averaging as a function of block size (blue line) and the convergence of the error (green dashed line). (b) Progression of standard deviation calculated using blocks of 10^3 simulation steps (blue line) and $3 \cdot 10^5$ simulation steps (orange line).

3.2.2 On-the-fly Variance

Although block averaging is a reliable way to determine the variance once a simulation is completed, it would be advantageous to estimate the convergence as the data becomes available (i.e. while the simulation is still running).

In the MFI method, the average of the mean force is updated every time a new time window is sampled. It would, therefore, be possible to evaluate the variance of that average at the same frequency because a weighted average is used to estimate the mean force, a corresponding weighted variance must be employed [79, 80, 81]. The weighted variance of variable x is:

$$\sigma^2 = \frac{\sum_{i=1}^{N}\left(w_i(x_i - \langle x\rangle)^2\right)}{\sum_{i=1}^{N} w_i} \cdot \frac{n_{eff}}{n_{eff}-1} = \left(\frac{\sum_{i=1}^{N} w_i x_i^2}{\sum_{i=1}^{N} w_i} - \langle x\rangle^2\right) \cdot \frac{n_{eff}}{n_{eff}-1}\,, \quad (3.6)$$

where N is the total number of samples, w_i is the weight of sample x_i, the subscript i denotes the sample index and $\langle x\rangle$ is the weighted average of all samples. The factor $\frac{n_{eff}}{n_{eff}-1}$ is analogous to the Bessel correction and accounts for the bias introduced by using unequal weights. The effective number of samples, n_{eff} , is given by:

$$n_{eff} = \frac{\left(\sum_{i=1}^{N} w_i\right)^2}{\sum_{i=1}^{N}(w_i^2)}\,. \quad (3.7)$$

The standard error is calculated as $\sigma = \sqrt{\frac{\sigma^2}{n_{eff}}}$, yielding the expression:

$$\sigma = \sqrt{\left(\frac{\sum_{i=1}^{N} w_i x_i^2}{\sum_{i=1}^{N} w_i} - \langle x\rangle^2\right) \cdot \frac{\sum_{i=1}^{N}\left(w_i^2\right)}{\left(\sum_{i=1}^{N} w_i\right)^2 - \sum_{i=1}^{N}\left(w_i^2\right)}}\,, \quad (3.8)$$

Equation 3.8 can be used to compute the weighted variance of the av-

erage mean force up to time t:

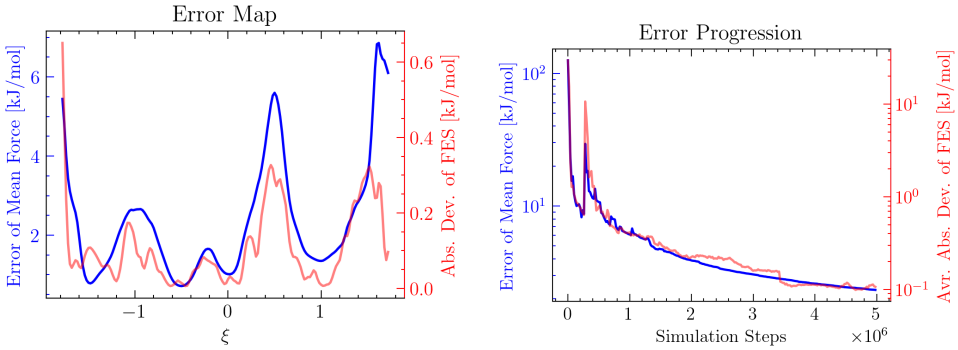
$$\sigma_t(\xi) = \sqrt{\left(\frac{\sum_{t'=0}^{t} P_{t'}^b(\xi) \left(\frac{\partial F_{t'}(\xi)}{\partial \xi} \right)^2}{\sum_{t'=0}^{t} P_{t'}^b(\xi)} - \left\langle \frac{\partial F(\xi)}{\partial \xi} \right\rangle_t^2 \right) \cdot \frac{\sum_{t'=0}^{t} \left(P_{t'}^b(\xi)^2 \right)}{\left(\sum_{t'=0}^{t} P_{t'}^b(\xi) \right)^2 - \sum_{t'=0}^{t} \left(P_{t'}^b(\xi)^2 \right)}} \tag{3.9}$$

where $\left\langle \frac{\partial F(\xi)}{\partial \xi} \right\rangle_t$ is the average mean force up to time t, $\frac{\partial F_{t'}(\xi)}{\partial \xi}$ is the mean force of time window t', and $P_{t'}^b(\xi)$ the corresponding biased probability density. Equation 3.9 enables estimation of the convergence of the average mean force while the simulation is running. Most terms appearing in equation 3.9 must be computed to calculate the mean force anyway, so the additional overhead is negligible.

Because equation 3.9 calculates the error from consecutive, and thus correlated, time windows, it underestimates the uncertainty of the mean force (see discussion in Section 3.2.1). Nonetheless, this *on-the-fly* error provides a qualitative assessment of the error of the mean force, enabling an estimation of the position-dependent and global convergence. Once a simulation has ended or run sufficiently long, the uncorrelated error of the mean force can be found via block averaging (see Section 3.2.1).

The calculation of the *on-the-fly* error is demonstrated on the simulation described at the start of Section 3.2. Figure 3.6 displays the final error map and the evolution of the global average, along with the absolute deviation of the FES from its analytical reference. These two error maps exhibit similar qualitative behaviour: both show larger errors around the energy barriers and near the edges. Their global error progressions likewise follow parallel trends, although the absolute average deviation (AAD) displays more pronounced fluctuations towards the end of the simulation. This mismatch is a result of the error propagation through the integration of the mean force. A local change in the mean force does not affect other regions directly. However, when integrated, this local change can result in shifting large parts of the FES. Furthermore, both error progressions exhibit a notable peak

after approximately 300,000 simulation steps, which corresponds to the exploration of a newly accessed basin. Before sampling this basin, the error in that region was undefined and did not contribute to the global average. When it was first sampled, the error of that region was immediately very large but rapidly decreased as it was sampled further.



(a) (b)

Figure 3.6: (a) Final *on-the-fly* error of the mean force as a function of ξ (blue line, left y-axis) and the final absolute deviation of the FES as a function of ξ (red line, right y-axis). (b) Progression of the averaged *on-the-fly* error of the mean force (blue line, left y-axis) and the averaged absolute deviation of the FES (red line, right y-axis).

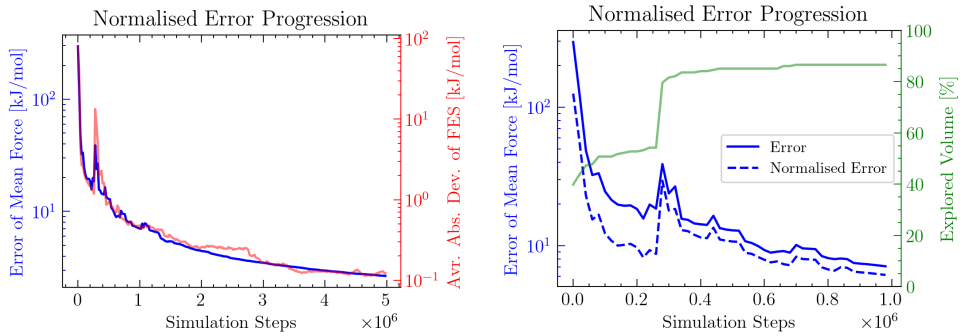
3.2.3 Volume Normalised Error

In the example of Section 3.2.2, Figure 3.6 shows an increase in the error when a previously unsampled basin is explored. However, this increase in error is a misleading representation, as the exploration of a new basin should have a favourable contribution to the overall convergence.

To address this limitation, a *volume normalised* error is introduced, in which the global error, $\overline{\sigma}$, is divided by the explored volume denoted by v. The explored volume is defined as the ratio of the sampled CV space to the total CV space under consideration.

Figure 3.7 illustrates this approach by comparing the error presented in Section 3.2.2 with its volume-normalised counterpart. Figure 3.7 (a) shows that the volume-normalised progression of the mean force error and the absolute deviation of the FES have similar profiles, consistent with the ob-

servations made in Figure 3.6. This is expected as both are scaled by $1/v$. In Figure 3.7 (b), once the new basin is sampled (and v increases significantly), the volume normalised error exhibits a smaller increase and decreases more rapidly below its pre-exploration level. By contrast, the non-normalised error remains elevated for a longer period.



(a) (b)

Figure 3.7: (a) Progression of the volume normalised averaged *on-the-fly* error of the mean force (blue line, left y-axis), and the averaged volume normalised absolute deviation of the FES (red line, right y-axis). (b) Initial progression (first 1,000,000 simulation steps) of the averaged *on-the-fly* error of the mean force (blue line, left y-axis) and the volume normalised version (blue dashed line, left y-axis), and the explored volume (green line, right y-axis).

3.2.4 Variance of Independent Simulations

Up to this point, the error calculations for a single simulation have been discussed. One of the advantages of MFI is the ability to patch multiple simulations together, making it valuable to determine the error of the combined mean force. To achieve this, the average mean forces, $\left\langle \frac{\partial F(\xi)}{\partial \xi} \right\rangle_j$, from each simulation j, with their respective probability density, $P_j^b(\xi)$, are combined using equation 2.101, which yields the patched average mean force $\left\langle \frac{\partial F(\xi)}{\partial \xi} \right\rangle_{patch}$. The weighted variance can then be computed with an expression analogous to equation 3.8:

$$\sigma_{patch} = \sqrt{\left(\frac{\sum_{j=1}^{N} P_j^b(\xi) \left\langle \frac{\partial F(\xi)}{\partial \xi} \right\rangle_j^2}{\sum_{j=1}^{N} P_j^b(\xi)} - \left\langle \frac{\partial F(\xi)}{\partial \xi} \right\rangle_{patch}^2 \right) \cdot \frac{\sum_{j=1}^{N} \left(P_j^b(\xi)^2 \right)}{\left(\sum_{j=1}^{N} P_j^b(\xi) \right)^2 - \sum_{j=1}^{N} \left(P_j^b(\xi)^2 \right)}} \tag{3.10}$$

If each simulation is initiated with distinct starting configurations and momenta, the resulting trajectories are independent, and no additional block averaging is required to remove correlations. Furthermore, the *on-the-fly* error can be computed for multiple simulations (either run in series or parallel) by patching their mean forces and probability densities and applying an expression analogous to equation 3.10. This procedure is demonstrated in Section 3.4, where the *on-the-fly* error is calculated for 20 simulations performed in series.

3.2.5 Bootstrapping Error of the FES

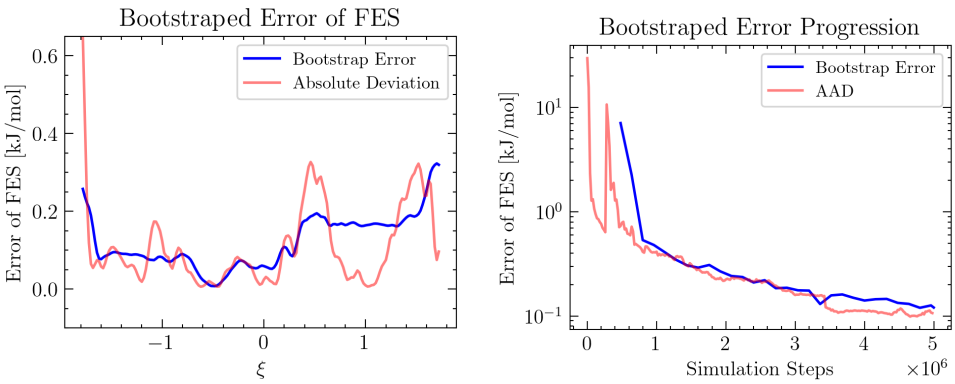
While the statistical error of the mean force is useful for estimating the convergence, the free energy is usually of greater interest. Propagating the error of the mean force throughout the integration to estimate the uncertainty in the FES is a feasible approach. However, because the error accumulates along the integration path, the uncertainty of the FES is relatively small near the reference point (e.g. $min\big(F(\xi)\big) = 0$) and grows along the integration path.

Bootstrapping the mean force offers an alternative approach and results in a better estimation of the error of the FES. Given a set of N uncorrelated mean force samples and their associated probability densities, N samples are randomly drawn with replacement, allowing for repetitions or omissions. These selected samples are then combined using a weighted average approach (see equation 2.95) and integrated to obtain an estimate of the FES. Repeating this protocol multiple times generates an ensemble of FES estimates. The uncertainty in the FES is evaluated by calculating the standard error across this ensemble.

This method is applied to the simulation described at the beginning of Section 3.2, and the resulting bootstrap error is compared with the absolute deviation of the FES, which is depicted in Figure 3.8. These two error metrics answer different questions: the bootstrap error measures the **precision** (statistical uncertainty) of the result, but can not detect systematic bias (e.g. inappropriate choice of CVs). Conversely, the absolute deviation evaluates the **accuracy** relative to the reference solution.

The two error maps, depicted in Figure 3.8 (a), indicate a similar magnitude of the error and have a significant overlap. However, the bootstrapping method slightly overestimates the error of the right basin (around $\xi = 1$), while underestimating the error of the energy barrier (around $\xi = 0.5$) separating the right basin from the rest of the surface. These discrepancies are a result of infrequent sampling of the energy barrier. The error progressions, depicted in Figure 3.8 (b), converge similarly once a sufficient amount of data is collected. Like block averaging, this procedure is not well suited to very small datasets, although the required data threshold for bootstrapping appears lower. Although this bootstrap error provides a useful evaluation of the statistical uncertainty of the FES, it should interpreted with caution, because systematic biases remain *invisible*.

While bootstrapping generally provides a robust FES error estimate, it demands more computational effort than *on-the-fly* mean-force error calculations, especially in higher-dimensional or finely resolved CV spaces. For large systems, however, the computational overhead of bootstrapping is still negligible compared to the MD simulation itself, allowing it to be performed *on-the-fly*.



(a) (b)

Figure 3.8: (a) Final bootstrap error of the FES as a function of ξ (blue line) compared with the final absolute deviation of the FES as a function of ξ (red line). (b) Progression of the global bootstrap error of the FES (blue line) and the global absolute deviation of the FES (red line).

3.3 Integration Methods

The integration of the mean force is an essential operation in MFI. Accuracy is paramount: even a well-converged mean force, if integrated with a poor algorithm, leads to an unreliable FES. The secondary consideration is computational efficiency, which becomes important when repeated evaluations of the FES are required (e.g. bootstrapping).

Although one-dimensional gradients are straightforward to integrate, higher-dimensional cases are more complex because each dimension has its own gradient. Managing these multiple gradients accurately can be challenging.

3.3.1 Finite Difference Method

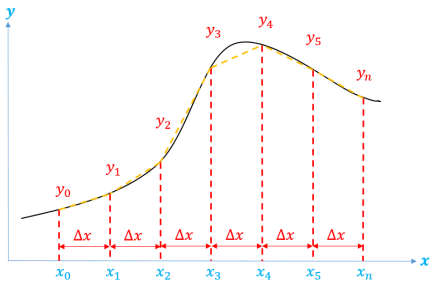


Figure 3.9: Visualisation of a finite difference integration method: the trapezoidal rule. The area under the curve is divided into N trapezoids indexed by $i = \{1, 2, ..., N\}$. The vertical edges lie along x_{i-1} and x_i; the lower edge is on the x-axis between x_{i-1} and x_i; and the upper edge is a line connecting y_{i-1} and y_i.

Finite difference methods are algebraic approximations that estimate derivatives or integrals using nearby values. When these values become infinitesimally close, the approximation converges to the exact solution. An example of finding the integral is the trapezoidal rule, which approximates a function with linear segments. Given a function, $f(x)$, defined on grid, $x = \{x_0, x_1, ..., x_N\}$, (e.g. the mean force as a function of a CV), the integral is approximated by N trapezoids, as illustrated in Figure 3.9. The integral is calculated with the following equation:

$$\int_{x_0}^{x_N} f(x)\,dx \approx \sum_{i=1}^{N} \frac{f(x_{i-1}) + f(x_i)}{2}(x_i - x_{i-1}) \tag{3.11}$$

From Figure 3.9, it can be seen that in concave sections of the curve, the area is underestimated, and in the convex sections, the area is overestimated. The accuracy can be improved by using more data points or by employing more sophisticated methods. In this work, the Simpson's rule [82] is primarily used for integrating one-dimensional gradients. For a function defined on an equally spaced grid (i.e., the grid spacing, $\Delta x = x_i - x_{i-1}$, is constant) with an even number of grid points, the integral is approximated as:

$$\int_{x_0}^{x_N} f(x)\,dx \approx \frac{\Delta x}{3}\left[f(x_0) + 4\sum_{i=1}^{N/2} f(x_{2i-1}) + 2\sum_{i=1}^{N/2-1} f(x_{2i}) + f(x_N)\right]. \tag{3.12}$$

If the total number of grid points is odd, all grid points except the last are integrated with equation 3.12, and the last grid point is accounted for by a correction term [83]:

$$\int_{x_0}^{x_N} f(x)\,dx \approx \int_{x_0}^{x_{N-1}} f(x)\,dx + \frac{\Delta x}{12}\left[5\,f(x_N) + 8\,f(x_{N-1}) - f(x_{N-2})\right]. \tag{3.13}$$

As opposed to the trapezoid rule, which approximates a function with linear segments, Simpson's rule approximates a function with segments of parabolas, which are better suited for capturing the function's curvature.

The finite difference method extends to functions defined on multiple dimensions. In the case of integrating two-dimensional gradients, one of the corners is selected as a starting point. From there, the x-gradient is integrated along the x-dimension, using a finite difference method (e.g. trapezoid rule. This yields a preliminary set of integrated values along x for a fixed y. Next, each of these integrated points is used as a starting point for integrating the y-gradient along the y-dimension. This concept is illustrated in Figure 3.10, which can likewise be performed in reverse order by first

integrating along y and then along x.

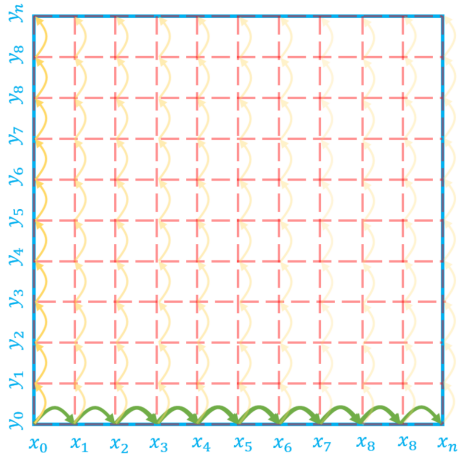


Figure 3.10: Integration in two dimensions using a finite difference integration method: Starting from a corner, the x-gradient is integrated along the x-dimension (green arrows). Next, each integrated point is a starting point for the integration of the y-gradient along the y-dimension (yellow arrows).

This method provides reasonable accuracy for smooth surfaces but becomes less reliable as the gradients become noisier. The inherent problem is that, in the presence of noise, nearby grid points can follow integration paths that yield inconsistent results. For example, the values obtained at (x_1, y_8) and (x_2, y_8) in Figure 3.10 may differ substantially if the noise along their integration paths differs. These discrepancies can lead to systematic errors (an example is depicted in Figure 3.12 (a)), making this method unsuitable for multi-dimensional data from simulations. Increasing the number of data points might reduce the mismatch, but it also increases computational cost. Consequently, a more sophisticated approach is preferable for higher-dimensional integration.

3.3.2 Least-Squares Finite Difference Method

As discussed in Section 3.3.1, conventional finite-difference methods can be unreliable for integrating multi-dimensional gradients when noise is present. A more robust solution is the least-squares finite-difference approach, in which neighbouring grid points are connected via finite-difference approxi-

mations:

$$f_x(x_i, y_j) = \frac{\partial F(x_i, y_j)}{\partial x} \approx \frac{F(x_i, y_j) - F(x_{i-1}, y_j)}{\Delta x}, \tag{3.14}$$

$$f_y(x_i, y_j) = \frac{\partial F(x_i, y_j)}{\partial y} \approx \frac{F(x_i, y_j) - F(x_i, y_{j-1})}{\Delta y}, \tag{3.15}$$

where $F(x_i, y_j)$ is a function defined on a two-dimensional grid, with $f_x(x_i, y_j)$ and $f_y(x_i, y_j)$ denoting gradients in x and y, respectively. There are a total of N_x grid points in the x-dimension, indexed by i, and a total of N_y grid points in the y-dimension, indexed by j.

This setup produces $2 \cdot N_x \cdot N_y$ equations, connecting the $2 \cdot N_x \cdot N_y$ *known* gradient values to $N_x \cdot N_y$ *unknown* values of F. This over-determined system of equations is solved using a least-squares optimisation, where F is iteratively refined to minimise the error in each equation. The resulting integral is much smoother in the presence of noise (see Figure 3.12). While this method is more reliable than the finite difference method presented in Section 3.3.1, this comes at the expense of increased computational cost.

For this work, a MATLAB implementation by John D'Errico [84] was translated to Python and employs the LSMR (Least Squares Minimal Residual) solver [85] from the SCIPY library [86].

3.3.3 Fast Fourier Transform

While the finite difference methods presented in sections 3.3.2 and 3.3.2 provide an intuitive integration algorithm, integrating the gradients into Fourier space can yield an even more robust method.

The Fourier Transform (FT) transforms a function from the spatial domain (Cartesian coordinates) to the frequency domain by decomposing it into a sum of sine and cosine terms [87]. While the Fast Fourier Transform (FFT) is applied to continuous functions, the Discrete Fourier Transform (DFT) is used for data defined on a discrete domain. Fast Fourier Transform (FFT) [88] is a highly efficient implementation of the DFT, taking advantage

of periodicity and symmetry to reduce computational costs. Due to the wide use of FFT in digital signal processing, it is a well-optimised tool available in numerous libraries. For this project, NumPy's [89] FFT module is employed in an integration algorithm, where the discrete gradients are transformed to the frequency domain, integrated, and then transformed back to the spatial domain [90, 91]. The computational complexity of the FFT scales as $O(N \log N)$ for N data points, rendering this approach efficient for surfaces defined on large grids.

3.3.3.1 Fourier Transform in One Dimension.

Given a function defined on a one-dimensional grid with N points, $x_n = \{x_0, x_1, \ldots, x_{N-1}\}$, the DFT finds the Fourier coefficients, $X_k = \{X_0, X_1, \ldots, X_{N-1}\}$, of that data [87]:

$$\begin{aligned} X_k &= \mathcal{F}(x_n) \\ &= \sum_{n=0}^{N-1} x_n \left[\cos\left(-\frac{2\pi k}{N} n \right) + i \sin\left(-\frac{2\pi k}{N} n \right) \right] \\ &= \sum_{n=0}^{N-1} x_n \exp\left(-i \frac{2\pi k}{N} n \right) \\ &= \sum_{n=0}^{N-1} x_n \exp\left(-i \omega_k n \right) , \end{aligned} \tag{3.16}$$

where there are a total of N Fourier coefficients that are indexed by k, $\omega_k = \frac{2\pi k}{N}$ is their angular frequency, $\mathcal{F}$ denotes the DFT operator, and i is the imaginary unit. The second and third lines are connected through Euler's equation: $e^{ix} = \cos x + i \sin x$. The data in the spatial domain can be obtained by applying the inverse Fourier transform operator, $\mathcal{F}^{-1}$, on the Fourier coefficients [87]:

$$\begin{aligned} x_n &= \mathcal{F}^{-1}(X_k) \\ &= \frac{1}{N} \sum_{k=0}^{N-1} X_k \exp\left(i \omega_k n \right) . \end{aligned} \tag{3.17}$$

The derivative of x_n is obtained by differentiating equation 3.17 with respect to n. Since X_k and ω_k only depends on k, the derivative is expressed as:

$$\begin{aligned}\frac{dx_n}{dn} &= \frac{1}{N}\sum_{k=0}^{N-1} X_k \frac{d\big(\exp\left(i\omega_k n\right)\big)}{dn} \\ &= \frac{1}{N}\sum_{k=0}^{N-1} i\omega_k X_k \exp\Big(i\omega_k n\Big) \\ \frac{dx_n}{dn} &= \mathcal{F}^{-1}\big(i\omega_k X_k\big)\ .\end{aligned} \tag{3.18}$$

Similarly, the integral of x_n is obtained by integrating equation 3.17 with respect to n:

$$\begin{aligned}\int x_n\, dn &= \frac{1}{N}\sum_{k=0}^{N-1} X_k \int \exp\Big(i\omega_k n\Big)\, dn \\ &= \frac{1}{N}\sum_{k=0}^{N-1} \frac{X_k}{i\omega_k} \exp\Big(i\omega_k n\Big) + C \\ \int x_n\, dn &= \mathcal{F}^{-1}\left(\frac{X_k}{i\omega_k}\right) + C\ .\end{aligned} \tag{3.19}$$

Here, C is an integration constant, which is usually set to zero in free energy calculations since only relative values are required. In practice, the integration is accomplished by a Fourier transform on the gradient data, dividing the Fourier coefficients by $i\omega_k$, and lastly using the inverse Fourier transform to obtain the integrated data [90, 91].

3.3.3.2 Extension to Multiple Dimensions.

This method can also be expanded to higher dimensions. Given a function, $x_{m,n}$, defined on a two-dimensional domain, with M data points along the first dimension indexed by m, and N data points along the second dimension indexed by n, the Fourier transform is calculated as follows:

$$
\begin{aligned}
X_{k,l} &= \mathcal{F}(x_{m,n}) \\
&= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{m,n} \exp\left(-i2\pi \left(\frac{mk}{M} + \frac{nl}{N} \right) \right) \\
&= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{m,n} \exp\left(-i(\omega_k m + \omega_l n) \right)
\end{aligned}
\tag{3.20}
$$

where there are a total of N Fourier coefficients along the first frequency dimension that are indexed by l, and M Fourier coefficients in the second frequency dimension that are indexed by k. The angular frequencies of the first and second dimensions are $\omega_k = \frac{2\pi k}{M}$ and $\omega_l = \frac{2\pi l}{N}$, respectively. The inverse Fourier transform is defined as:

$$
\begin{aligned}
x_{m,n} &= \mathcal{F}^{-1}(X_{k,l}) \\
&= \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} X_{k,l} \exp\left(i(\omega_k m + \omega_l n) \right)
\end{aligned}
\tag{3.21}
$$

In MFI, the objective is to integrate a vector field $\mathbf{x}_{m,n} = \left[x_{m,n}^{(m)}, x_{m,n}^{(n)} \right]^{\mathrm{T}}$, where $x_{m,n}^{(m)}$ represents the gradient in the m-dimension, and $x_{m,n}^{(n)}$ represents the gradient in the n-dimension, to obtain some scalar function $z_{m,n}$, which corresponds to the FES. Equation 3.22 relates $z_{m,n}$ to its known gradients via Poisson's equation:

$$
\Delta z_{m,n} = \nabla \cdot \mathbf{x}_{m,n} \, , \tag{3.22}
$$

where Δ denotes the Laplacian operator ($\Delta f = \nabla \cdot \nabla f$), and $\nabla \cdot \mathbf{x}_{m,n}$ is the divergence of the vector field, which can be expressed as:

$$
\begin{aligned}
\nabla \cdot \mathbf{x}_{m,n} &= \frac{\partial x_{m,n}^{(m)}}{\partial m} + \frac{\partial x_{m,n}^{(n)}}{\partial n} \\
&= \frac{\partial \mathcal{F}^{-1}\left(X_{k,l}^{(k)} \right)}{\partial m} + \frac{\partial \mathcal{F}^{-1}\left(X_{k,l}^{(l)} \right)}{\partial n} \\
&= \mathcal{F}^{-1}\left(i\omega_k X_{k,l}^{(k)} \right) + \mathcal{F}^{-1}\left(i\omega_l X_{k,l}^{(l)} \right) \, ,
\end{aligned}
\tag{3.23}
$$

where $\mathbf{X}_{k,l} = [X_{k,l}^{(k)}, X_{k,l}^{(l)}]^{\mathrm{T}}$ contains the Fourier coefficients of $\mathbf{x}_{m,n}$. Similarly,

the Laplacian of $z_{m,n}$ is expressed as:

$$\begin{aligned}\Delta z_{m,n} &= \frac{\partial^2 z_{m,n}}{\partial m^2} + \frac{\partial^2 z_{m,n}}{\partial n^2} \\ &= \frac{\partial^2 \mathcal{F}^{-1}(Z_{k,l})}{\partial m^2} + \frac{\partial^2 \mathcal{F}^{-1}(Z_{k,l})}{\partial n^2} \\ &= \mathcal{F}^{-1}\Big((i\omega_k)^2 Z_{k,l}\Big) + \mathcal{F}^{-1}\Big((i\omega_l)^2 Z_{k,l}\Big)\,,\end{aligned} \tag{3.24}$$

where $Z_{k,l}$ are the Fourier coefficients of $z_{m,n}$. By applying the Fourier operator on either side of equation 3.22, and using equations 3.23 and 3.23, the following expression is found:

$$\begin{aligned}\mathcal{F}(\Delta z_{m,n}) &= \mathcal{F}(\nabla \cdot \mathbf{x}_{m,n}) \\ (i\omega_k)^2 Z_{k,l} + (i\omega_j)^2 Z_{k,l} &= i\omega_k X^{(k)}_{k,l} + i\omega_j X^{(l)}_{k,l} \\ Z_{k,l} &= \frac{i\omega_k X^{(k)}_{k,l} + i\omega_j X^{(l)}_{k,l}}{-\omega_k^2 - \omega_l^2}\,,\end{aligned} \tag{3.25}$$

When applying the inverse Fourier transform to equation 3.25, the integral of some gradient $\mathbf{x}_{m,n}$ is found with the following equation:

$$z_{m,n} = \int\int \mathbf{x}_{m,n}\, dm\, dn = \mathcal{F}^{-1}\left(-\frac{i\omega_k X^{(k)}_{k,l} + i\omega_j X^{(l)}_{k,l}}{\omega_k^2 + \omega_l^2}\right) \tag{3.26}$$

In the context of MFI, where the mean forces, which are defined on a two-dimensional CV-space, $\frac{\partial F(\xi_1,\xi_2)}{\partial \xi_1}$ and $\frac{\partial F(\xi_1,\xi_2)}{\partial \xi_2}$, are integrated to find the FES, $F(\xi_1, \xi_2)$, equation 3.26 can be rewritten:

$$F(\xi_1,\xi_2) = \mathcal{F}^{-1}\left(-\frac{i\omega_k \mathcal{F}\left(\frac{\partial F(\xi_1,\xi_2)}{\partial \xi_1}\right) + i\omega_l \mathcal{F}\left(\frac{\partial F(\xi_1,\xi_2)}{\partial \xi_1}\right)}{\omega_k^2 + \omega_l^2}\right) \tag{3.27}$$

In D dimensions, a generalised version of equation 3.26 is used:

$$F(\boldsymbol{\xi}) = \mathcal{F}^{-1}\left(\frac{\sum_{d=1}^{D} \omega_d \mathcal{F}\left(\frac{\partial F(\boldsymbol{\xi})}{\partial \xi_d}\right)}{i\sum_{d=1}^{D} \omega_d^2}\right) \tag{3.28}$$

3.3.3.3 Antisymmetric Periodic Extension.

Because the Fourier transform assumes periodic boundary conditions, discontinuities can appear at the edges of the domain if the data is not inherently periodic. In gradient-based integration, these discontinuities manifest as artificial jumps in the integrated function. To mitigate this, each gradient component is extended with an antisymmetric (odd) copy in the dimension under consideration.

For a one-dimensional gradient, this is simply done by extending the gradient with a copy of it that is flipped around the force-axis and the CV-axis, as illustrated in Figure 3.11 (a). If the gradient $\nabla F(\xi)$ is originally defined over the domain $\xi \in [\xi_L, \xi_U]$, the extended gradient, defined on the extended domain $\xi^* \in [\xi_L, 2\xi_U - \xi_L]$, can be described by:

$$\nabla F_{\text{ext}}(\xi^*) = \begin{cases} \nabla F(\xi) & \text{if } \xi^* \leq \xi_U \ , \\ -\nabla F(-\xi) & \text{if } \xi^* > \xi_U \ . \end{cases} \tag{3.29}$$

After the extended gradient has been integrated, resulting in the extended integral, the final integral can be recovered from the original domain as shown in Figure 3.11 (b).

In multiple dimensions, this procedure is repeated for each dimension. The gradients are sequentially extended with a copy of themselves that is flipped around each dimension successively. If the flipped dimension corresponds to the dimension of the gradient, the extension is flipped around the force axis. Given a CV space with D dimensions, 2^D copies of each gradient are required. An illustration of this procedure is shown in Figure 3.11, where two-dimensional x- and y-gradients, shown in Figures 3.11 (c) and (d), are extended and integrated. The resulting extended integral is depicted in Figure 3.11 (e), and the final integral can be obtained from the original domain.

Even though the extended gradients have discontinuities in their abso-

lute values, their slopes remain continuous. This continuity is crucial for the FFT integration algorithm because it relies on smooth changes in the derivative to reconstruct the integral accurately.

3.3.4 Comparing Integration Methods

The accuracy of each integration method was assessed by integrating an exact (analytical) force and multiple forces that contain random noise mimicking simulation data. The noisy forces are produced by adding random fluctuations to the analytical force and applying a Gaussian filter to correlate the fluctuations in a controlled manner, thereby creating fewer but broader fluctuations. A range of fluctuations and correlation lengths were used to generate 3,200 noisy gradients, which were integrated with each method,

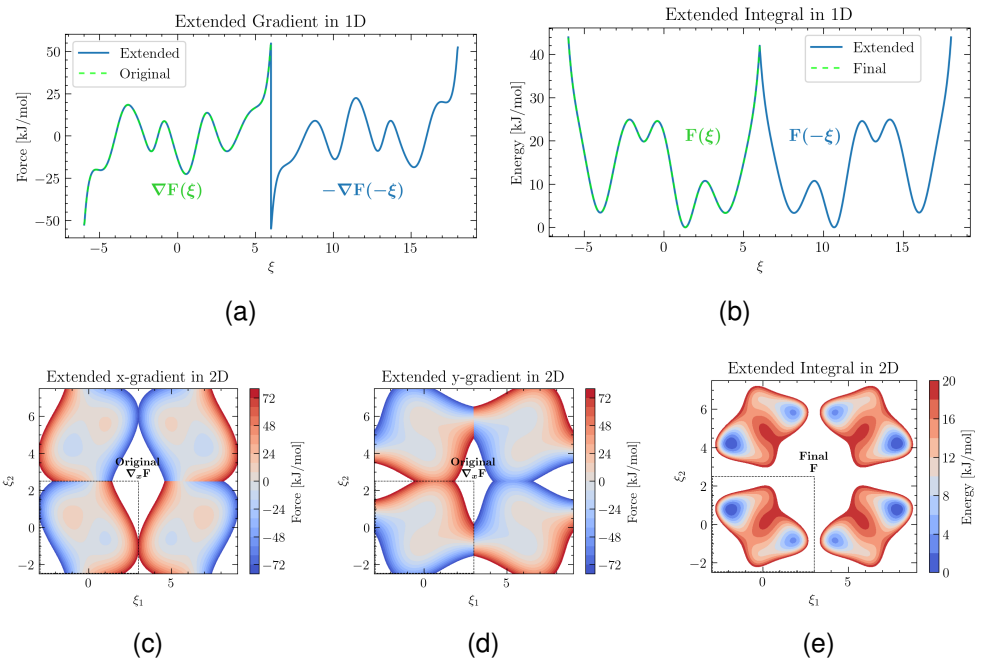


Figure 3.11: Illustration of the periodic extension that makes gradients appear continuous at the domain edges for Fourier-based integration. (a) Extension of a one-dimensional gradient (blue line) with the original gradient (green-dashed line), and (b) the resulting extended integral (blue line) with the final integral (green-dashed line). (c) Extension of the two-dimensional x-gradient with the original gradient indicated by the black box. (d) Extension of the two-dimensional y-gradient with the original gradient indicated by the black box. (e) Extended two-dimensional integral with the final integral indicated by the black box.

and their respective integration errors were averaged (More details in appendix A.1).

The first comparison involves integrating a one-dimensional potential energy surface:

$$U(\xi) = \frac{4}{3}\Big(- 14e^{-0.25(\xi+3.5)^4} - 25e^{-0.25(\xi-3.5)^4} - 10e^{-(\xi+0.5)^2} - 2\sin(-8\xi) + e^{-2\xi-9} + e^{2\xi-9}\Big) ,$$

where ξ is defined on a grid with 500 bins. The accuracy and the computational time for each method are summarised in Table 3.2. All approaches yield a very low error when integrating the analytical force. For the noisy force, each method also performs well, with Simpson's rule showing a slight edge. Simpson's rule is further distinguished by its speed, requiring only $4\,\mu\mathrm{s}$, whereas the fast FFT integration takes over 20 times longer and the least-squares finite difference method requires more than 5,000 times longer. Consequently, Simpson's rule is used predominantly to integrate one-dimensional gradients in this work.

Comparison of Integration Methods in One-Dimensional Space

Method	AAD of FES (Analytical Force)		AAD of FES (Noisy Force)		Computation time
Units	[kJ/mol]	[%]	[kJ/mol]	[%]	[μs]
Simpson's Rule	0.00016	0.00283	1.59671	12.575	4
Least-Squares Finite Difference	0.04336	0.93223	1.60280	13.093	21000
FFT integration	0.00376	0.04233	1.59649	12.570	90

Table 3.2: AAD, %AAD and computation time for various one-dimensional integration methods.

Next, the integration methods are compared for two-dimensional gradients of the potential surface [76]:

$$U(\xi_1,\xi_2) = 1.35\xi_1^4 + 1.90\xi_1^3\xi_2 + 3.93\xi_1^2\xi_2^2 - 6.44\xi_1^2 - 1.90\xi_1\xi_2^3 + 5.59\xi_1\xi_2 + 1.33\xi_1 + 1.35\xi_2^4 - 5.56\xi_2^2 + 0.90\xi_2 + 18.59 ,$$

Comparison of Integration Methods in Two-Dimensional Space

Method	AAD of FES (Analytical Force)		AAD of FES (Noisy Force)		Computation time
Units	[kJ/mol]	[%]	[kJ/mol]	[%]	[ms]
Finite Difference	0.378	6.161	2.974	118.035	490
Least-Squares Finite Difference	0.515	5.240	0.879	9.593	2200
FFT integration	0.004	0.137	0.328	7.231	21

Table 3.3: AAD, %AAD and computation time for various two-dimensional integration methods.

which is defined on a grid with 250×250 bins and is depicted in Figure 3.12 (a).

Table 3.3 summarises the accuracy and computational time for each method. The FFT approach clearly achieves the lowest error when integrating both analytical and noisy gradients. While the finite difference and least-squares finite difference methods show comparable accuracy for the analytical gradient, the finite difference approach performs poorly on noisy data due to its vulnerability to inconsistent noise along integration paths (see Section 3.3.1 and Figure 3.12 b). The FFT method is also the fastest, requiring only 21 ms, whereas the finite difference method is more than 20 times slower, and the least-squares method is over 100 times slower. Consequently, FFT integration is chosen for two-dimensional gradients in this work.

The reduced error for integrating the noisy gradients with the FFT method partly results from small fluctuations not being propagated through the Fourier transform. Hence, it is important to use sufficiently fine grids in scenarios where sharp features are expected in the FES. Moreover, the least-squares integration relies on scipy optimisation tools [86, 85], which could be replaced with custom implementations to achieve better performance. However, this was not prioritised in this work, because the FFT method performs this well.

To give an impression of the advantage of the FFT integration, a noisy mean force from a non-converged MD simulation was integrated with each method, and the resulting FESs are shown in Figure 3.12. The FES integrated with the finite difference method (Figure 3.12 b) shows significant distortions caused by inconsistent noise along different paths in the x-gradient. The least-squares approach (Figure 3.12 c) yields a smooth surface but underestimates the free energy of the right basin. In contrast, the FFT method (Figure 3.12 d) results in an FES that most closely resembles the analytical reference (Figure 3.12 a).

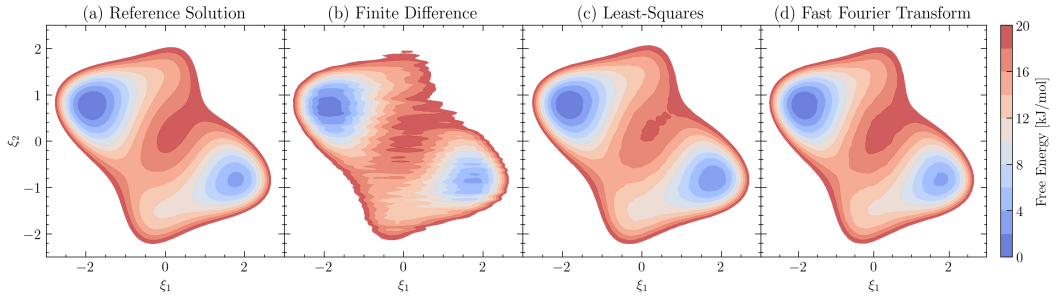


Figure 3.12: Comparing the integration of noisy two-dimensional gradients using various methods. (a) Reference solution, (b) finite difference integration, (c) least-squares finite difference integration, and (d) fast Fourier transform integration.

3.4 Testing the Methods on Alanine Dipeptide

Alanine dipeptide was chosen as a test system to evaluate the methods introduced in this chapter. This small organic molecule is often used to benchmark enhanced sampling techniques [37, 92, 73, 77]. Its conformational free energy depends on the Ramachandran dihedral angles Φ and Ψ, depicted in Figure 3.13. The two configurations illustrated in Figure 3.13 are Alanine Dipeptide's two stable states, which are separated by a large energy barrier (see Figure 3.14). The energy barrier is so large that it is highly unlikely to cross it in an unbiased MD simulation, making enhanced sampling methods necessary. Alanine dipeptide was simulated in vacuo at

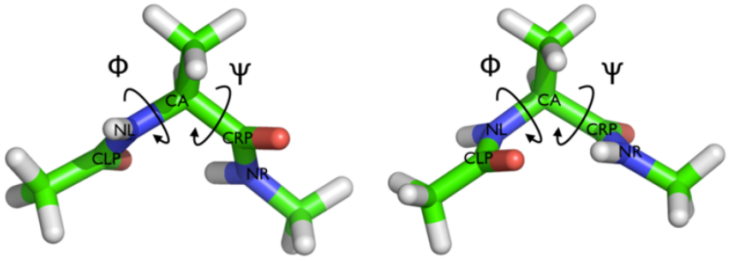
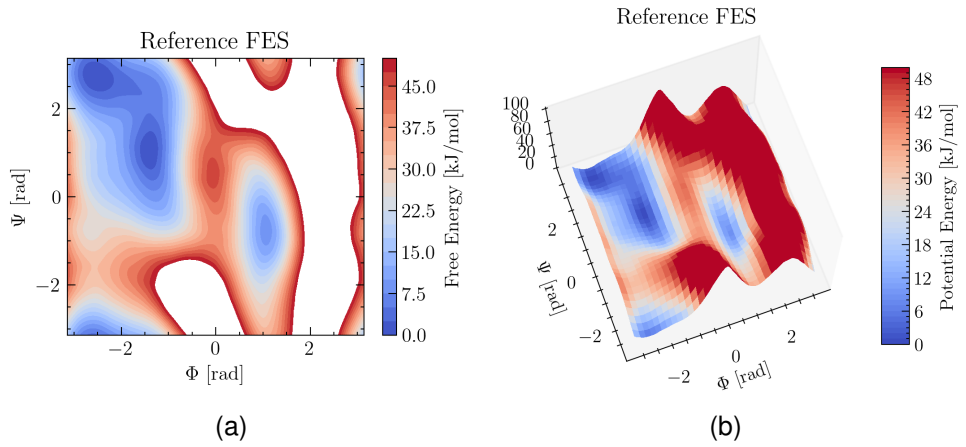


Figure 3.13: Visualisation of alanine dipeptide. Each molecule represents one of its two stable conformations. The two dihedral angles Φ and Ψ are indicated by the arrows. Figure adapted from plumed.org [4]

300 K, using GROMACS [71] with the AMBER99SB force field [68]. Newton's equations of motion were integrated using the leap-frog algorithm [70] with a time step of 1 fs. A cutoff of 1.2 nm was applied for non-bonded interactions, and electrostatic interactions were calculated using the Particle Mesh Ewald (PME) method [70]. The temperature was maintained at 300 K through a velocity-rescaling thermostat [70]. The simulation was carried out in a triclinic simulation box, with side lengths of 2.856 nm and cell angles of 60°, 60°, and 90°. The MD simulation was biased with PLUMED [93, 94].

Initially, alanine dipeptide was simulated under a metadynamics bias for 100 ns. This is a very long simulation time for such a small system, providing a high certainty that the resulting FES is converged. The data was analysed with MFI and the FES illustrated in Figure 3.14. Since there is a

high confidence that the FES is accurate, it will serve as a *reference FES* for subsequent examples where the absolute deviation (AD) is calculated.



(a) (b)

Figure 3.14: Reference FES as a function of dihedral angles Φ and Ψ.

To demonstrate the capabilities of MFIs, three different simulation protocols were conducted:

1. **Long**: A single 20 ns simulation.
2. **Short**: Twenty independent 1 ns MetaD simulations. Their starting configurations were alternated between the two stable configurations.
3. **Hybrid**: The first ten MetaD simulations from protocol (2) were combined with ten additional 1 ns simulations using MetaD and a harmonic constraint across the energy barrier.

All simulations employed WT-MetaD bias that was updated every 0.5 ps with Gaussian hills characterised by an initial height of 3 kJ/mol, a width of 0.1 rad for each CV, and a bias factor of 20. The harmonic restraint had a force constant of 40 kJ/mol/rad^2, with their centres indicated by the yellow circles in Figure 3.15 (j). The CVs were recorded every 0.05 ps, and the probability density was constructed using Gaussian kernels with a width of 0.1 rad for both dimensions.

The final results of all protocols are illustrated in Figure 3.15, where the results of the *long* simulation are shown in the first row. The results of the

combined twenty *short* simulations are shown in the second row, and the last row shows the results of the combined *hybrid* simulations. Visually, the FES of each protocol appears similar, but the AD reveals that the *long* simulation has the lowest overall error, while the *short* simulations exhibit larger deviations, particularly around the transition region ($\Phi \approx 0$). This difference can be explained by examining their probability densities: The long simulation had sufficient time to build a MetaD potential that facilitates nearly uniform sampling. On the other side, the short simulations develop the MetaD potential only briefly, leading to thorough sampling in low-energy regions but a sparse coverage of the transition region. Nevertheless, the energy difference between the two basins is estimated with an error of about 1 kJ/mol for the *long* simulation, while that error is 2.5 kJ/mol for the *short* simulations. By contrast, in the hybrid approach, the sampling of the transition region is increased considerably due to the harmonic constraints, resulting in an AD along the transition path that is comparable to that of the *long* simulations.

Additionally, the absolute deviation of the FES of each protocol (depicted in the fourth column of Figure 3.15) is normalised by the statistical error map of the 100 ns reference simulation. The statistical error is calculated via block averaging and is depicted in Figure 3.16 (a), and the resulting normalised absolute deviations are depicted in Figure 3.16 (b-d). Blue regions where the normalised absolute deviations lie within the 1σ uncertainty of the reference and are therefore statistically indistinguishable from it, whereas the red zones highlight significant discrepancies.

The block-averaged error of the reference simulation (Figure 3.16 a) is typically below 0.5 kJ/mol in the low-energy basins, and rising to about $1 - 1.5$ kJ/mol along the high-energy regions. Figure 3.16 (b) shows that the long simulation reproduces the reference FES within two standard deviations over most of the landscape, with a significant exception in the transition region around $\Phi \approx -0$ rad and $\Psi \approx 0$ rad), indicating a persistent bias where transitions remain under-sampled even after 20 ns. The set of short,

independent simulations (Figure 3.16 c) exhibits significant systematic deviations in high energy regions, white the metastable basins are reproduced within three standard deviations. The hybrid protocol (Figure 3.16 d) performs similarly well compared to the long simulation protocol, but better in the transition region around $\Phi \approx -0$ rad and $\Psi \approx 0$ rad).

These results demonstrate that multiple short simulations can be conducted instead of a long simulation, yielding converged, self-consistent free-energy estimates.

Although a single long simulation may converge faster than multiple

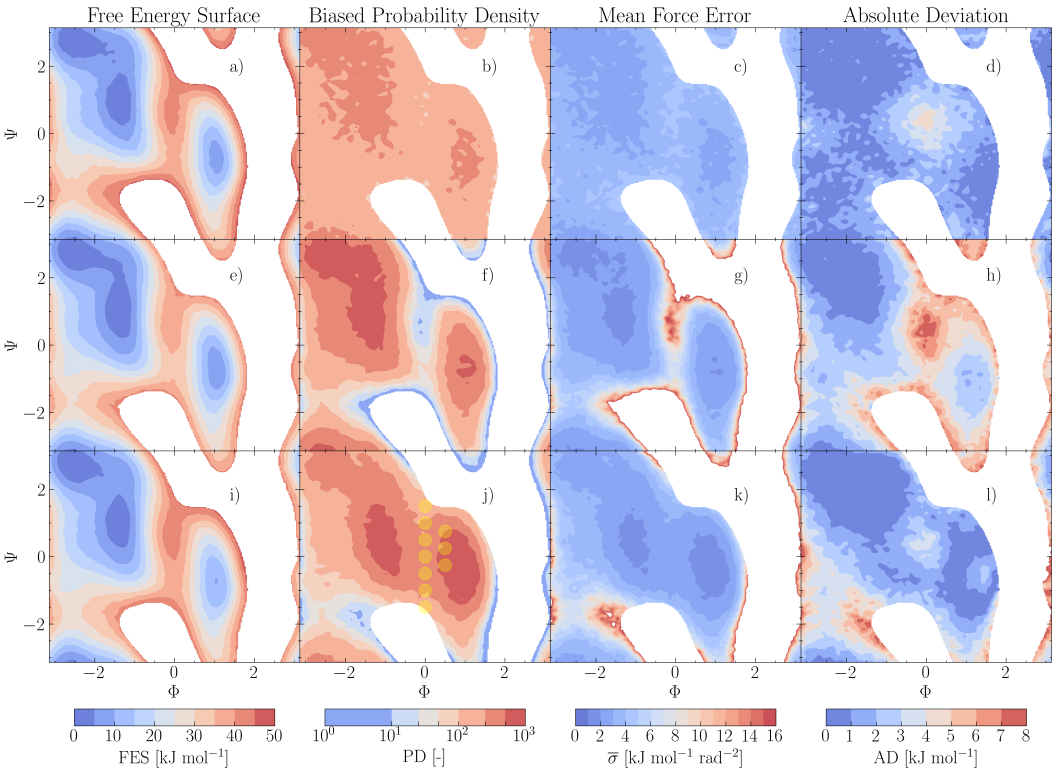


Figure 3.15: Comparison of different simulation strategies. The first row (a - d) shows results from a single 20 ns MetaD simulation. The second row (e-h) shows the combined results from twenty 1 ns MetaD simulations. The third row (i - l) shows the combined results from the first ten short simulations of the second row, plus ten more 1 ns simulations employing a MetaD bias and a harmonic constraint in one of the yellow circles of panel (j). The first column shows the final FES, the second column shows the final biased probability density, the third column shows the error of the mean force, and the fourth column shows the absolute deviation of the FES.

short ones, alanine dipeptide represents a relatively simple system compared to many of the systems typically investigated. More complex systems require substantially greater computational resources, such that a single simulation run long enough to achieve convergence might be beyond the computational resources of the researchers. Dividing the simulation into several shorter runs offers increased flexibility in resource management. This approach allows for parallel execution, which improves overall throughput, or sequential runs, which enables the optimisation of the biasing parameters.

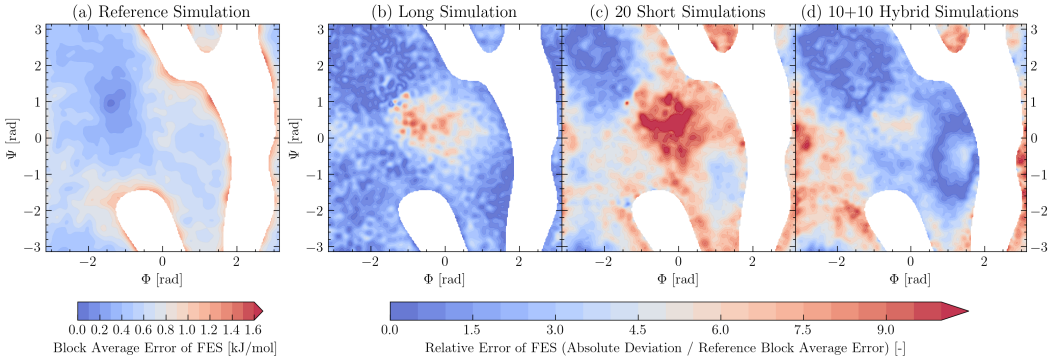


Figure 3.16: (a) Uncertainty in FES of the 100 ns reference simulation calculated with block averaging. (b-d) Normalised absolute deviations of three MFI protocols: (b) single long simulation, (c) 20 short simulations, (d) hybrid protocol. Deep blue regions denote agreement within the statistical uncertainty of the reference; red regions highlight notable discrepancies.

3.5 Conclusions

This chapter presented new developments in the MFI post-processing method. It was shown that multiple independent simulations subject to various static and history-dependent bias potentials can be merged in post-processing to produce a single coherent estimate of the FES. This capability allows combining suboptimal or incomplete simulations with new runs employing improved biasing parameters, thereby conserving simulation data that would otherwise be discarded and resampled.

An *on-the-fly* estimation of the local convergence was introduced, which evaluates the weighted variance of the average mean force in real time. This convergence estimator was shown to correlate well with the absolute deviation of the FES, as demonstrated by various analytical models and a study of alanine dipeptide. It can be further validated by a bootstrapping procedure that quantifies the statistical uncertainty in the FES through repeated sampling, patching and integration of uncorrelated blocks of mean forces. Moreover, a volume-normalised global error is proposed, ensuring that sampling a new region of CV space won't significantly increase the convergence estimation.

In combination with the fast FFT-based integration method for multi-dimensional gradients, these convergence diagnostics enable the identification of undersampled regions. If required, additional sampling can be focused on those regions via harmonic restraints or other biasing strategies, and all the simulation data can then be combined through MFI. This approach enables a more effective allocation of computational resources.

Moreover, the ability to patch multiple simulations greatly benefits computational workflows involving large chemical systems or expensive calculations (e.g. quantum-mechanical). Instead of running a single, very long simulation, the total sampling can be decomposed into smaller runs. These shorter simulations may be run in parallel, thereby increasing overall throughput. Alternatively, they can be conducted in a sequential fashion,

allowing biasing parameters to be refined systematically. Such splitting also offers the flexibility to distribute calculations across different machines or allocation periods, mitigating hardware constraints and monthly usage limits.

The applicability of these methods was demonstrated through analytical models of increasing complexity and the conformational changes of alanine dipeptide. All examples are implemented via the pyMFI Python library, which is publicly accessible at https://github.com/mme-ucl/pyMFI, making these techniques readily accessible for post-processing biased simulations for all researchers. Use cases and simple examples of the use of pyMFI to post-process biased simulations are provided within the repository and in the Appendix B.

Chapter 4

Applications to Nucleation Problems

In this chapter, the methods introduced in Chapter 3 are applied to analyse and combine multiple short simulations of nucleating systems. In section 4.2, the nucleation of supersaturated argon vapour to form a liquid droplet is investigated, and in section 4.3, the two-step crystallisation of a colloidal system is analysed.

The contents of this chapter were published by the author of this thesis in the *Journal of Chemical Theory and Computation* [77].

4.1 Relevance of Nucleation

Nucleation marks the emergence of a new stable phase (nucleus) from a metastable parent phase. This event frequently determines the rate at which first-order phase transitions occur, including the crystallisation of solids from solution or the formation of liquid droplets in vapour [95, 96]. In the chemical industry, nucleation is the initiating step in a wide range of crystallisation-based separation operations, in which a solute is selectively precipitated from the bulk solution and simultaneously converted into its final form. Therefore, precise control over this step is critical for tailoring key product qualities such as purity, particle size, and morphology [28, 36, 32]. Moreover, in natural phenomena, nucleation governs vital processes rang-

ing from ice crystal formation in clouds [97, 98] and biomineralisation in living organisms [99, 100] to the growth of minerals in geothermal reservoirs [99, 101, 102]. Therefore, a comprehensive understanding of nucleation mechanisms is essential for optimising industrial processes and elucidating the principles driving natural phenomena.

The initial formation of the nucleus is unstable, as the free energy cost associated with creating a new interface (*surface*) between the two phases exceeds the free energy reduction from converting a portion of the metastable phase into the stable phase (new *volume*) [95]. However, as the nucleus grows, the *surface-to-volume* ratio decreases, gradually reducing the relative contribution of the interfacial free energy penalty. Once the nucleus surpasses the critical nucleation size, further growth becomes thermodynamically favoured.

The driving force of the nucleation process is commonly expressed as the supersaturation, S, which is the ratio of the parent phase concentration to the saturation concentration (or parent phase vapour pressure to saturation vapour pressure for gaseous systems) [95, 103]. For realistic supersaturation conditions, the energy barrier and kinetic factors such as limited diffusion associated with nucleation are significant, rendering the process a typical *rare event*. Even when employing enhanced sampling techniques, observing multiple crossings of this energy barrier requires extensive computational resources. Such challenges motivate the development of more cost-effective strategies, including the approaches presented in the following sections.

4.2 Condensation of Supersaturated Argon

The first system being analysed is the nucleation of supersaturated argon vapour (depicted in Figure 4.1 a) to form a liquid droplet (depicted in Figure 4.1 b), which is a benchmark problem in investigating nucleation phenomena [104, 105]. Earlier work by Salvalaglio *et al.* conducted numerous

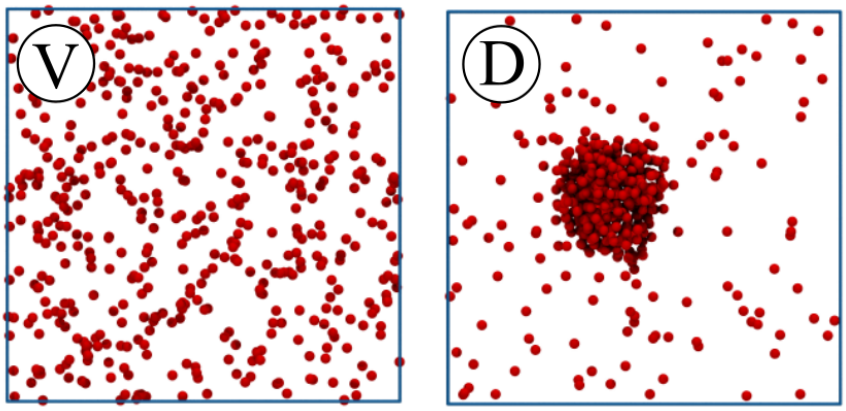


Figure 4.1: Illustration of the phase transition from (a) supersaturated argon vapour to (b) liquid argon droplet.

small-scale MetaD simulations to calculate nucleation rates and published their results in *The Journal of Chemical Physics* [103]. In this section, the simulation data obtained is analysed and merged with MFI to determine the combined FES, which was not done in the original work [103], but has been published in an article published by the author of this thesis [77].

4.2.1 Introduction

Nucleation is inherently stochastic, involving random fluctuations that form an incipient cluster. Once the cluster exceeds the critical size, further growth becomes energetically favourable. In molecular simulations, using an increased number of particles heightens the probability of observing such a rare event over a shorter simulation time. However, increasing system size also leads to higher computational costs, so there is a balance between the likelihood of capturing nucleation within a single trajectory and the total computational resources required.

A notable demonstration of large-scale brute-force nucleation simulation is a study of argon condensation employing one billion particles, simulated for 1.2 μs [105]. While their results were in good agreement with experimental studies, the computational resources required were so substantial that they were prohibitively expensive for most researchers.

Enhanced sampling strategies such as MetaD can help to accelerate transitions, but applying them effectively to nucleating systems can be challenging due to the asymmetry of the free energy barrier. For the forward transition (condensation), a steep energy barrier must be overcome, necessitating narrow bias Gaussian hills. In contrast, the backwards transition (evaporation) starts from a deeper basin with a gentler gradient, requiring larger hills (see Figure 4.2). A compromise in the shape of the hills can be counterproductive, whereas using solely short and narrow hills will require long simulation times to accumulate a sufficiently high bias. Nevertheless, by utilising the flexibility that MFI offers, it is possible to develop biasing protocols that adapt the bias to each transition pathway, as described in the following sections.

4.2.2 Methods

Salvalaglio *et al.* [103] simulated a system consisting of 512 argon atoms at 72 K under NVT conditions using GROMACS 4.6.3 [71]. The Lennard-Jones potential [106, 15] described interatomic interactions, with $\epsilon = 0.99797$ kJ/mol and $\sigma = 0.3405$ nm, truncated at $5\,\sigma$. The temperature was maintained via the Bussi-Donadio-Parinello thermostat [107], and the equations of motion were integrated with a velocity-Verlet scheme with a 5 fs time step. The system was simulated at four supersaturation conditions, which are reported in Table 4.1 together with the length of the simulation box edge, l, and the initial pressure, p.

Each supersaturation condition was examined via 50 simulations initialised in the vapour phase, which were terminated upon reaching the condensed state, and another 50 simulations starting in the condensed state, which were terminated once they reverted to the vapour phase. All simulations were biased with WT-MetaD using PLUMED 2.0 [93, 94]. For the forward transition, the bias was updated every 25 ps, using an initial height of 0.005 kJ/mol and a width of 0.25, tempered with a bias factor of 2.5. For the backwards transition, the bias was updated every 2.5 ps, with an initial

S	l [nm]	p [bar]	δ	$\omega_{0,\text{cond}}$ [kJ/mol]	γ_{cond}	$\omega_{0,\text{evap}}$ [kJ/mol]	γ_{evap}
11.95	15.93	1.27	0.25	0.005	2.5	5	50
14.03	14.60	1.49	0.25	0.005	2.5	5	50
15.57	14.04	1.66	0.25	0.005	2.5	5	50
16.86	13.62	1.79	0.25	0.005	2.5	5	50

Table 4.1: Simulation and WT-MetaD biasing parameters used to simulate the nucleation of supersaturated Argon. The parameter δ is the Gaussian width, $\omega_{0,\text{cond}}$ and γ_{cond} denote the initial height and bias factor for condensation, while $\omega_{0,\text{evap}}$ and γ_{evap} denote the corresponding parameters for evaporation.

height of 5 kJ/mol and a width of 0.25, tempered with a bias factor of 50. For the lowest supersaturation ($S = 11.95$), the energy barrier was anticipated to be particularly large, prompting the use of an additional static bias in the forward transition to help overcome the energy barrier.

The CV to describe the nucleation estimates the total number of particles in the liquid phase, N_L, which is determined with the *ten Wolde-Frenkel* criterion [108]. It is calculated with a continuous and differentiable switching function that acts on c_i, the coordination number of particle i :

$$N_L = \sum_i \frac{1 - \left(\frac{c_l}{c_i}\right)^6}{1 - \left(\frac{c_l}{c_i}\right)^{12}} , \tag{4.1}$$

where c_l is the coordination number threshold set to 5. The switching function has a value of 0 for low coordination numbers, increasing smoothly to 1 as the threshold is surpassed. The coordination number is defined as $c_i = \sum_{j \neq i} f(r_{ij})$, where r_{ij} denotes the distance between particles i and j, determined via a switching function that decays smoothly to 0 as the cutoff distance, r_c, is exceeded:

$$f(r_{ij}) = \frac{1 - \left(\frac{r_{ij}}{r_c}\right)^6}{1 - \left(\frac{r_{ij}}{r_c}\right)^{12}} . \tag{4.2}$$

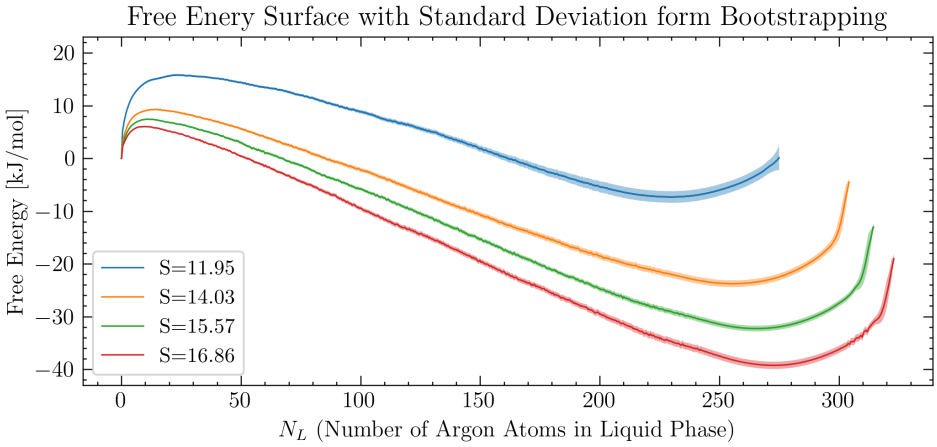


Figure 4.2: Illustration of the free energy surface of the nucleation of supersaturated argon vapour (solid line) with the bootstrapping error (shaded region). Colours represent the supersaturation levels, indicated in the legend.

4.2.3 Results and Discussion

The mean forces from all simulations were evaluated and combined with MFI to reconstruct the FESs depicted in Figure 4.2. The global convergence of the mean force was monitored with the *on-the-fly* error, as illustrated in Figure 4.3 (a). In addition, a bootstrap analysis of the independent forces provides an estimate of the uncorrelated error of the FES, which is shown as shaded regions in Figure 4.2. Furthermore, the bootstrap analysis was performed with increasing numbers of simulations, revealing the progression of the global uncertainty of the FES, presented in Figure 4.3 (b).

The shape of the FESs of the nucleation is consistent with the typical FESs of nucleation, exhibiting an initial steep energy barrier followed by a flat basin corresponding to the liquid state. The largest energy barrier of nucleation is observed at the lowest supersaturation, whereas higher supersaturations exhibit a lower barrier. In contrast, the free energy value of the basin characterising the liquid state decreases as the supersaturation increases. This behaviour aligns with physical expectations, as a higher supersaturation provides a strong thermodynamic driving force, thereby reducing the energy barrier and stabilising the nucleated phase.

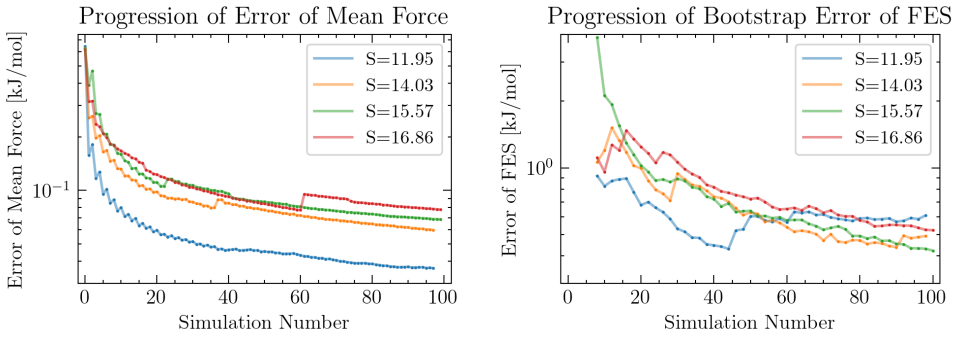


Figure 4.3: (a) *On-the-fly* error of the mean force as a function of the total simulation number and (b) bootstrapping error of the free energy surface as a function of the total simulation number. Colours represent the supersaturation conditions, indicated in the legends.

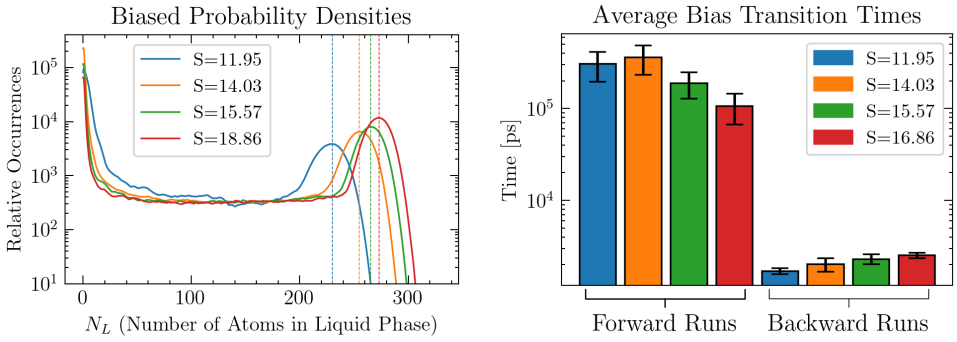


Figure 4.4: (a) Biased probability densities, where the dotted lines mark the location of FES minima (see Figure 4.2). (b) Average transition times of the forward simulations (first four bars) and backwards simulations (last four bars), together with their error bars. Colours represent supersaturation levels, indicated in the legends.

The bootstrap analysis indicates a low uncertainty in the transition regions. Higher uncertainties are observed for larger values of N_L, because the FES of the fully vaporised system is fixed at zero ($F(N_L = 0) = 0$), causing noise to accumulate along the integration path. Moreover, the biased probability densities in Figure 4.4 (a) show that the liquid states of the system with the lowest supersaturation ($S = 11.95$) were sampled the least. This results in a more significant uncertainty in that region of the FES for the system with the lowest supersaturation.

The results presented in this section highlight the versatility of MFI to overcome the practical limitations of conventional MetaD to effectively bias systems that are characterised by asymmetric free energy landscapes: the initial free energy minima, corresponding to the homogenous vapour state, is characterised by a steep gradient, whereas the second minima, corresponding to the nucleated state, has a more moderate slope but is considerably deeper. Employing exclusively small Gaussian hills in the MetaD bias would result in an accurate sampling of the minima, but would require excessively long simulation times to accumulate enough bias to escape the deeper basin. On the other hand, utilising solely large Gaussian hills would accelerate the escape from the beeper basin, at the cost of inadequate sampling of the initial basin. This renders a single MetaD simulation ineffective, as it would require either overly long simulation times or result in deficient sampling of the initial free energy minima. However, the asymmetric free energy landscape can be sampled more effectively by splitting the sampling into numerous shorter runs, each employing biasing parameters tailored for sampling one of the free energy minima. The independent biased trajectories, subject to different bias potentials, are analysed and combined with MFI and the FES is calculated. Calculating the FES from 100 independent MetaD simulations in a self-consistent manner would not have been possible with conventional methods. This illustrates that MFI, in combination with appropriately chosen biasing strategies, offers a computationally efficient route to estimate FESs in systems characterised by asymmetric energy landscapes.

4.3 Crystallisation of Colloidal System

The second system investigates a two-step crystallisation process [109, 25, 26, 29] in a colloidal system. Four separate simulations with varying bias parameters were carried out by Dietrich *et al.* [109]. The simulation data was combined with the MFI Python library developed by the author of this thesis, as well as his assistance, and published in the *Journal of Chemical Theory and Computation* [109]. This section discusses these results and presents an uncertainty analysis not included in the original publication [109], but published in an article by the author of this thesis [77].

4.3.1 Introduction

According to classical nucleation theory, crystallisation is conceptualised as the emergence of an ordered nucleus out of solution (one-step nucleation). Recent studies, however, have demonstrated that some nucleating systems undergo a two-step mechanism. In the first step, the initial homogenous solution (see Figure 4.5 a) forms an intermediate dense cluster that lacks crystalline order (see Figure 4.5 b). Thereafter, a second transformation occurs where the cluster reorganises into an ordered crystalline structure (see Figure 4.5 c).

Two CVs are required to describe such a two-step crystallisation process. The first CV, n, measures the number of particles in the dense phase, and is calculated analogously to N_L in section 4.2.2 (see equations 4.1 and 4.2). The second CV, $n(Q_6)$, measures the number of particles that exhibit a crystalline order via the sixth-order Steinhardt parameter [110, 111, 26, 109], which quantifies the sixfold symmetry around particle i. Initially, the sixth-order Steinhardt vectors are computed:

$$q_{6m}(i) = \frac{\sum_j f(r_{ij}) Y_{6m}(r_{ij})}{\sum_j f(r_{ij})} , \tag{4.3}$$

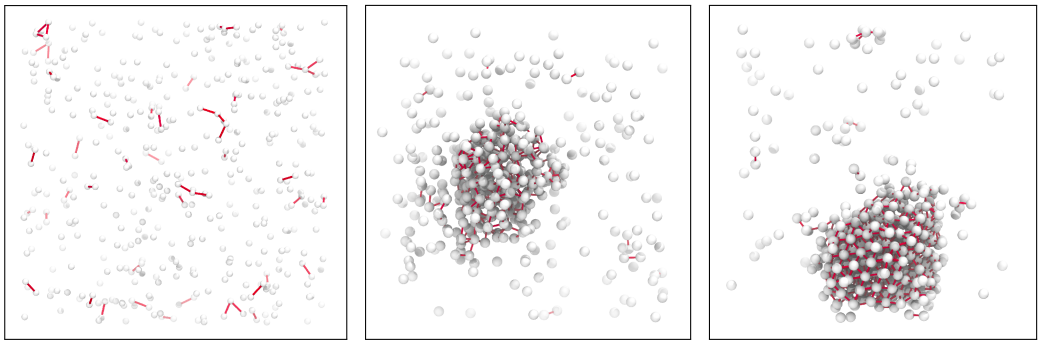
where $f(r_{ij})$ is given in equation 4.2, r_{ij} is the distance between particles i

and j and $Y_{6m}(r_{ij})$ calculates the spherical harmonics. Subsequently, each local Steinhardt parameter $Q_{6,i}$ is computed as:

$$Q_{6,i} = \frac{\sum_j f(r_{ij}) \sum_{m=-6}^{6} q^*_{6m}(i)\, q_{6m}(i)}{\sum_j f(r_{ij})}\,, \tag{4.4}$$

where $q^*_{6m}(i)$ is the complex conjugate of $q_{6m}(i)$. A switching function analogous to equation (4.1) is then used to count how many particles in the simulation have $Q_{6,i}$ values above a threshold, yielding $n(Q_6)$ [99, 109].

While calculating n poses a manageable computational cost, computing $n(Q_6)$ is significantly more demanding and can become a computational bottleneck. This issue led to the development of novel methods to approximate the CVs using Machine Learning (ML) techniques. In Dietrich's work, both CVs were approximated with ML-CVs, resulting in a significant computational speed-up. Additional details about the simulation and the ML-CVs are presented in the next section.



(a) (b) (c)

Figure 4.5: Illustration of the two-step nucleation of a colloidal system. (a) Initial homogenous phase. (b) Intermediate emergence of a dense cluster lacking crystalline order. (c) Ordered crystalline structure. The red lines indicate pairs of particles lying within the cutoff distance r_c (see equation 4.2.

4.3.2 Methods

Dietrich *et al.* [109] simulated a colloid system of 421 particles with LAMMPS [112] in an NVT ensemble with a temperature of 2T* in a cubic box of length $92.83\,\sigma$, where T* is the reduced temperature and σ the characteristic length scale. The colloidal interactions were described via a Derjaguin-Landau-Verwey-Overbeek (DLVO) potential [113, 114, 115, 99] truncated at $12.5\,\sigma$.

Four simulations were conducted, employing a MetaD bias characterised by Gaussian hills with an initial height of 0.1 kJ/mol, and a width of 0.18 and 0.05 for the n and $n(Q_6)$ CV, respectively. The first simulation was non-tempered, while the other three simulations employed WT-MetaD with a bias factor of 40, 50 and 60.

Both CVs were approximated via an ML model from the NNucleate package, which was developed by Dietrich [109, 116]. In this approach, atomic configurations are represented as graphs, where nodes correspond to individual particles and edges capture nearest-neighbour relationships. A multi-layer neural network aggregates local neighbourhood information in a permutation-invariant manner. Ultimately, a global pooling operation is applied to reduce the node features into a single vector, which is mapped by a final decoder to approximate n and $n(Q_6)$.

Calculating the CV with such a model requires pre-training, where model parameters are trained and validated on existing trajectory and CV data. Once trained, these models can be applied to systems of different sizes, as they generalise effectively across varying numbers of particles. The computational gain is considerable: for this system with 421 particles, the ML-CVs were computed roughly 3.5 faster than the direct calculation of $n(Q_6)$, and around 200 times faster for a system with $10,000$ particles [109].

The NNucleate package uses modules from the PyTorch [117], MDAnalysis [118, 119] and MDTraj [120] packages. The ML-CVs are converted with Alphabet's Jax and Flax packages [121, 122] and communicated to PLUMED using the PyCV fork [123]. Additional details are available in Diet-

rich's original publication [109] and on PLUMED NEST (https://www.plumed-nest.org/, plumID:23.026).

4.3.3 Results and Discussion

Each of the four simulations has been analysed individually using MFI, and the resulting FESs are shown in Figure 4.6, panels a) - d). In all four cases, the two-step crystallisation pathway is captured. Starting in the bottom left region, corresponding to homogenous configurations (see Figure 4.5 a), a path leads toward the first metastable basin. Along this path, n increases but $n(Q_6)$ remains low, indicating an aggregation of colloidal particles into a dense, amorphous cluster (see Figure 4.5 b). A subsequent transition leads to a deeper basin at higher $n(Q_6)$ values, representing configurations that contain dense clusters with crystalline order, as depicted in Figure 4.5 (c). This transition is characterised by an increase in $n(Q_6)$, while n remains largely unchanged, indicating that the emergence of the ordered structure requires a rearrangement of the amorphous colloidal cluster. Once the crystalline structure is formed, further growth proceeds along a diagonal path in $(n, n(Q_6))$ space, reflecting continued aggregation of colloidal particles and the ongoing development of crystalline structure.

Although the overall two-step mechanism is qualitatively consistent across all four simulations, there are notable differences in the location and magnitude of the metastable states and energy barriers. These discrepancies suggest insufficient sampling and inherent uncertainties in each individual simulation, which is confirmed by the mean force errors in Figure 4.6, panels e) - h), and the bootstrap errors of the FES in Figure 4.6, panels i) - l).

To address these uncertainties, the datasets from the four simulations were merged with MFI. The resulting combined FES is presented in Figure 4.7 (a), where the energy barrier separating the amorphous from crystalline configurations appears clearly around $n(Q_6) \approx 1$. In addition, the corresponding mean force error, depicted in Figure 4.7, and bootstrap error of

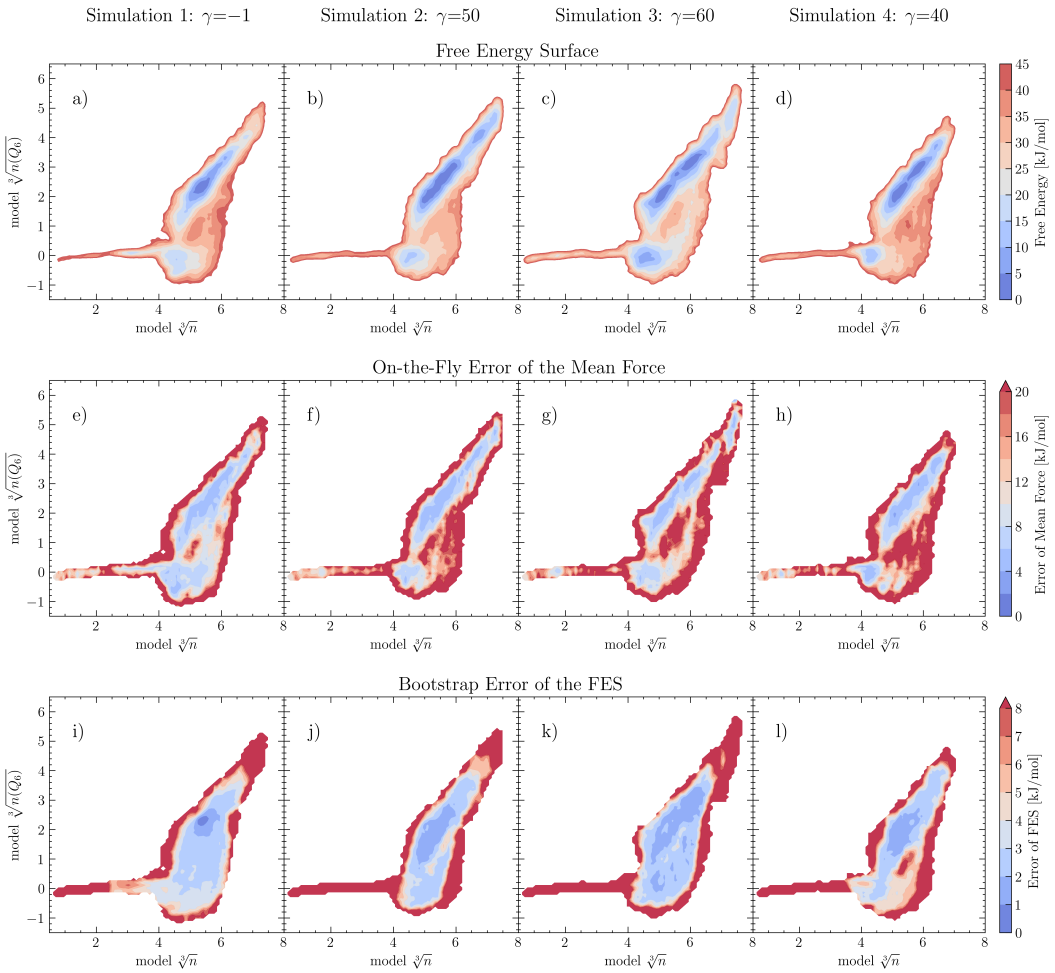
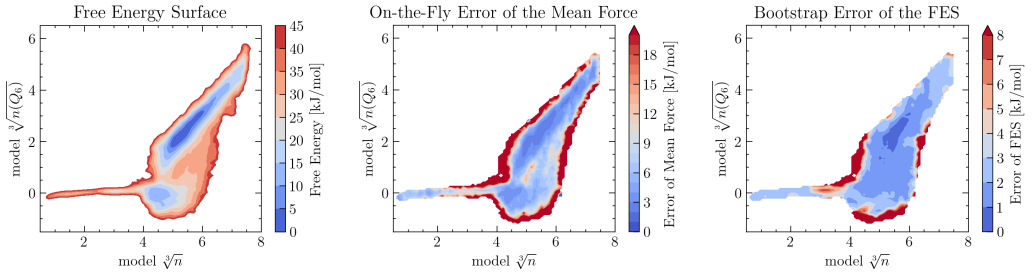


Figure 4.6: Results of the four separate simulations, obtained with MFI. The first row (panels a-d) shows the free energy surfaces, the second row (panels e-h) shows the on-the-fly errors of the mean force, and the third row (panels i - l) shows the bootstrap errors of the FES. In each row, the first simulation (non-tempered) is shown in the first column, the second simulation (bias factor=50) is shown in the second column, the third simulation (bias factor=60) is shown in the third column, and the fourth simulation (bias factor=40) is shown in the fourth column.



(a) (b) (c)

Figure 4.7: Results of the combined simulation data: (a) the FES, (b) the on-the-fly error of the mean force, and (c) the bootstrap error of the FES.

the FES, depicted in Figure 4.7, show a significant reduction in the uncertainty. In particular, the error in estimating the free energy barrier is lowered to about 1.5 kJ/mol.

The evolution of the uncertainties is illustrated in Figure 4.8. The error of the mean force, depicted as the blue line in Figure 4.8 (a), generally decreases but undergoes sudden increases. These occur when previously unsampled regions of the CV space are visited, as indicated by the green line in Figure 4.8 (a), capturing the progression of sampled volume. Figure 4.8 (b) shows the volume-normalised error of the mean force (see Section 3.2.3), where these fluctuations are almost removed. Overall, both error estimates indicate a convergence of the combined simulations. Furthermore, the progression of the bootstrap error of the FES, shown in Figure 4.8 (c), compares the convergence of the individual simulations (coloured lines) with that of the combined simulation. While the individual simulations don't appear to be converged and have a final global uncertainty estimation of $8 - 10$ kJ/mol, the bootstrap error of the combined simulations displays a converged behaviour with a final global uncertainty estimation of 4 kJ/mol.

These results demonstrate the ability of MFI to combine independent datasets collected under different biasing regimes. Although the individual simulations exhibit significant uncertainties, merging them into a single FES substantially reduces both on-the-fly and bootstrap errors. The four datasets are representative of multiple attempts to find optimal biasing parameters.

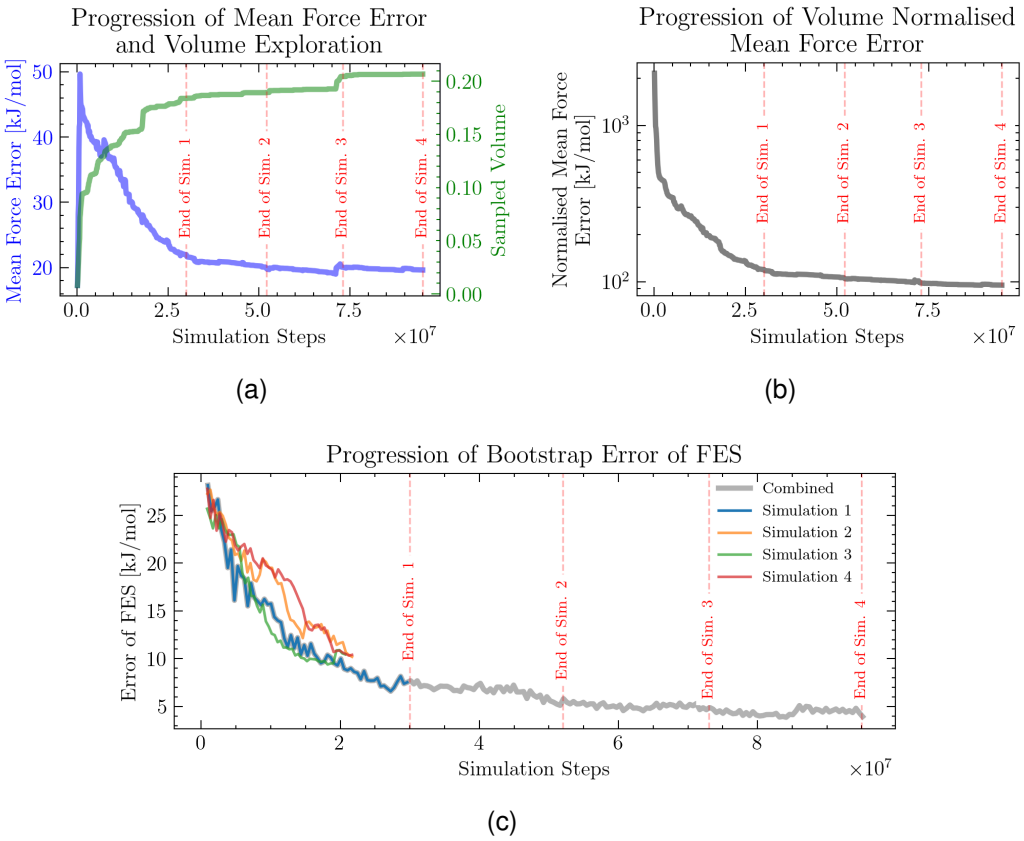


Figure 4.8: Uncertainty progression of the combined simulations, where the red-dashed lines indicate the end of each simulation. (a) Error of the mean force (blue line, left y-axis) with the ratio of the sampled volume (green line, right y-axis). (b) Volume normalised Error of the mean force. (c) Bootstrap error of the FES of the combined simulations (grey line) and individual simulations (coloured lines).

Commonly, the best dataset would be expanded, while the other datasets would be discarded. However, using MFI, all simulations can be combined to give a single FES, and additional simulation data can be added if needed, providing a robust and efficient approach for estimating the FES of complex systems.

4.4 Conclusions

This chapter has shown that MFI enables an efficient and dynamic sampling of complex nucleating systems, by partitioning the simulations into several shorter, independent runs. In Section 4.2, the asymmetric free energy landscape of the nucleation of supersaturated Argon was biased efficiently with two complementary schemes: one optimised for the forward transition and another for the reverse transition. In Section 4.3 a two-step nucleation process in a colloidal model was explored through four simulations, each conducted with different biasing parameters.

It would not be possible to merge the simulation data in a self-consistent manner using conventional post-processing tools. By contrast, with MFI, the independent trajectories are united in a straightforward fashion, yielding the combined FES. The accompanying error analysis demonstrated that incorporating additional simulations systematically reduces both the uncertainty of the mean-force error and the FES.

These results, on the one hand, underscore the advantage of employing diverse biasing strategies to optimally bias a highly irregular free energy landscape. On the other hand, they demonstrate that trajectories subject to suboptimal or exploratory biasing conditions can be integrated with other trajectories instead of discarding them altogether.

Chapter 5

Reinitialisation of Simulations and Real-Time Analysis

The preceding chapters demonstrated that independent MetaD simulations can be combined, enabling a simulation protocol employing multiple shorter simulations instead of a single long one. However, although the total number of simulation steps was identical, the convergence of the combined short simulations was worse than that of a single long simulation. This discrepancy arises because new short MetaD simulations are *uninformed* about the previously explored free energy landscape, and the MetaD bias is built independently before exploring higher energy states and potentially crossing energy barriers.

This chapter presents and evaluates biasing protocols where information is passed along subsequent simulations. Thereafter, it is demonstrated how a series of short simulations, referred to as *simulation campaigns*, can be designed by employing suitable biasing strategies to enhance the overall convergence. Subsequently, the performance of these methods is tested and evaluated against each other and conventional single-trajectory simulations.

5.1 Reinitialisation of Simulations

Carrying forward information from preceding simulations can circumvent the problem of *uninformed* new simulations through two approaches (illustrated in Figure 5.1):

1. **Reuse of the MetaD bias.** The MetaD bias constructed in the previous simulations is used as a static bias in the following simulation, which is supplemented by a new MetaD bias (see Figure 5.1 a and b).

2. **Inverse free-energy (InvF) bias.** The latest estimate of the FES is used to create a static bias for the subsequent simulations, called the *inverse FES bias* (InvF). This is accomplished by taking the FES below a specific energy limit F_{lim} and inverting it around the energy axis (see Figure 5.1 a, c, d, and e).

$$V_{\text{InvF}}(\xi) = \begin{cases} -F(\xi) + F_{\text{lim}}, & F(\xi) \leq F_{\text{lim}}, \\ 0 \qquad\quad , & F(\xi) > F_{\text{lim}}. \end{cases} \tag{5.1}$$

   The F_{lim} must be high enough to include all relevant configurations, yet sufficiently low to exclude unimportant high-energy states. In practice, overestimating F_{lim} is safer to avoid missing relevant high-energy regions.

The first approach is practical because it does not explicitly require identifying the F_{lim} parameter. However, it can introduce undesirable artefacts when the bias is combined from multiple independent MetaD simulations. By contrast, the second approach aims to create a flat bias potential energy landscape, which becomes increasingly smooth as the estimation of the FES improves. If the estimated FES is noisy, filtering (e.g., Gaussian filtering) can be applied to produce a smoother InvF. Since the underlying potential energy in most chemical systems is generally smooth, a filtered InvF yields a flatter biased potential surface (Figure 5.1 e).

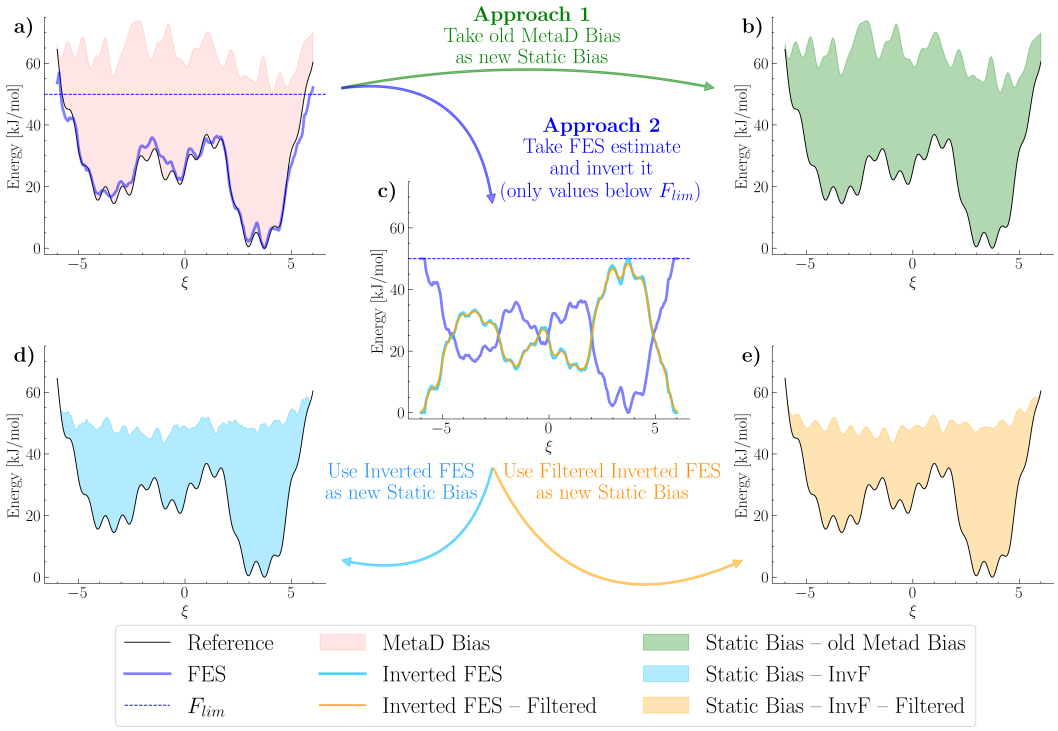


Figure 5.1: Construction of the static bias. (a) FES (blue line) with its reference solution (black line) and the MetaD bias (red-shaded region). The MetaD bias is employed as a static bias (green-shaded region) in (b), depicting the first approach. (c) FES below F_{lim} (blue line), the inverted FES (light-blue line) corresponding to the *inverse FES bias* (InvF), and the smooth (filtered) InvF (orange-dashed line). (d) static bias (light-blue-shaded region) constructed from the InvF, and (e) static bias (orange-shaded region) constructed from the smooth InvF. (d) and (e) depict the second approach.

A scaling factor of one would leave the InvF bias unchanged, while a scaling factor lower than one would compress the InvF bias. The resulting static bias would be more compressed in low-energy regions and resemble the bias obtained in WT-MetaD

The static bias may be further modified by multiplying it with a *scaling factor*, sf: $V_{\text{InvF}}^{(\text{scaled})}(\xi) = sf \cdot V_{\text{InvF}}(\xi)$, as illustrated in Figure 5.2. A scaling factor of one would leave the InvF bias unchanged, while a scaling factor lower than one would compress the InvF bias. The resulting static bias would be more compressed in low-energy regions and resemble the bias obtained in WT-MetaD, as is exemplified in Figure 5.2 (a) with a *scaling factor* of 0.9. Vice versa, Figure 5.2 (b) displays the InvF multiplied by a *scal-*

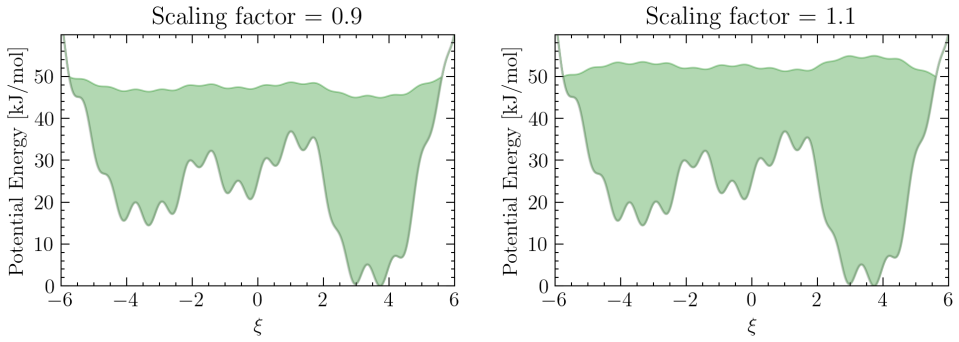


Figure 5.2: Impact of the *scaling factor* on the InvF static bias (green-shaded region). (a) InvF with a *scaling factor* of 0.9 and (b) InvF with a *scaling factor* of 1.1

ing factor of 1.1, thereby increasing the InvF bias for lower-energy regions. As a result, with a *scaling factor* below one, lower-energy regions are sampled preferentially, whereas with a *scaling factor* above one, the sampling of higher-energy regions is promoted.

5.2 New Initial Configurations

Although inheriting information on the free energy landscape is advantageous, each simulation must remain independent to recover statistically meaningful trajectories. If two simulations begin with identical starting conditions and evolve under a similar bias, their trajectories can correlate strongly. This correlation undermines the independence required for reliable statistical analysis and reduces the overall convergence. To avoid such issues, previously unused initial configurations should be employed for new simulations, which can be obtained by randomly selecting a configuration from an earlier simulation.

5.3 Simulation Stages

MFI has been introduced as a method that combines data from independent simulations, subject to arbitrary bias potentials, conducted in series, parallel, or both. This flexibility provides numerous options for designing *simulation campaigns*, but some strategies are more effective than others. To facilitate a more structured development of simulation strategies, it is helpful to divide the simulation campaign into several stages, each with a distinct objective. The python code used for this section is presented in Appendix B.2 and the full library is available in a GitHub repository.

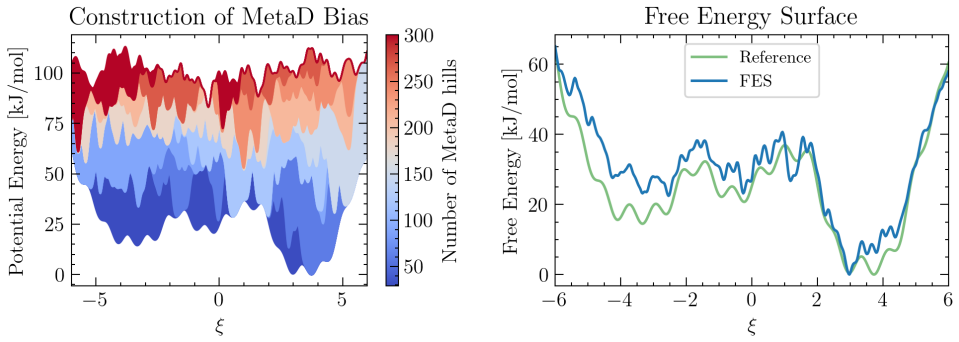
5.3.1 Exploration Stage

The initial objective of any simulation is to explore all thermodynamically relevant regions of the CV space and avoid getting trapped in metastable wells. During the *exploration stage*, the aim is to rapidly explore the CV space up to F_{lim}, as depicted in Figure 5.3. To accomplish this, an aggressive MetaD bias is used, where the Gaussian hills are deposited at a high frequency and the bias factor is set to at least F_{lim}, ensuring that the height of the MetaD Gaussians is not damped significantly (Python code in Appendix B.2.1).

The initial MetaD Gaussian height, ω_0, can be estimated by assuming that the area (or volume in higher dimensions) of the energy landscape to be filled with the MetaD potential resembles an inverted pyramid with a height of F_{lim}. Given the CV range, (CV_{max}, CV_{max}), the time between two consecutive Gaussians, τ_G, their width, σ_H, and some time budget allocated to the exploration stage, $t_{\text{exploration}}$, the Gaussian height in one dimension can be approximated by:

$$\omega = \frac{\frac{1}{2} \cdot F_{\text{lim}} \cdot (\text{CV}_{\text{max}} - \text{CV}_{\text{min}})}{\sqrt{2\pi} \cdot \sigma_H \cdot \frac{t_{\text{exploration}}}{\tau_G}}. \tag{5.2}$$

The exploration stage simulation is terminated when the FES estimate plus the MetaD bias reaches F_{lim} across all sampled CV space. It is crucial to choose a value of F_{lim} high enough to avoid missing important regions.



(a) (b)

Figure 5.3: Illustration of the Exploration stage. (a) shows the rapid build-up of the MetaD bias, where blue-shaded regions represent early hills, and red-shaded regions represent hills deposited later in the simulation. (b) compares the estimated FES (blue line) with the reference FES (green line)

The resulting FES obtained in this initial exploration stage is likely inaccurate, but sufficient for constructing the InvF bias for the next stage, enabling faster access to high-energy regions.

A further derivation for the estimation of the Gaussian height during the exploration stage for two-dimensional systems is provided in Appendix A.2.

5.3.2 MetaD Stage

Once the CV space has been explored, the FES estimate is used to construct an InvF bias for the next stage. Because this FES estimate is prone to inaccuracies and noise, the resulting biased potential surface can contain significant fluctuations, as depicted in Figure 5.4 (a) (green-shaded region), where the largest energy barrier is approximately 15 kJ/mol. A moderate Gaussian height and bias factor should be used for the MetaD bias to guarantee that such energy barriers are overcome with a single simulation (Python code in Appendix B.2.2). Moreover, the width of the MetaD Gaussians can be estimated by identifying the basins, fitting a Gaussian curve to each basin, and taking the median of the resulting widths (more details and example are provided in Appendix A.3).

The MetaD stage converges at a rate comparable to a conventional

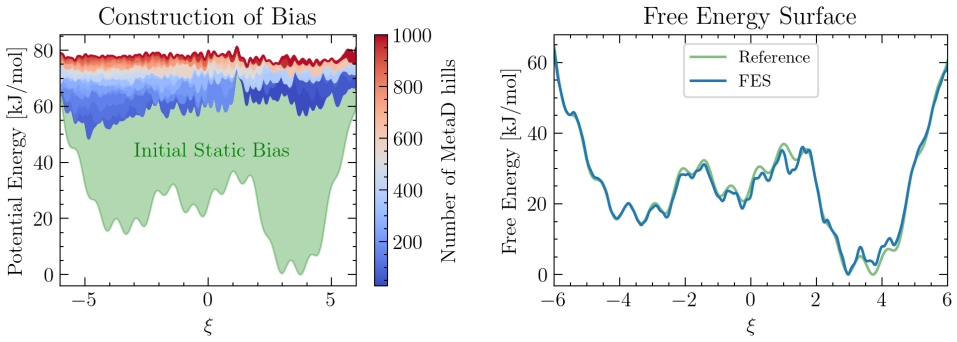
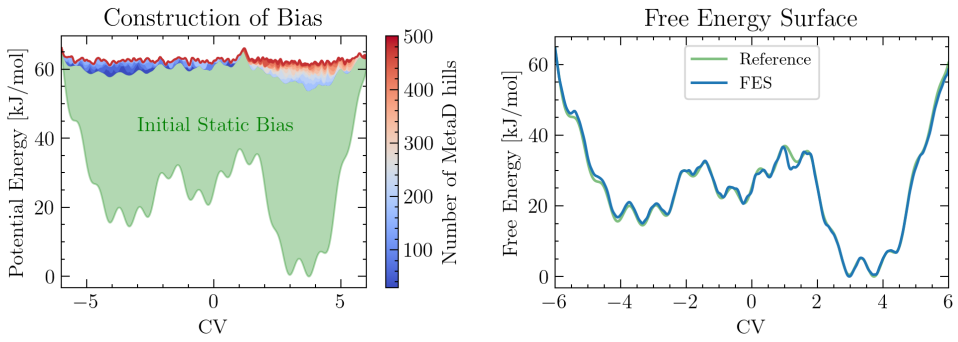


Figure 5.4: Illustration of the MetaD stage. (a) shows the initial static bias (green-shaded region) with the build-up of the MetaD bias, where blue-shaded regions represent early hills, and red-shaded regions represent hills deposited later in the simulation. (b) shows the estimate of the FES (blue line) with the reference FES (green line)

MetaD simulation and can be run without reinitialisation until the desired convergence is reached. However, if further stages are intended, a real-time convergence assessment can be utilised to terminate the MetaD stage once estimated error decreases below a chosen threshold. Ideally, the estimated error of the FES should be around $k_B T$, ensuring that the biased potential surface of the subsequent simulation contains energy barriers that can be overcome without substantial biasing.

5.3.3 Flat Stage

When the FES has reached a reasonable level of accuracy but requires additional refinement, the flat stage is employed (Python code in Appendix B.2.3). The InvF bias is now relatively accurate and generates a nearly flat biased energy surface. Consequently, there may be no need for an additional MetaD bias. However, it has been observed that a conservative MetaD bias (low bias factor, reduced Gaussian height and deposition rate) improves the convergence more reliably. Additionally, energy barriers might increase if the static bias is adjusted with the scaling factor, providing further reason to employ a conservative MetaD bias. Moreover, the improved estimate of the FES can be used to re-estimate the MetaD width through



(a) (b)

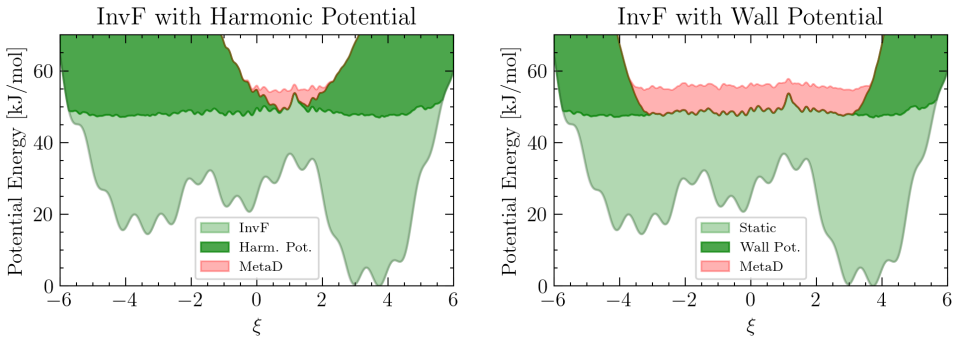
Figure 5.5: Illustration of the Flat stage. (a) shows the initial static bias (green-shaded region) with the build-up of the MetaD bias, where blue-shaded regions represent early hills, and red-shaded regions represent hills deposited later in the simulation. (b) shows the estimate of the FES (blue line) with the reference FES (green line)

Gaussian fitting of the basins.

Depending on the system being simulated, the desired accuracy and the computational resources available, the flat stage can be continued until the convergence goal is reached. Alternatively, other biasing approaches may be employed.

5.3.4 Focused Stage

In situations where the uncertainty within a specific region of CV space is particularly large, the sampling can be concentrated on that region. This can be arranged by combining the InvF bias with a harmonic restraint, as depicted in Figure 5.6 (a). The force constant of the restraint can be adjusted to reflect the width of the target region. If that region is too large to be effectively sampled with a harmonic restraint, a wall potential can be used instead. This wall potential is defined as half of the harmonic restraint and can be set up as either a lower wall or an upper wall, corresponding to the lower (left) or upper (right) portion of a full harmonic restraint. The CV region to be sampled extensively can be set up by combining the InvF bias with a lower and an upper wall (Python code in Appendix B.2.4), as illustrated in Figure 5.6 (b).



(a) (b)

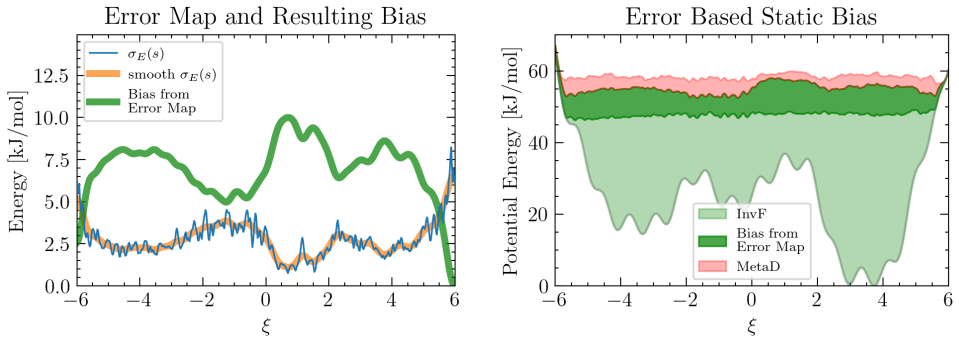
Figure 5.6: Illustration of the focused stage. The initial static bias is the light-green-shaded region, while the MetaD bias is the red-shaded region. The dark-green-shaded region in (a) indicates the harmonic potential and the wall potential in (b).

Throughout this stage, the uncertainty within the focused region should be monitored, and the simulation should be terminated once that uncertainty falls below the global uncertainty. Afterwards, sampling can either continue the focused stage in another poorly converged region, return to flat sampling, or transition to a different biasing stage.

5.3.5 Additional Stages

The stages introduced in the previous sections mirror standard enhanced-sampling practices. However, countless other biasing strategies could be implemented. For instance, the error map can be utilised to promote the sampling of less converged regions of CV space. This is achieved by smoothing the error map (e.g. with a Gaussian filter), inverting it around the energy axis, and then superimposing it on the InvF bias (Python code in Appendix B.2.5), as illustrated in Figure 5.7.

A conceptually similar approach is to construct the static bias from the probability density map instead, creating a biased energy surface that favours the sampling of regions that have been sparsely visited. In principle, the number of possible refinements is unlimited and can be tailored to the kinetics and potential energy landscape of the system under investigation.



(a) (b)

Figure 5.7: Construction of a bias potential that promotes sampling less converged regions. Figure (a) shows the error estimation (blue line), a smooth version of it (orange line), and the bias contribution from the error estimation (green line). Figure (b) shows the initial static bias (light-green-shaded region) with the static bias from the error estimation (dark-green-shaded region), and the MetaD bias (red-shaded region).

5.3.6 Transition Path Bias

A further possibility is constructing a static bias that constrains the sampling near the transition path between two stable states. In many applications, it is more relevant to sample the transition path than to explore the whole CV space uniformly, especially when dealing with multidimensional CV spaces, which grow exponentially with the number of dimensions.

The transition path can be crudely approximated by connecting the stable states with a straight line. However, connecting the stable states through the minimum free energy pathway, like the String Method [124], is much more reliable and meaningful. This not only avoids the sampling of unlikely high-energy states but also provides a representative free energy profile along the true transition path.

Once the transition path is obtained, the bias surface is created by assigning lower values to grid points near the transition path and higher values to all other points, creating a “canal” along the transition path. The resulting surface is added to the InvF, thus creating the *transition path bias*, which can be used as a static bias (Python code in Appendix B.4).

This method is demonstrated by sampling the transition path between

the stable conformations of alanine dipeptide (see Figure 3.13). An estimate of the FES from an initial simulation of alanine dipeptide, shown in Figure 5.8 (a), is used to predict the transition path, which is represented by the yellow dots in Figure 5.8 (a). That path was then used to create a *transition path bias*, and the biased potential surface is depicted in Figure 5.8 (b). This bias was employed in a second simulation of alanine dipeptide, and the data from the two simulations were combined with MFI. By comparing the absolute deviation of the FES from the initial simulation to that of the combined simulations (shown in Figure 5.8 c and d), it can be observed that the error reduced along the transition path, while it remained mostly unchanged elsewhere.

5.3.7 Results

A practical example of the simulation stages introduced above is given through the simulation of the analytical potential depicted in Figure 5.9, which is defined as:

$$\begin{aligned} U(\xi) = \frac{4}{3}\Big(&- 14e^{-0.25(\xi+3.5)^4} - 25e^{-0.25(\xi-3.5)^4} - 10e^{-(\xi+0.5)^2} \\ &- 2\sin(-8\xi) + e^{-2\xi-9} + e^{2\xi-9}\Big) . \end{aligned} \tag{5.3}$$

Several Langevin dynamics simulations are conducted, each at a constant reduced temperature of 1 (corresponding to 120 K), with a time step of 5 fs, and initialised with a random CV value.

An initial simulation in the exploration stage was conducted to explore the CV space rapidly and consisted of $1.5 \cdot 10^5$ simulation steps. It was biased with MetaD using Gaussian hills with an initial height of 20 kJ/mol, a width of 0.1, and a bias factor of 80, updated every 500 simulation steps.

Next, a simulation with $5 \cdot 10^5$ simulation steps was conducted in the MetaD stage to refine the estimate of the FES, employing an InvF bias combined with a MetaD bias using Gaussian hills with an initial height of 10 kJ/mol, a width of 0.1, and a bias factor of 10, updated every 500 simulation steps.

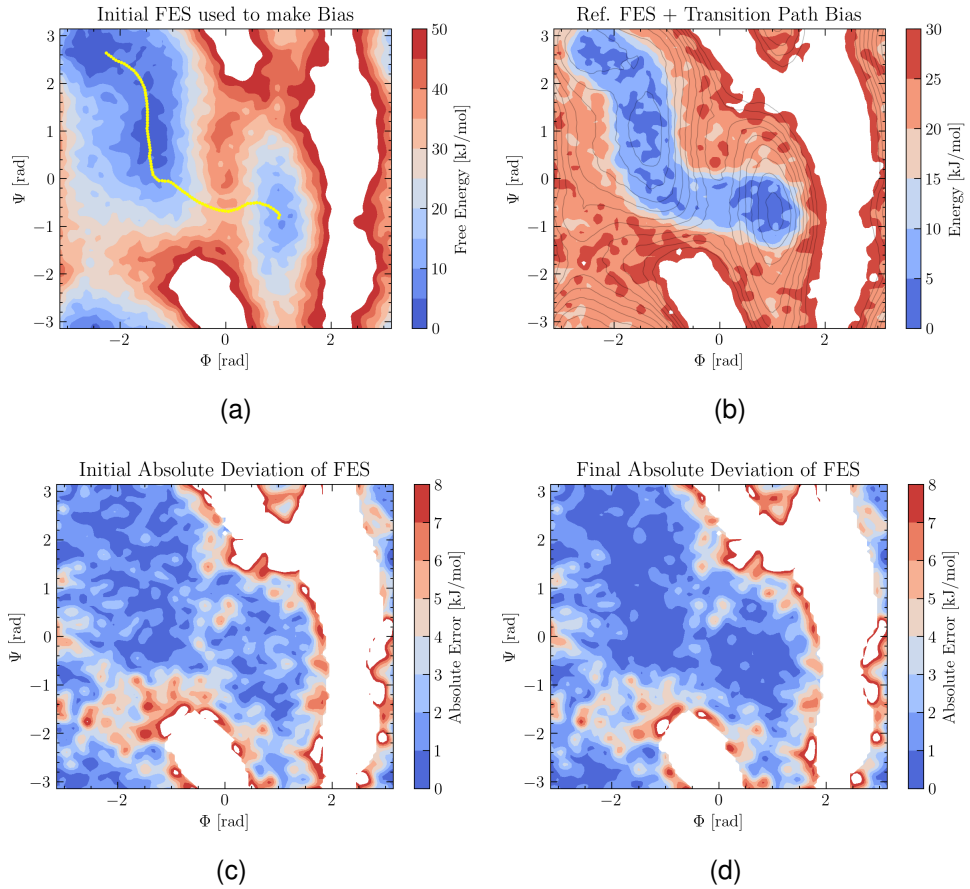


Figure 5.8: Demonstration of transition path bias. (a) The initial FES estimate used to approximate the transition path between the stable states (yellow dotted line). (b) The biased potential surface (reference FES plus transition path bias) with the reference FES indicated by the grey contour lines. (c) shows the absolute deviation of the initial simulation, and (d) shows the absolute deviation of the simulation employing the transition path bias combined with the initial simulation.

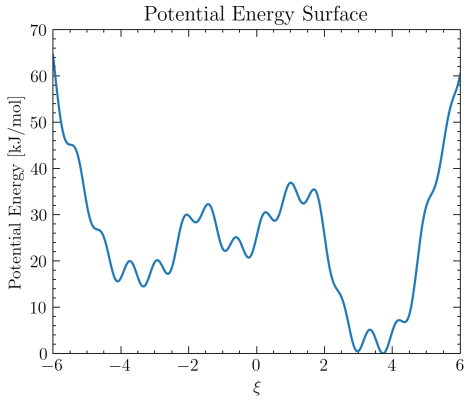
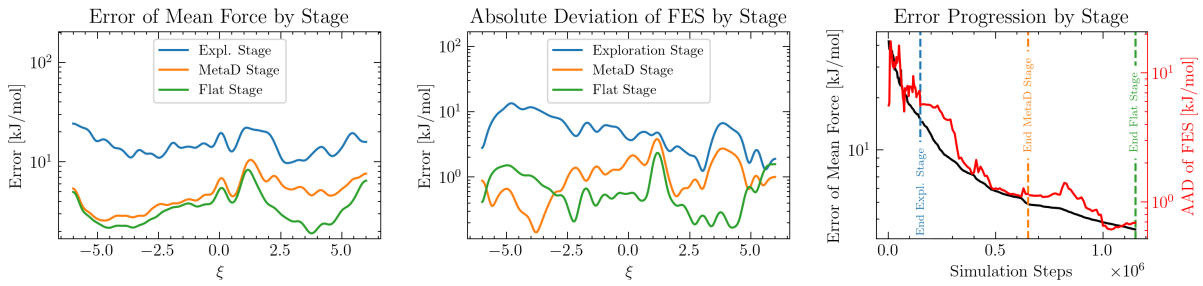


Figure 5.9: Analytical potential surface used to assess the reinitialisation methods.

After that, a simulation in the flat stage followed with $5 \cdot 10^5$ simulation steps, subject to an InvF bias combined with a MetaD bias characterised by Gaussian hills with an initial height of 1 kJ/mol, a width of 0.05 and a bias factor of 5, updated every 1000 simulation steps.

The resulting errors are illustrated in Figure 5.10, where Figure 5.10 (c) displays the progression of the global errors. It can be observed that the progression of the AAD in the exploration phase contains significant fluctuations, which are a consequence of the rapid exploration of the surface. In the MetaD stage, both convergence metrics decrease consistently, and the AAD reduces to an error of around 1 kJ/mol. At that point, the next simulation in the flat stage begins, where the errors are still decreasing, but at a slower rate. The local uncertainty of the mean force in Figure 5.10 (a) and the absolute deviation of the FES in Figure 5.10 (b) both confirm a general reduction of the error. However, the latter simulations (MetaD and Flat stage) exhibit a notably larger error around $\xi = 1$, corresponding to the location of the highest energy barrier.



(a) (b) (c)

Figure 5.10: Illustration of the error of the exploration stage (blue line), MetaD stage (orange line) and flat stage (green line). Figure (a) shows the localised error of the mean force. Figure (b) shows the localised absolute deviation of the FES to the exact solution. Figure (c) shows the progression of the global average error, where the error of the mean force is given by the black line, which is defined on the left y-axis, and the average absolute deviation of the FES is given by the red line, which is defined on the right y-axis.

Following the flat stage simulation, three additional simulations with $5 \cdot 10^5$ simulation steps were performed, with the end of the flat stage as a

starting point. In each simulation, the FES from the flat stage was used to construct an InvF bias, supplemented with a conservative MetaD bias that was updated every 1000 simulation step. It used Gaussian hills with an initial height of 1 kJ/mol, a width of 0.05. and a bias factor of 5.
The first two simulations were conducted in the focused stage, aiming to reduce the uncertainty of the region around the largest energy barrier ($\xi \approx 1$, see Figures 5.10 a and b). The first simulation employed a harmonic restraint centred at $\xi = 1$ with a force constant of $\kappa = 5$ (see figure 5.6 a), and the second simulation utilises a lower and upper wall potential centred at $\xi = -1$ and $\xi = 3$ respectively, each with a force constant of $\kappa = 10$ (see figure 5.6 b). The third simulation used a bias potential constructed with the error map, which promotes the sampling of high-error regions (explained in Section 5.3.5 and depicted in Figure 5.7).

The final error maps of each simulation are illustrated in Figure 5.11 (b-d) together with the starting error map, illustrated in Figure 5.11 (a). The error profile of the first simulation (see Figure 5.11 b) reveals a substantial improvement around $\xi = 1$, which is less pronounced in the second simulation (see Figure 5.11 c). Nonetheless, in the second simulation, the errors decreased across a larger range of the CV space because the wall potential restricted the sampling to a larger area than the harmonic potential, as indicated by the green-shaded areas. In the third simulation, the sampling is not confined to a specific area, resulting in a decrease in uncertainty across the whole CV space (see figure 5.11 d). The global error of the mean force and the global AAD of the FES, indicated by the horizontal dashed lines, show that all three simulations improved approximately equally well, albeit the third simulation has a slight advantage.

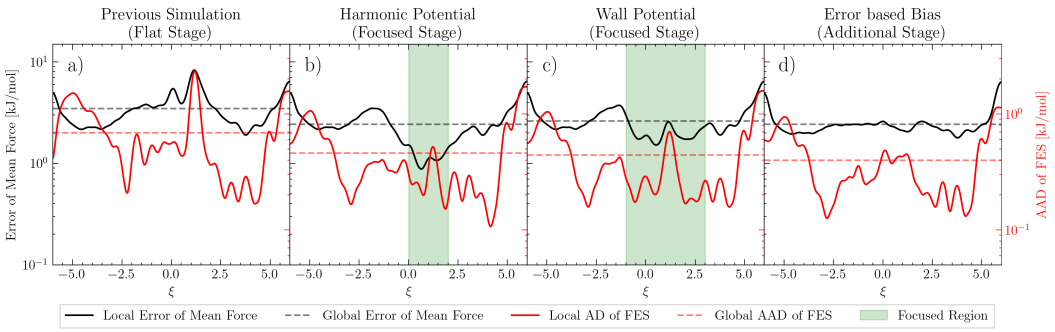


Figure 5.11: Figure shows error profile of various simulations. The black line represents the local error of the mean force, and the dashed-black line represents its global average, which are defined on the left y-axis. The red line represents the local absolute deviation of the FES, and the red-dashed line represents its global average, which are defined on the right y-axis. Figure (a) depicts the error of a simulation in the flat stage, which serves as a starting point for the other simulations. Figures (b) and (c) depict the error of simulations in the focused stage, where (b) employs a harmonic potential and (c) employs a lower and upper wall potential. The green-shaded areas indicate the focus regions. Figure (d) depicts the error of a simulation that uses a bias that promotes the sampling of high-error regions (see section 5.3.5). All error maps are smoothed with a Gaussian filter for better comparison.

5.4 Real-Time Reinitialisation

The results presented so far were obtained from simulations that were run for a pre-simulation time and analysed after they concluded. However, inefficiencies might arise when active simulations converge suboptimally, for example by over-sampling already converged regions, while other relevant regions remain weakly sampled. On the other hand, a simulation that significantly benefits the convergence might stop because the preset simulation time was reached. These inefficiencies could be avoided by analysing active simulations in real-time. Based on the progression of the real-time results, simulations are terminated autonomously when some *termination criteria* are met (summarised in Table 5.1). Subsequently, the combined results of the previous simulations are used to create a new bias for the succeeding simulation using some *initialisation rules* (summarised in Table 5.1). This is repeated until some convergence goal is met or the simulation time budget is reached. If implemented effectively, this approach may enable improved sampling distributions and offer faster convergence than a single MetaD simulation. Additionally, the results can be monitored in real time, and researchers can change the biasing strategies if they choose to.

Central to the real-time reinitialisation approach is the continuous analysis of the trajectory while it is being produced. This is done by initialising simulations with the subprocess package in Python [125], so that the simulations can run in the background and be manipulated when needed. Concurrently, a Python algorithm reads the CV and HILLS output files from PLUMED. When a given number of new CV values become available (e.g. 200 new CV values are printed), these new values are analysed and merged with the running estimate of the mean force, as well as other computations such as evaluation of convergence and FES.

The campaign always starts with a simulation in the exploration stage, such that the CV space is explored quickly. For this purpose, the MetaD Gaussian width is set to a relatively large value and the height is estimated

Stage	Initialisation rules	Termination criteria
Exploration	• MetaD bias factor >> F_{lim} • Large MetaD Gaus. width • Estimate MetaD Gaus. height	MetaD bias plus FES exceeds F_{lim} globally
MetaD	• Moderate MetaD bias factor and height • Estimate MetaD width from FES	Error of FES $\approx k_B T$
Flat	• Small MetaD bias factor and height • Reduced MetaD deposition rate • Estimate MetaD width from FES	Convergence plateaus, or move to other stage, or desired convergence reached
Focused	• Harmonic or wall bias restraint • Moderate restraint force constant • Small MetaD bias factor and height • Reduced MetaD deposition rate • Small MetaD width	Error in focused region larger than global error
Additional Stages (Error Based)	*Depends on biasing strategy Generally (and error based bias):* • Small MetaD bias factor and height • Reduced MetaD deposition rate • Moderate MetaD width	Convergence plateaus, or move to other stage, or desired convergence reached
Transition Path Sampling	• Transition path bias • Small MetaD bias factor and height	Convergence plateaus, or desired convergence reached

Table 5.1: Initialisation rules and termination criteria for various stages. The first column indicates the stage type, the second row indicates how the biasing parameters should be initialised, and the third row states the criteria for terminating a simulation.

so that the exploration finishes within 10% of the time budget of the campaign (see Section 5.3.1). The bias factor is set to a large value, for example $5 \cdot F_{\text{lim}}$, and the simulation is terminated as soon as the MetaD bias plus the FES estimate exceed F_{lim} globally.

Next, the estimate of the FES is used to construct an InvF bias for the second simulation in the MetaD stage, where the estimate of the FES is refined. Here, the width of the MetaD Gaussian can be estimated from the FES with Gaussian fitting, while the height and bias factor should be reduced, for example to $\omega_0 = F_{\text{lim}}/20$ and $\gamma = 1 + F_{\text{lim}}/10$, respectively. The MetaD stage should be continued until the uncertainty of the FES is below approximately $k_B T$, such that in the subsequent simulation, an InvF static bias (and a conservative MetaD bias) is enough to sample the relevant CV space.

This is followed by a simulation in the flat stage or one of the other stages (see Sections 5.3.3 - 5.3.6), again using an InvF static bias. The width of the MetaD Gaussian is estimated again, while the height and bias factor should be reduced further, for example, to $\omega_0 = F_{\text{lim}}/100$ and $\gamma = 1 + F_{\text{lim}}/40$, respectively. Also, the deposition rate of the MetaD bias is also reduced, resulting in a less fluctuating bias potential. At this point, the convergence, preferably the bootstrap error of the FES, $\sigma(t)$, as well as its time derivative $\Delta\sigma(t)/\Delta t$, are closely monitored. A simulation is terminated if the $\sigma(t)$ does not decrease during the simulation, or $\Delta\sigma(t)/\Delta t$ is non-negative over five consecutive checks (new data is analysed).

Additionally, two safeguards are imposed for simulations after the MetaD stage is completed: a *guaranteed* or *minimum* simulation time ensures that transient statistical fluctuations cannot trigger a premature stop, whereas a *maximum* simulation time provides an incentive to try new simulation parameters.

The methods presented in this chapter were performed with the MFI Python library, which is openly accessible on GitHub. Whereas the methods described in Chapter 3 followed a function-oriented paradigm, the reinitialisation methods presented in this chapter use an object-oriented design. The class that handles one-dimensional CV surfaces is provided at `https://github.com/mme-ucl/MFI/tree/master/MFI_class1D`, and the corresponding two-dimensional implementation at `https://github.com/mme-ucl/MFI/tree/master/MFI_class2D`. Both repositories include illustrative Jupyter notebooks that demonstrate how the classes can be used to reinitialise simulations manually or to conduct complete reinitialisation campaigns.

In the next two sections, the real-time reinitialisation approach is subdivided into *Serial Real-Time Reinitialisation* (SRTR) campaigns and *Parallel Real-Time Reinitialisation* (PRTR) campaigns. In the former, only one simulation runs at any time, which is initialised and terminated repeatedly,

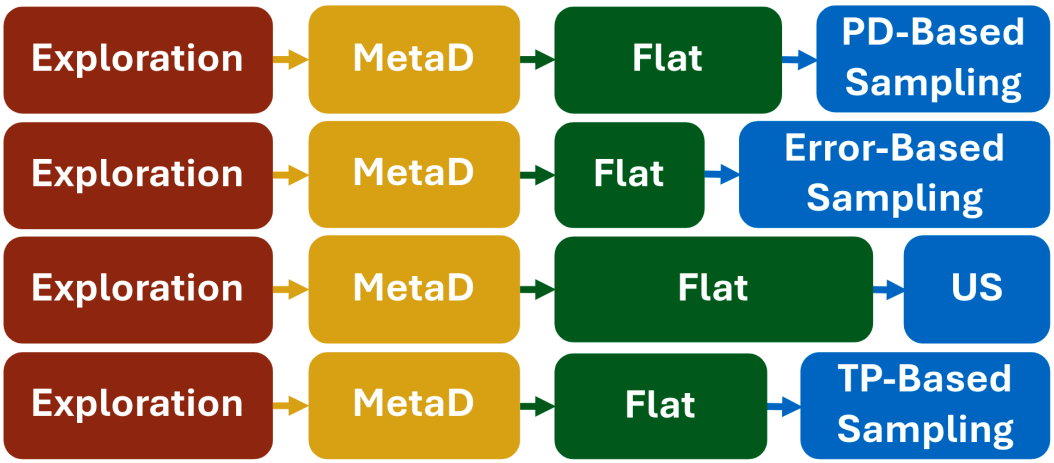


Figure 5.12: Illustration of a Parallel Simulation Campaign. Each row represents a series of simulations that are being reinitialised in new simulation stages. The red boxes indicate a simulation in the exploration stage, the yellow boxes are a simulation in the MetaD stage, the green boxes are a simulation in the flat sampling stage, and the blue boxes indicate some special sampling stage (1st row: probability density based sampling described in Section 5.3.5; 2nd row: error based sampling described in Section 5.3.5; 3rd row: umbrella sampling described in Section 5.3.4; 4th row: transition path sampling described in Section 5.3.6).

following the criteria presented in Table 5.1. In the latter, a fixed number of simulations are run synchronously and are analysed and patched in real time. The individual simulations in the PRTR campaigns are initialised similarly to those in the SRTR campaign, and are terminated by considering both the individual and the collective progress, which is explained further in Section 5.6. A simplified schematic of a PRTR campaign employing four concurrent simulations is shown in Figure 5.12, and only a single row of that figure would be a simplified representation of an SRTR campaign. In Appendix B.3, the Python code is provided, demonstrating how to run SRTR and PRTR campaigns.

In the following two sections, the real-time reinitialisation strategies are tested on the analytical potential defined in equation 5.3 and depicted in Figure 5.9 using Langevin dynamics simulations. Later, these strategies are tested on a wide range of one- and two-dimensional analytical potentials and Alanine Dipeptide.

5.5 Serial Real-Time Reinitialisation

Following the procedure introduced in Section 5.4, an SRTR campaign was simulated with a time budget of $2\,000\,000$ simulation steps and an F_{lim} of 80 kJ/mol. The individual simulations (excluding exploration stage) were limited to a minimum of 10^5 steps and a maximum of $6 \cdot 10^5$ simulation steps (Python code in Appendix B.3.1). The simulation stages and their biasing parameters were determined based on the results analysed in real time and are summarised in Table 5.2.

Stage	Simulation steps	Height [kJ/mol]	Width	Bias factor	MetaD deposition rate
Exploration	70 000	3.33	0.3	50	200
MetaD	335 000	0.67	0.123	6	200
Flat	630 100	0.17	0.122	2	500
Error based sampling	608 000	0.17	0.122	2	500
PD based sampling	367 200	0.17	0.122	2	500

Table 5.2: SRTR campaign summary: The first row presents the stages in the order they were simulated, and their simulation steps in the second column. In the remaining columns, the MetaD parameters are described, including the initial height of the Gaussian hills (third column) and their width (fourth column), the bias factor used (fifth column) and the MetaD deposition rate measured in simulation steps (sixth column). Each row represents a simulation conducted in a specific stage, with the last row indicating the probability density-based sampling stage.

The error progression of the SRTR campaign is depicted in Figure 5.13. The error rapidly decreases in the exploration and MetaD stage, but starts to plateau in the flat stage. After the flat stage, a simulation is conducted that uses an error-based InvF bias, followed by a simulation that uses a probability-density-based InvF. The last two simulations show only minor improvements in the overall error, as the simulation is almost converged. Although the error of the mean force (black line) differs from the error of the FES (red lines) in magnitude, it closely resembles the trend of the errors of the FES. Moreover, the bootstrapping error of the FES matches the average

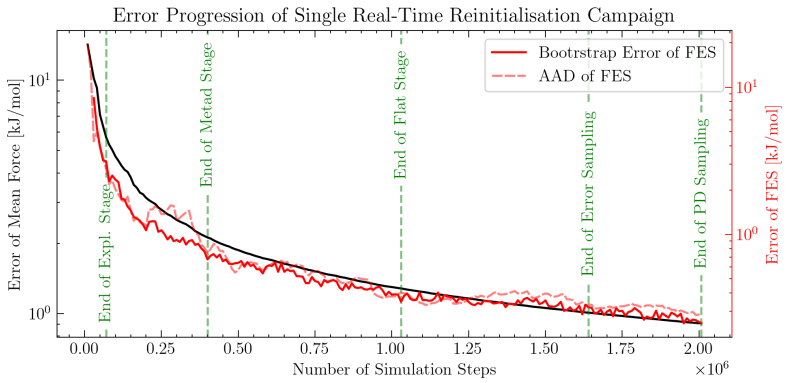
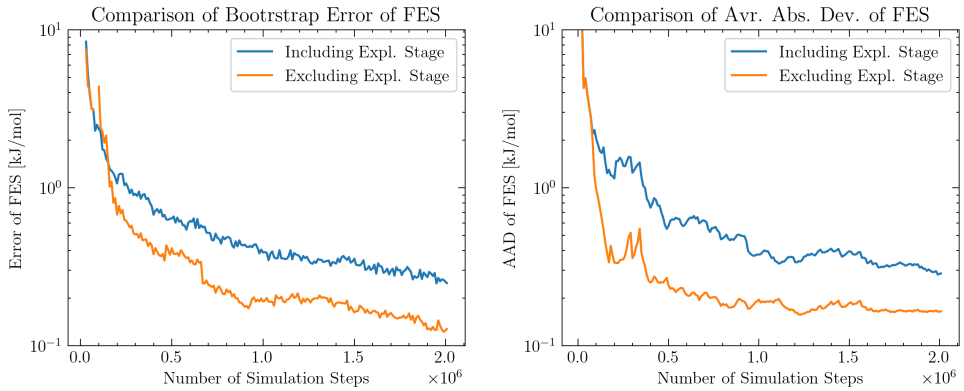


Figure 5.13: Representative error progression of a *Serial Real Time Reinitialisation* campaign. The black line indicates the error of the mean force (left y-axis), the red line indicates the bootstrap error of the FES (right y-axis), and the red-dashed line represents the average absolute deviation of the FES. Vertical green lines indicate the end of the simulation stages.

absolute deviation of the FES qualitatively and quantitatively.

Even though the objective of the initial stage is to rapidly explore the whole CV space, in some cases, it is completed so fast that the sampled data negatively impacts the accuracy of the combined result. For this reason, it is worth comparing the uncertainty of the combined data, including the exploration stage, to the uncertainty of the combined data, excluding the exploration stage. This comparison is illustrated by comparing the bootstrapped error of the FES in Figure 5.14 (a) and the absolute deviation of the FES in Figure 5.14 (a) for both scenarios. It can be observed in either error progression that removing the exploration stage data results in a lower uncertainty. While conducting a simulation campaign, it would be beneficial to make this comparison at the end of the MetaD stage, as excluding the exploration stage data might result in faster convergence.

To further demonstrate the effectiveness of the SRTR approach, 100 campaigns were conducted, and the average of the error progressions was calculated. This is compared with 100 single long simulations with 10^6 simulation steps that employ a MetaD bias, where the parameters were chosen *intuitively* (referred to as "*long-intuitive*"). Moreover, they are compared with



(a) (b)

Figure 5.14: Comparison of the error progression of a *Serial Real Time Reinitialisation* campaign where the exploration stage is included in the combination of the data (blue line) and excluded in the combination of the data (orange line). (a) shows the bootstrap error of the FES and (b) depicts the average absolute deviation of the FES.

100 additional long simulations with the same number of simulation steps, which employ a MetaD bias that was optimised by *trial and error* (referred to as "*long-optimised*"). Additionally, 100 sets of sequential simulations were conducted, consisting of an initial exploration stage simulation with 10^5 simulation steps, and a second simulation in the MetaD stage with $9 \cdot 10^5$ simulation steps (referred to as "*two-step*"). With this approach, it was also found that removing the data collected in the exploration stage resulted in a lower final error. All simulations were started in a random configuration, and the MetaD bias parameters are displayed in Table 5.3.

The error progressions of each approach are reported in Figure 5.15. When comparing the average absolute deviation of the FES, shown in Figure 5.15 (b), it can be observed that the SRTR campaigns have the best convergence from the beginning. For the SRTR and *two-step* approach, the error suddenly increases around 10^5 simulation steps, corresponding to the end of the exploration stage. At this point, the errors are calculated without considering the data collected in the exploration stage. For the SRTR campaigns, this jump is lower because the exploration stage is terminated dynamically at different times across the 100 campaigns. Consequently, the

	MetaD Bias Parameters			
	Height [kJ/mol]	Width	Bias factor	MetaD deposition rate
SRTR	*dynamic* (see example in Table 5.2)			
long-intuitive	5	0.1	50	200
long-optimised	1.67	0.1	16	200
two-step	3.33 and 1.67	0.1	166 and 4.33	200

Table 5.3: MetaD bias parameters for various biasing approaches (first column): Initial height of the Gaussian hills (second column), width of the hills (third column), the bias factor (fourth column), and the MetaD deposition rate measured in simulation steps (sixth column). Each row represents a specific biasing approach: SRTR (second row) denotes *Serial Real-Time Reinitialisation* campaigns, where the biasing parameters are evaluated dynamically, with an example provided in Table 5.2. *long-intuitive* (third row) and *long-optimised* (fourth row) represent single long simulations, where the parameters were chosen intuitively and by trial and error, respectively. *two-step* (fifth row) indicates two sequential simulations, with the first using MetaD parameters that facilitate a fast exploration and the second using more conservative MetaD parameters.

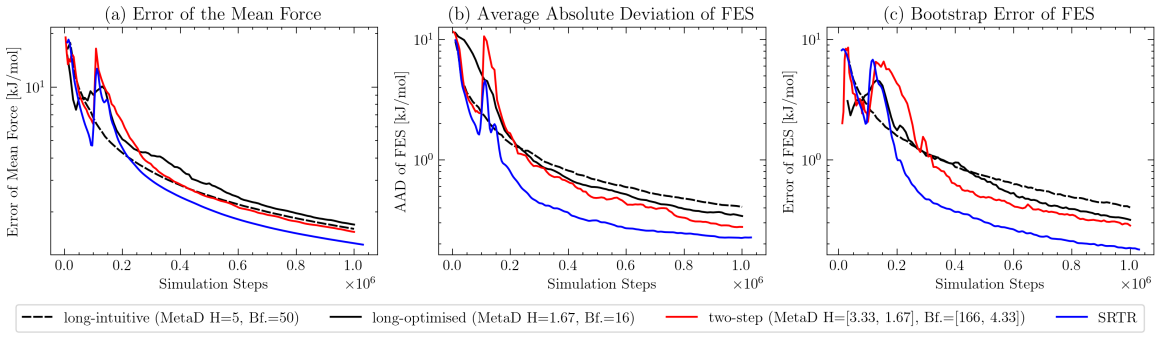


Figure 5.15: Illustration of error progression of long MetaD simulation (black solid and black-dashed line), combined exploration stage plus MetaD stage simulation (red line) and *Serial Real Time Reinitialisation* (SRTR) campaign (blue line). (a) depicts the progression of the average error of the mean force, (b) shows the progression of the average absolute deviation of the FES and (c) illustrates the progression of the average bootstrap error of the FES.

error jump is spread across a larger time range.

The *long-intuitive* approach also converges relatively fast initially, but performs poorly towards the end of the simulation compared to the other

approaches, resulting in the largest final error of 0.407 kJ/mol. Adversely, the error progression of the *long-optimised* simulations initially has a slower convergence, but eventually overtakes the former and finishes with a final error of 0.341 kJ/mol. The reason for this trend is that in the *long-intuitive* approach, the large MetaD height and bias factor enable a rapid initial exploration, which takes longer in the *long-optimised* approach, due to the reduced height and bias factor. Nevertheless, in the long run, the MetaD bias in the former approach continues to grow and change, while the MetaD bias in the latter approach stabilises faster, enabling a better convergence.

The intention in the SRTR and *two-step* approach is to *take the best of both worlds*, aiming for a fast convergence in the exploration stage, followed by a mostly stable bias potential in the long run. As a result, these two methods have the lowest final error, with 0.233 kJ/mol for the SRTR approach 0.277 kJ/mol for the *two-step* approach. This shows that multiple short sequential simulations are not necessarily required and that two simulations alone can result in better convergence than only one. However, in this example, a relatively simple one-dimensional potential surface is simulated that was known *a priori*, allowing for an informed choice of the biasing parameters. When dealing with more complex unknown systems, employing several short simulations would enable an iterative tuning of the biasing parameters as more information about the FES is available.

When comparing the error progression of the average absolute deviation of the FES, shown in Figure 5.15 (b), to the bootstrap error of the FES, illustrated in Figure 5.15 (c), a significant match can be observed, both qualitatively and quantitatively. The final uncertainty in the FES is 0.402 kJ/mol for the *long-intuitive* approach, 0.318 for the *long-optimised* approach, 0.285 kJ/mol for the *two-step* approach and only 0.192 for the SRTR approach. However, the bootstrap error at the start of the simulation is less reliable, as there is not sufficient data to compute it reliably. Further comparison shows that the error of the mean force, depicted in Figure 5.15 (a), measures the

convergence satisfactorily, where the SRTR method has the lowest error and the *two-step* approach has the second lowest error. However, the uncertainty of the *long-intuitive* approach is slightly underestimated, so that the *long-intuitive* approach has the biggest final error of the mean force.

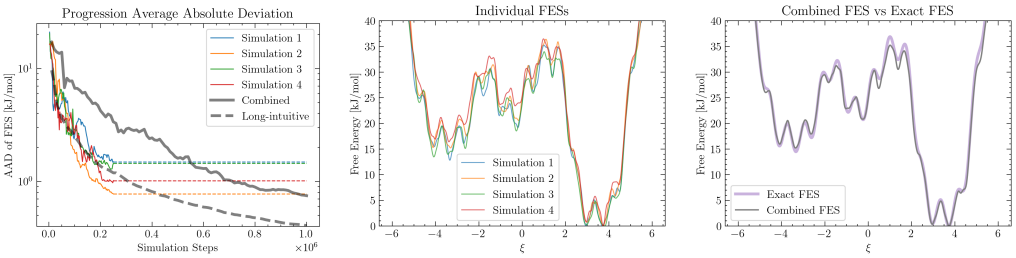
5.6 Parallel Real-Time Reinitialisation

The sequential improvement of biasing parameters, as it is done in SRTR campaigns, can be an effective way to enhance the convergence of biased MD simulations. Moreover, in PRTR campaigns, the SRTR approach is conducted multiple times in parallel, as conveyed in Figure 5.12, where further computational efficiency can be achieved. While such an approach opens the possibility for even more complex biasing strategies, it is essential to share information between simulations in the PRTR approach effectively to avoid poor convergence performance.

5.6.1 Parallel Simulations

Prior to introducing methods to analyse and reinitialise parallel simulations in real-time, parallel simulations that run fully independently from each other are considered. This approach will be referred to as *parallel*, and serves as an illustration of the inefficiencies and for benchmarking later on.

This approach is demonstrated with four *parallel* simulations, initialised in random positions. To enable a fast initial convergence, the biasing parameters of the *long-intuitive* simulation are summarised in Table 5.5. The error progression of each individual simulation is shown in Figure 5.16 (a) along with the error progression produced by combining the data as soon as it was available (i.e. the combined error at time t_{combo} was calculated from the data from each simulation j up to time $t_j = t_{\text{combo}}/4$). Because the individual simulations started in random positions, they converge differently. The first simulation has the highest final error, 1.495 kJ/mol, the second simulation has the lowest final error, 0.776 kJ/mol, and the average final error of the individual simulations is 1.185 kJ/mol. Moreover, the final combined error is 0.7488 kJ/mol, which is notably smaller than the average, but only slightly smaller than that of the second simulation. This is because each simulation spends considerable time exploring the surface, resulting in noisy estimates of the FESs, as can be seen in Figure 5.16 (b). Merging these noisy FESs yields a smoother combined FES, as illustrated in Figure 5.16 (c), but it still contains



(a) (b) (c)

Figure 5.16: (a) Illustration of the progression of the average absolute deviation of the FES. The coloured lines represent the error progression of the individual simulations (simulation 1: blue line; simulation 2: orange line; simulation 3: green line; and simulation 4: red line), and the coloured dashed lines show a horizontal extension of the final errors of the respective simulations. The grey line depicts the combined error progression, which was evaluated by merging the data as soon as it was available. The dashed grey line shows the error progression of *long-intuitive* simulation discussed in Section 5.5 and serves as a reference. (b) FESs of the individual simulations and (c) combined FES (grey line) with exact FES (purple line).

inaccuracies. In contrast, the average error progression of the *long-intuitive* simulations (grey-dashed line in Figure 5.16 (a), same biasing parameters as the parallel simulations), converges at a similar rate compared to the parallel simulations, but notably faster than the combined convergence. A similar trend was also observed in section 3.4, where alanine dipeptide was simulated with twenty short simulations. Here, the question arises whether an alternative approach can be employed, where combining the data from parallel simulations can result in better convergence.

5.6.2 Simple Parallel Simulation Campaign

To improve the combined convergence of several parallel simulations, a PRTR approach is implemented, where the simulation data is analysed as soon as it is available and patched with earlier results. Then, some criteria are used to terminate active simulations and initialise new simulations in their place, using the combined results of all simulations. This approach is initially implemented using the same initialisation and termination criteria as for the SRTR case and will be referred to as *simple PRTR*. To demonstrate

this approach, a campaign using four active simulations at any time was simulated with a time budget of $1\,000\,000$ simulation steps (corresponding to 5 ns) and an F_{lim} of 80 kJ/mol. The minimum simulation steps were set to 10^5 and the maximum to $6 \cdot 10^5$. The simulation stages and their biasing parameters are summarised in Table 5.4.

Stage	Simulation steps	Height [kJ/mol]	Width	Bias factor	MetaD deposition rate
Exploration	50 200	10.0	0.264	200	200
Exploration	67 200	10.0	0.248	200	200
Exploration	88 100	10.0	0.232	200	200
Exploration	110 000	10.0	0.216	200	200
MetaD	119 600	2.0	0.121	6	200
MetaD	131 400	2.0	0.106	6	200
MetaD	147 200	2.0	0.121	6	200
MetaD	110 700	2.0	0.118	6	200
Flat	118 500	0.17	0.122	2	500
Error based sampling	92 100	0.17	0.122	2	500
PD based sampling	68 200	0.17	0.122	2	500
Flat	60 900	0.17	0.122	2	500

Table 5.4: *Simple* PRTR campaign summary: The first row presents the stages in the order they were initialised, together with their simulation steps in the second column. In the remaining columns, the MetaD parameters are described, including the initial height of the Gaussian hills (third column) and their width (fourth column), the bias factor used (fifth column) and the MetaD deposition rate measured in simulation steps (sixth column). Each row represents a simulation conducted in a specific sampling stage.

Figure 5.17 shows the progressions of the average absolute deviation of the exemplar *simple PRTR* campaign. The errors of the individual simulations are shown with coloured lines, which are defined on the lower x-axis, while the combined error of the campaign is depicted with the grey line, which is described by the upper x-axis. The campaign started with four simulations in the exploration phase, which lasted for a total of $316\,500$

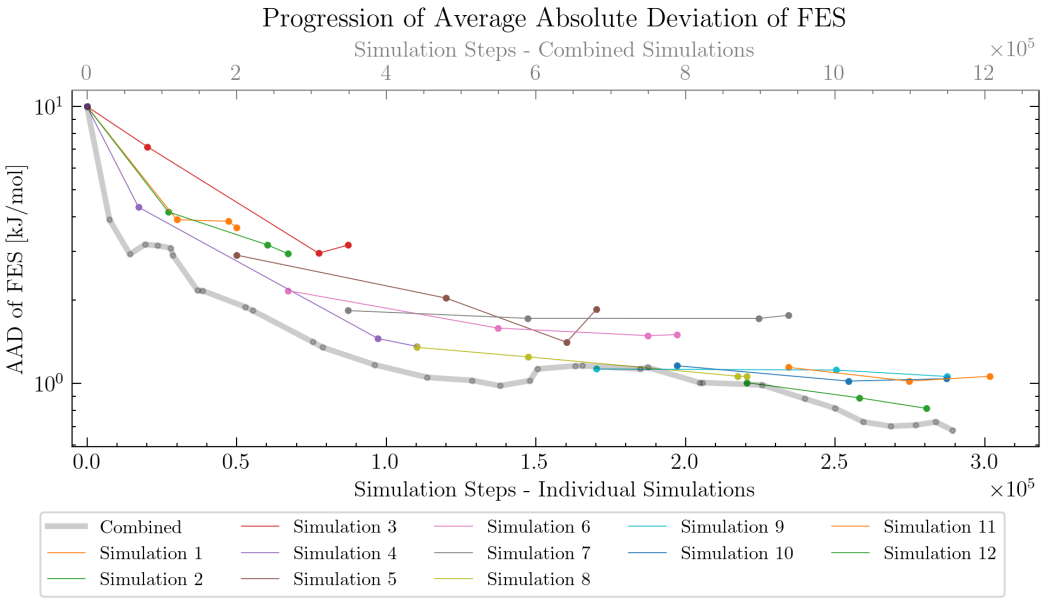


Figure 5.17: Illustration of the progression of the average absolute deviation of the FES of a *simple Parallel Real Time Reinitialisation campaign* (PRTR) employing four parallel simulations at any time. The combined error is represented by the grey line and defined on the top x-axis. The coloured lines depict the error of the individual simulations, which are defined on the bottom x-axis.

simulation steps. As the termination criteria were met in the exploration simulations, they were interrupted one by one, and a simulation in the MetaD stage was initialised in their place. The four simulations in the MetaD stage lasted for a total of $508\,900$ simulation steps, which were replaced by two simulations in the flat stage, one using an error-based InvF bias and one using a probability-density-based InvF bias. The last four simulations only ran for a total of $339\,700$ simulation steps, and finished with a final error of 0.684 kJ/mol using $1\,165\,100$ steps. However, the error after $1\,000\,000$ steps was 0.812 kJ/mol, which is notably worse than that of the single and SRTR simulations (see Section 5.5), and comparable to the *parallel* approach. The main drawback of this *simple* approach is that when a simulation is terminated and a new one is initialised in its place, it is done without the latest sampling from the other simulations. This is particularly detrimental when shifting from the exploration stage to the MetaD stage. With the current implementation, simulations in the exploration stage are terminated

and reinitialised in the MetaD stage without fully analysing all other active simulations. At this point in the campaign, only little sampling is available, and including the data from the other exploratory simulations would improve the InvF bias of the subsequent simulation considerably, and consequently also its convergence. Furthermore, after the first simulation in the exploration phase is terminated and reinitialised with an improved bias, the other simulations are still sampling with the aggressive MetaD bias. However, at this time, the system has already been fully explored, and these simulations could employ a new bias that better contributes to the convergence.

5.6.3 Parallel Real Time Reinitialisation

To improve on the drawbacks of the *simple* PRTR approach, the termination of the exploration stage is synchronised, such that all simulations stop when any of them fulfil the termination criteria (FES + MetaD bias $> F_{\text{lim}}$). Moreover, it was found that a synchronised termination of the MetaD stage is also beneficial. However, when the flat stage is reached (or subsequent stages), the biasing parameters would not change significantly, and a synchronised termination is not necessary. Similar to the SRTR method, it was found that ignoring the data sampled during the exploration stage improves the final convergence. Furthermore, since the exploration stage is removed, a variable number of simulations in the exploration stage is adopted. When a relatively simple potential is simulated, a single simulation is enough. However, more simulations could be advantageous when investigating more complex systems defined on a higher-dimensional CV space, such as the examples in Chapter 4 and 6. It would increase the likelihood of rapidly exploring the CV-space and provide a better InvF bias for the MetaD stage. The final improvement of the PRTR approach concerns the termination criteria after the MetaD stage. Rather than only evaluating the change in error of a single simulation, it is also compared to the overall error. In addition to the existing criteria, the change in error of an individual simulation must be smaller (i.e. more negative) than the change in error of the combined simulations divided

by two:

$$\frac{\Delta\sigma_i(t)}{\Delta t} \leq \frac{1}{2}\frac{\Delta\sigma(t)}{\Delta t} \tag{5.4}$$

If this criterion is not met for five consecutive checks, simulation i is terminated.

To demonstrate this approach, a campaign using four active simulations at any time was simulated with a time budget of $1\,000\,000$ simulation steps (corresponding to 5 ns) and an F_{lim} of 80 kJ/mol. The minimum simulation steps were set to 10^5 and the maximum to $6 \cdot 10^5$ (Python code in Appendix B.3.2). The simulation stages and their biasing parameters are summarised in Table 5.5.

Stage	Simulation steps	Height [kJ/mol]	Width	Bias factor	MetaD deposition rate
Exploration	40 000	5.0	0.24	200	200
MetaD	70 200	2.0	0.139	6	200
MetaD	60 200	2.0	0.139	6	200
MetaD	53 800	2.0	0.139	6	200
MetaD	50 000	2.0	0.139	6	200
Flat	120 200	0.5	0.120	2	500
Error based sampling	230 200	0.5	0.120	2	500
PD based sampling	210 000	0.5	0.120	2	500
Flat	120 200	0.5	0.120	2	500
Error based sampling	110 000	0.5	0.119	2.25	500
PD based sampling	80 000	0.5	0.118	2.25	500

Table 5.5: PRTR campaign summary: The first row presents the stages in the order they were initialised, together with their simulation steps in the second column. In the remaining columns, the MetaD parameters are described, including the initial height of the Gaussian hills (third column) and their width (fourth column), the bias factor used (fifth column) and the MetaD deposition rate measured in simulation steps (sixth column). Each row represents a simulation conducted in a specific sampling stage.

Figure 5.18 shows the progressions of the average absolute deviation of an exemplar PRTR campaign, with four active simulations at any time. The errors of the individual simulations are shown with coloured lines, which are defined on the lower x-axis, while the combined error of the campaign is depicted with the grey line, which is described by the upper x-axis. The campaign started with a single simulation in the exploration phase, which lasted for only $40\,000$ simulation steps. Even though the error at the end of the exploration stage is higher than that of the *simple* PRTR approach, the exploration stage finished much faster, as it only involved analysing and reinitialising one simulation. This allowed the campaign to shift to the MetaD stage faster, where more optimal biasing conditions are employed. In the MetaD stage, the estimate of the FES improved rapidly, resulting in a prompt shift to the flat stage after a combined $234\,200$ steps. At this point, the error is already smaller than the final error of the *simple PRTR* example, using about a quarter of its time budget. In the flat stage (including error-based and probability-density-based InvF biases), the error decreases further and converges to around 0.3 kJ/mol. While two of the simulations initialised at the start of the flat stage run to the end, the other two exhibit an increase in error and are reinitialised with new simulations. The average absolute deviation of the FES at $1\,000\,000$ simulation steps is 0.310 kJ/mol, and the final error after $1\,144\,000$ steps is 0.245 kJ/mol. This is a substantial improvement compared to the *simple* PRTR approach, and is comparable to the single simulation and SRTR approach (see Section 5.5).

Next, the two PRTR approaches are compared with each other, as well as the parallel approach discussed in Section 5.6.1 and the *long-optimised* approach discussed in Section 5.5. Each approach was simulated 100 times with a total of 10^6 simulation steps, and the error progressions are shown in Figure 5.19. Inspecting the progressions of the average absolute deviation of the FES, shown in Figure 5.19 (b), it can be seen that the *parallel* approach has the biggest final error, while the *simple PRTR* approach has a

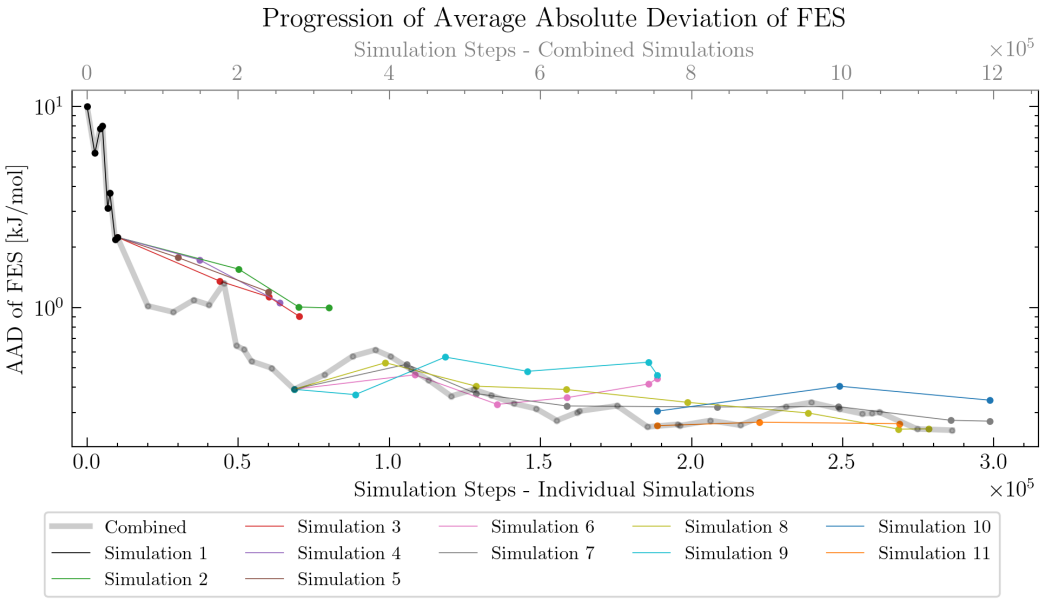


Figure 5.18: Illustration of the progression of the average absolute deviation of the FES of a *Parallel Real Time Reinitialisation campaign* (PRTR) employing one simulation in the exploration stage (black line defined on the top x-axis) and in the remaining stages, four parallel simulations at any time. The combined error is represented by the grey line and defined on the top x-axis. The coloured lines depict the error of the individual simulations, which are defined on the bottom x-axis (excluding the exploration stage).

better initial convergence, but finishes with only a marginally better final error. In contrast, the *long-optimised* approach has a poorer start, but finishes with a significantly lower final error than the previous two approaches. The PRTR approach has the lowest error throughout the entire simulation time.

The progression of the bootstrap error of the FES, depicted in Figure 5.19 (c), matches the trend of the average absolute deviation of the FES qualitatively, however, the error of the *parallel* and the *simple* PRTR approach is underestimated. On the other side, the mean force error progression, illustrated in Figure 5.19 (a), measures the convergence less reliably. The errors of each approach are much closer to each other compared to the average absolute deviation of the FES. Also, the error of the mean force of the *simple* PRTR approach is lower than that of the *long-optimised* approach, even though the opposite is the case.

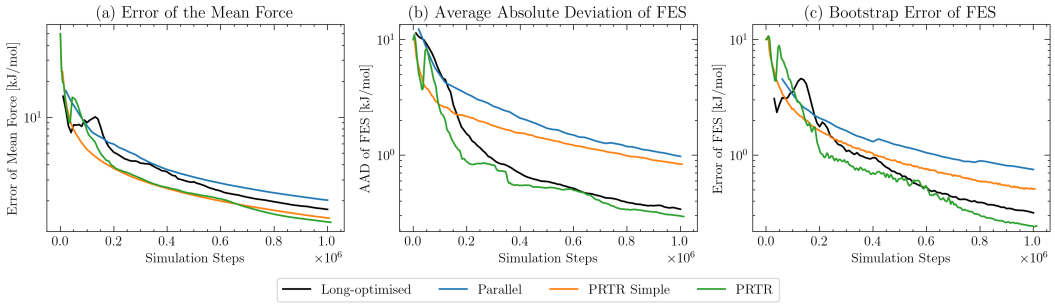


Figure 5.19: Illustration of error progression of long MetaD simulation (black solid and black-dashed line), combined exploration stage plus MetaD stage simulation (red line) and *Serial Real Time Reinitialisation* (SRTR) campaign (blue line). (a) depicts the progression of the average error of the mean force, (b) shows the progression of the average absolute deviation of the FES and (c) illustrates the progression of the average bootstrap error of the FES.

5.7 Testing on Alanine Dipeptide

The effectiveness of the real-time reinitialisation methods introduced in this chapter is further tested on alanine dipeptide by running an SRTR and a PRTR campaign on this system. For information about alanine dipeptide and the simulation software and parameters used, the reader is referred to Section 3.4, where these have been introduced in detail. Moreover, the error progression and the final results of the single long simulation from Section 3.4 will be compared with those from the SRTR and a PRTR campaign. Both campaigns will have a time budget of 20 ns and a guaranteed and maximum simulation time of 1 ns and 5 ns, respectively.

All results are shown in Figure 5.20, where the first row (panels: a - d) shows the final results from the long simulation, the second row (panels: e - f) shows the final results from the SRTR campaign and the third row (panels: i - l) shows the final results of the PRTR campaign. In these first three rows, the final FES is shown in the first column, the biased probability density is depicted in the second column, the *on-the-fly* error of the mean force is illustrated in the third column, and the absolute deviation of the FES is presented in the fourth column. In the last row, the progression of the *on-the-fly* error of the mean force is illustrated in Figure 5.20 (m) and the progression of the average absolute deviation of the FES is depicted in Figure 5.20 (n).

The most notable difference is in the final biased probability density. While the long simulation was sampled longer in lower energy regions, the other two approaches have more uniform sampling. The former is typical of WT-MetaD simulations, whereas in the other two cases, the flat sampling distribution results from repeatedly initialising simulations with an almost flat biased potential surface. In all three cases, the final FESs appear smooth, and the final absolute deviations of the FESs are very similar, with an error of about 2 kJ/mol between the two free energy basins. Moreover, the absolute deviation of the long simulation (Figure 5.20 d) and that of the SRTR cam-

paign (Figure 5.20 h) are almost identical, while that of the PRTR campaign (Figure 5.20 j) is larger in the right basin. By inspecting the progression of the average absolute deviation, depicted in Figure 5.20, it can be confirmed that the long simulation and the SRTR campaign have a very similar accuracy, with a final average error of 0.972 k/mol and 0.967 kJ/mol, respectively. The PRTR campaign has a larger final average error of 1.156 k/mol. Furthermore, the SRTR exhibits the fastest initial convergence, while the PRTR campaign almost matches the long simulation. Conversely, the final error of the mean force (Figure 5.20 m) is the highest for the long simulation, while that of the other two is underestimated. By comparing the error map of the mean force of the long simulation (Figure 5.20 c) with its biased probability density (Figure 5.20 b), it can be observed that regions with a higher uncertainty overlap with higher energy regions that were sampled less. Due to the uniform sampling in the other two cases, the uncertainty in the higher energy regions is smaller, resulting in a lower global uncertainty of the mean force.

The results obtained with the real-time reinitialisation campaigns pose a substantial improvement compared to the previous results presented in Section 3.4. It shows how passing on the information gathered in early simulations to subsequent ones using an InvF bias can improve the convergence, even when multiple simulations are employed in parallel.

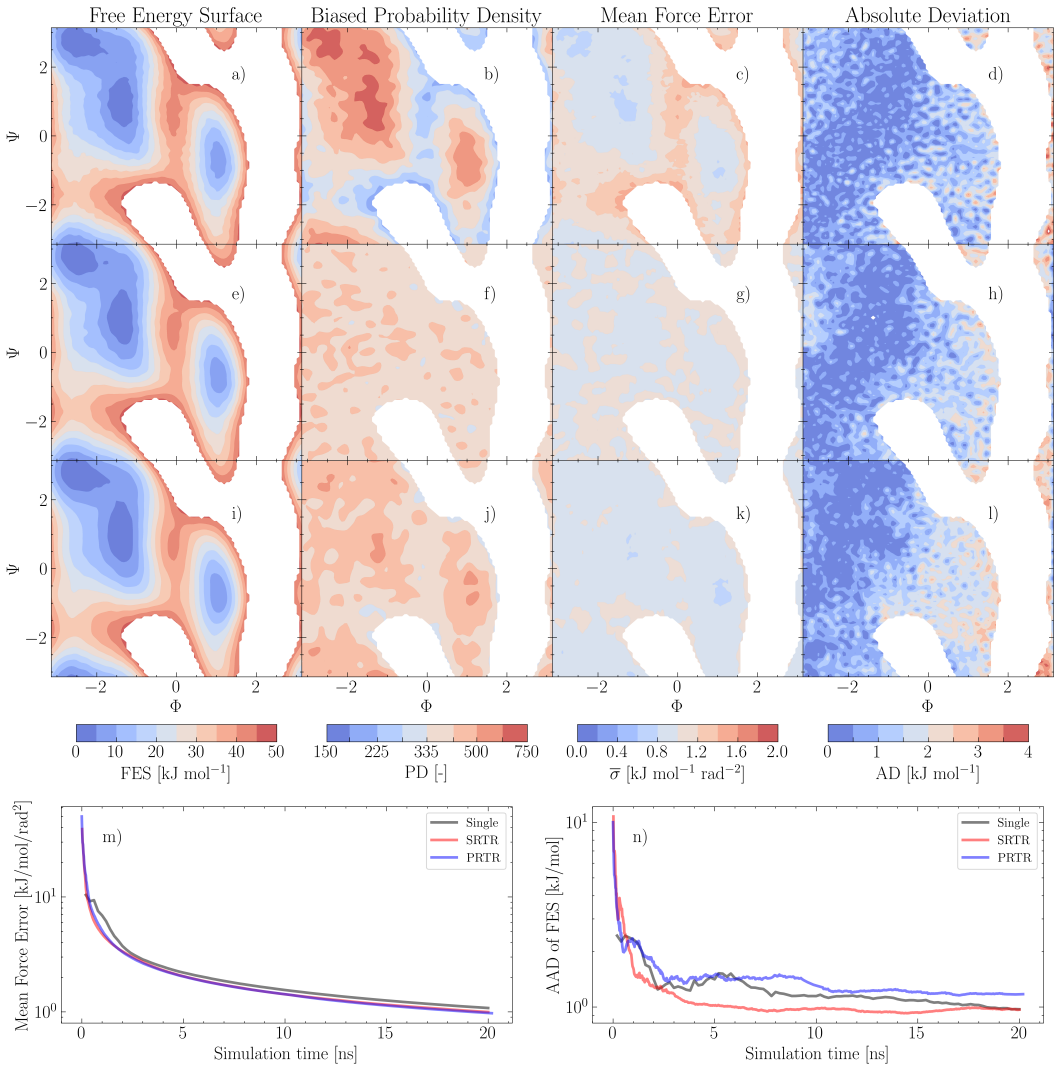


Figure 5.20: Comparison of different simulation strategies. The first row (a-d) shows results from a single 20 ns MetaD simulation. The second row (e-h) shows the results from an SRTR campaign with a time budget of 20 ns. The third row (i - l) shows the results from a PRTR campaign with a time budget of 20 ns. The first column shows the final FES, the second column shows the final biased probability density, the third column shows the error of the mean force, and the fourth column shows the absolute deviation of the FES. (m) Progression of the *on-the-fly* error of the mean force and (n) progression of the average absolute deviation of the FES, where the grey line represents the single simulation, the red line represents the SRTR campaign, and the blue line represents the PRTR campaign.

5.8 Testing on Multiple Surfaces

For a more comprehensive evaluation of the effectiveness of the real-time reinitialisation approaches, they are tested on a range of analytical potentials.

The analytical forms of all one–dimensional test potentials are collected in Appendix A.4, whereas the two–dimensional surfaces are defined in Appendix A.6. The corresponding averaged error progressions are presented in Appendix A.5 for the one–dimensional benchmarks and in Appendix A.7 for the two–dimensional cases.

5.8.1 One-Dimensional Surfaces

First, 20 one-dimensional surfaces are employed, which are depicted in Figure 5.21. These are characterised by varying surface roughness and energy barrier height.

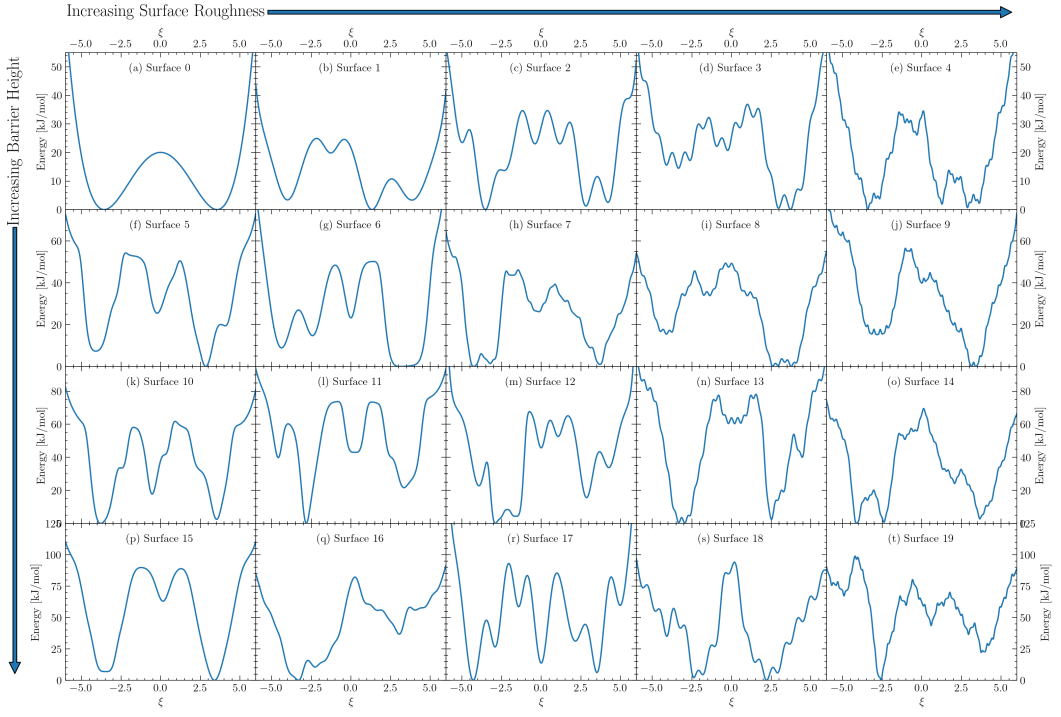


Figure 5.21: The figure shows a range of one-dimensional analytical potential surfaces. Moving from left to right, the surface roughness increases, while moving from top to bottom, the energy barrier height rises.

Each surface was simulated with 100 SRTR campaigns and 100 PRTR

campaigns, each with a time budget of 10^6 simulation steps, a minimum of 10^5 and a maximum of $6 \cdot 10^5$ steps per simulation. For comparison, 100 long WT-MetaD simulations were carried out for each surface, with an initial Gaussian height of 5 kJ/mol, and a bias factor of 50 (identical to the long-intuitive approach discussed in Section 5.5). Additionally, 100 long WT-MetaD simulations were carried out with an adjusted height and bias factor. It was found that an initial Gaussian height of $F_{\text{lim}}/30$ kJ/mol, and a bias factor of $F_{\text{lim}}/3$, where F_{lim} is the largest energy barrier, generally resulted in a good convergence (similar to the long-optimal approach discussed in Section 5.5). All long simulations were simulated with 10^6 simulation steps. Also, each surface was simulated 100 times with a *two-step* approach (discussed in Section 5.5), which employed an exploration stage with 10^5 simulation steps, using an initial Gaussian height of $F_{\text{lim}}/15$ kJ/mol and a bias factor of $3.33 \cdot F_{\text{lim}}$. This was followed by a MetaD stage with $9 \cdot 10^5$ simulation steps employing Gaussian hills characterised by an initial height of $F_{\text{lim}}/30$ kJ/mol and a bias factor of $1 + F_{\text{lim}}/15$. All simulations used a width for the MetaD bias of 0.1, except the SRTR and PRTR campaigns, which evaluated the width based on the FES estimate.

The final errors of each method are summarised in Figure 5.22, where the *on-the-fly* error of the mean force is given in the left table, the average absolute deviation of the FES is provided in the centre table and the average bootstrap error is tabulated on the right side. The tables use the same 4×5 format as Figure 5.21, so that the cell location of each error value corresponds to the cell location of the surfaces depicted in Figure 5.21. For an easier comparison, the cells are coloured based on their respective values. For the error of the mean force, a dark green colour is used for values below 1.3 kJ/mol, a dark red colour for values above 2.5 kJ/mol, and values between 1.3 and 2.5 kJ/mol shift progressively from green to yellow and red as they increase. Similarly, for the error of the FES (centre and right table), a dark green colour is used for values below 0.15 kJ/mol, a dark red colour for

values above 1.3 kJ/mol, while values in between shift linearly from green to red. The first row in Figure 5.21 shows the final error from the long simulation with the constant MetaD parameters, the second row shows the results from the long simulation with the adjusted MetaD parameters, and the third row shows the results from the *two-step* approach. The fourth row shows the results from the SRTR method, and the last row shows the results from the PRTR method. The general trend for all error types is that the surfaces with a lower energy barrier and surface roughness are predicted with better accuracy. For all methods, either the first or the second surface (smooth surface with small energy barrier) is evaluated with the highest precision, and the last surface (very rough surface with large energy barrier) is predicted with the most significant error.

Moreover, by taking the mean of all final average absolute deviations of the FES, it is found that the *long-intuitive* approach performs the worst, with an average final error of 0.622 kJ/mol, and the *long-optimised* approach is marginally better with 0.615 kJ/mol. The PRTR approach has a slightly lower average final error of 0.592 kJ/mol, while the SRTR approach is notably more accurate with 0.430 kJ/mol. Nevertheless, considering the parallel efficiency in the PRTR approach, it could be an effective alternative when simulating computationally demanding systems. Overall, the *two-step approach* estimated the FESs most precisely with an average final error of 0.403 kJ/mol. Furthermore, the final error difference between the two *long* approaches and the *reinitialisation* approaches is more pronounced for surfaces with higher energy barriers, highlighting the benefit of starting with an exploratory stage and moving faster to a conservative MetaD bias. The fact that the *two-step* approach performed the best suggests that it is not necessary to employ several consecutive simulations, and that a fast exploratory simulation followed by a conservative MetaD simulation is an effective strategy.

When comparing the average absolute deviation of the FES with the

Figure 5.22: Comparison of final error for the simulation of different surfaces (see Figure 5.21) with different simulation strategies. Long-intuitive (first row), long-optimised (second row), two-step (third row), SRTR (fourth row), and PRTR (fifth row). Average error of the mean force (first column), average absolute deviation of FES (second column), and average bootstrap error of FES (third column).

bootstrap error of the FES, the values align qualitatively. However, the bootstrap error is systematically underestimated by around 15%. The only exception is the bootstrap error of the PRTR approach, which is underestimated by around 50%. Likewise, the error of the mean force of the PRTR approach is notably underestimated. Overall, the PRTR has the lowest final error of the mean force compared to the other methods, even though the actual error is larger. In contrast, the error of the mean force of the other approaches qualitatively reflects the trend observed in the error of the FES.

5.8.2 Two-Dimensional Surfaces

Lastly, the SRTR and PRTR approaches are tested on four analytical two-dimensional surfaces of increasing complexity, depicted in Figure 5.23. The potential displayed in Figure 5.23 (a) was first proposed by Invernizzi *et al.* [76] and is, therefore, referred to as the *Invernizzi potential*. The other surfaces, illustrated in Figure 5.23 (a), (b) and (c), were created for this work, and are referred to as Potential 1, 2 and 3, respectively. The *Invernizzi potential* is comparatively the smoothest surface with an $F_{\text{lim}} = 20$, and it was simulated with a time budget of 10^6 simulation steps. Potential 1 is also smooth, but with an $F_{\text{lim}} = 40$, so it was simulated with a time budget of $1.5 \cdot 10^6$ simulation steps. Potentials 2 and 3 also have an $F_{\text{lim}} = 40$, but are more challenging to estimate correctly, which is why the time budget for these surfaces was set to $3 \cdot 10^6$ simulation steps.

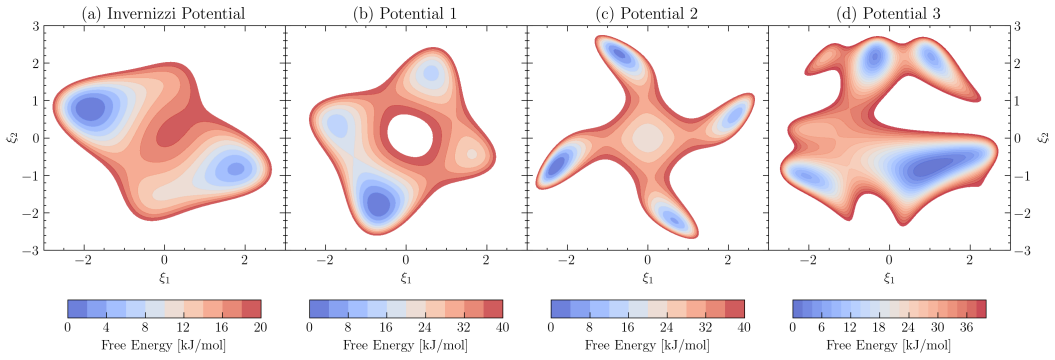


Figure 5.23: Four two-dimensional analytical potential surfaces, increasing in complexity from left to right.

The real-time reinitialisation approaches are each simulated 100 times for each surface. Additionally, each surface was simulated 100 times with single, long simulations, employing a WT-MetaD bias with an initial Gaussian height of $F_{\text{lim}}/20$ and a bias factor of $F_{\text{lim}}/2$. For the *Invernizzi potential*, these MetaD parameters, together with a Gaussian width of 0.186 for both CVs, correspond to the MetaD parameters reported by Invernizzi *et al.* in their publication [76]. For the other surfaces, a Gaussian width of 0.1 was used for both CVs. Furthermore, each surface was simulated 100 times using

a two-step approach (see Section 5.5). With this approach, the exploration simulation used an initial Gaussian height of $F_{\text{lim}}/4$ and a bias factor of $5{\cdot}F_{\text{lim}}$, and the second simulation was characterised by an initial Gaussian height of $F_{\text{lim}}/20$ (same as long simulation) and a biasfactor of $F_{\text{lim}}/2$. The Gaussian width was also set to 0.186 for both CVs for the *Invernizzi potential*, and 0.1 for both CVs for the other potentials. For both the long simulation and the *two-step* approach, the total simulation steps are identical to the time budget of the real-time reinitialisation approaches, with the exploration simulation in the *two-step* approach having 10% of the total simulation steps.

The final errors of each method are summarised in Figure 5.24, where the *on-the-fly* error of the mean force is given in the left table, the average absolute deviation of the FES is provided in the centre table and the average bootstrap error is tabulated on the right side. Similar to Figure 5.22, the cells are coloured based on their respective values for an easier comparison. Contrary to Figure 5.22, the colour scale for the average absolute deviation of the FES (Figure 5.24 b) is different from that of the average bootstrap error of the FES (Figure 5.24 c), as it is systematically underestimated by 50% to 75%.

When inspecting the average absolute deviation of the FES closer, it is apparent that the *Invernizzi potential* was predicted most accurately with an average error of 0.444 kJ/mol, while potential 2 was hardest to predict with an average error of 2.292 kJ/mol. Furthermore, the *two-step* approach was the most effective method, with an average error of 0.698 kJ/mol. The long simulations and the SRTR method performed similarly well, with an average error of 1.082 kJ/mol and 1.099 kJ/mol, respectively, whereas the PRTR method was the least accurate with an average final error of 1.646 kJ/mol.

As stated above, the average bootstrap error of the FES is underestimated systematically, however, it correctly establishes the two-step approach as the most precise simulation strategy. Even so, the long simu-

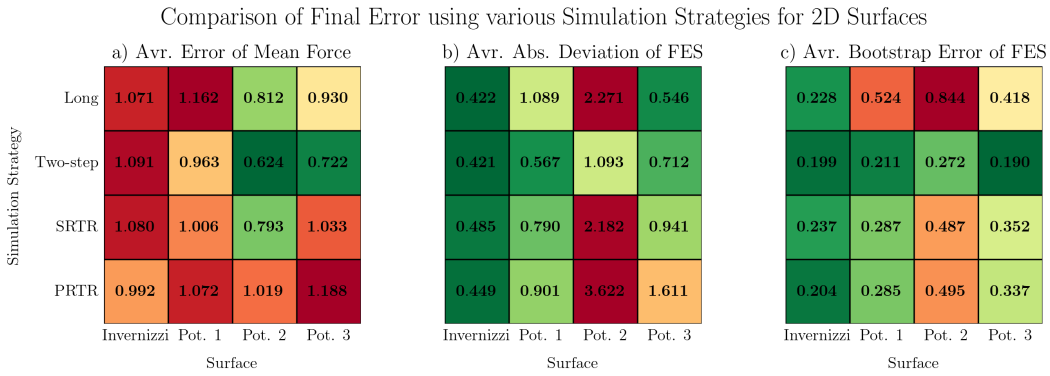


Figure 5.24: Comparison of final error for the simulation of different two-dimensional surfaces. a) Average error of the mean force, (b) average absolute deviation of FES, and (c) average bootstrap error of FES. Each error is given in a table where the columns indicate the surface (see Figure 5.23) and the rows provide the simulation strategy used to simulate the surface. The simulation strategies are: Long (first row), two-step (second row), SRTR (third row), and PRTR (fourth row). The cells of the tables are coloured, where the smallest value of each table is dark red, the largest value of each cell is dark green, and values in between gradually shift from green to yellow and red.

lations are wrongly determined to be the least accurate, and the SRTR and PRTR approaches have roughly the same bootstrap error, which also does not reflect the real precision of the methods. The error of the mean force is also not an adequate representation of the real error. Even though it correctly determines the two-step approach to be more accurate and the PRTR approach the least accurate, the error of the *Invernizzi potential* is the highest, and that of potential 2 is the lowest, which is the opposite of the true error.

Overall, results resonate with the results obtained from the comparison with the one-dimensional potentials, where the *two-step* approach was identified as the most effective method. It emerges as a straightforward technique to enhance the convergence of MetaD simulations by effectively shortening the exploration of the system and moving faster to a less fluctuating MetaD bias.

5.9 Conclusions

The results presented in this chapter demonstrate the practicality and versatility of the *inverse-FES* (InvF) bias as a means for transferring information from existing simulations to new ones. Inverting the most recent FES estimate and employing it as a static bias allows the new simulation to access higher-energy states directly and circumvents the need to construct a MetaD bias from scratch. The InvF bias can be supplemented with an additional MetaD bias and arbitrary surfaces tailored to user needs. This enables bias potentials that facilitate the sampling of less converged regions or some other target distribution. It was demonstrated how this strategy could be used to create a canal along an approximate transition pathway, and proved effective for restricting the trajectory of an alanine dipeptide simulation to the vicinity of that transition path.

A novel simulation strategy was introduced, where one long simulation is replaced by a campaign of several shorter simulations. The campaign is divided into various stages, beginning with an exploration stage that uses an aggressive MetaD bias to explore the whole CV space rapidly. Subsequent stages use progressively smoother MetaD biases in combination with increasingly accurate InvF potentials derived from the evolving FES estimate.

This concept was formalised in the Serial Real-Time Reinitialisation (SRTR) approach, in which each short simulation is terminated and replaced as the campaign moves through its predefined stages. Automated criteria were devised for the initialisation and termination of every simulation, allowing the bias parameters to improve incrementally as convergence advances. An alternative Parallel Real-Time Reinitialisation (PRTR) approach was devised for parallel computing environments. In this approach, termination and reinitialisation decisions were made similarly to the SRTR approach; however, the combined information from all simulations was used, and synchronised actions were taken.

Benchmark calculations against conventional single WT-MetaD con-

firmed the advantages of reinitialisation. For one-dimensional surfaces, SRTR campaigns consistently converged faster than single simulations, while PRTR campaigns delivered similar accuracy with superior parallel scalability. However, the advantage of SRTR and PRTR campaigns was less pronounced for two-dimensional surfaces. Nevertheless, across all examples, a *two-step* strategy emerged as particularly effective and reliable. This approach employed an initial brief exploratory simulation, followed by a longer simulation with a gentler MetaD bias. This highlighted the effectiveness of starting with a short and rough exploration stage, and moving faster to a stage with a less fluctuating MetaD bias where the system converges faster. Furthermore, it suggested that using an overly complex biasing strategy, as was used in SRTR, might not always be advantageous, rendering the optimal balance between straightforward protocols (*two-step*) and more elaborate adaptive schemes (SRTR) an endeavour for future work.

The bootstrap error of the FES served as a reliable convergence metric across all one-dimensional test cases. However, in two dimensions, its robustness decreased, underscoring the need for improved error quantification. Moreover, the mean-force error tended to overestimate the accuracy of reinitialised simulations, in particular that of PRTR campaigns. Finally, applications to complex chemical systems will be essential to validate the reinitialisation approaches presented in this section.

The calculations reported in this chapter were performed with the MFI Python library, which is openly available on GitHub. The class that handles one-dimensional surfaces is provided at `https://github.com/mme-ucl/MFI/tree/master/MFI_class1D`, and the corresponding two-dimensional implementation at `https://github.com/mme-ucl/MFI/tree/master/MFI_class2D`. Both repositories include illustrative Jupyter notebooks that demonstrate how the classes can be used to reinitialise simulations manually or to conduct complete SRTR and PRTR campaigns.

Chapter 6

Solvent Effects on Beta-Scission via Mean Force Integration

In this chapter, MFI is employed to analyse and combine multiple short MD simulations of radical polymerisation reactions. The study was conducted in collaboration with Francesco Serse (from the Department of Chemistry, Materials, and Chemical Engineering of the Politecnico di Milano), who spent several months at University College London as a visiting PhD student. Serse designed and executed the simulations, including preparing the force fields and the simulation environment. The contributions of the writer consist of assisting in the post-processing of the data with Mean Force Integration (MFI) and optimising biasing parameters. The contents of this chapter were published in the article "Unveiling solvent effects on β-scissions through metadynamics and mean force integration" in the *Journal of Chemical Theory and Computation* (https://doi.org/10.1021/acs.jctc.4c00383) [5], with Serse being the first author of the article and the writer of this thesis the second author.

6.1 Introduction

Free radical polymerisation (FRP) and pyrolysis are central to numerous industrial applications for synthesising and recycling polymeric materials, which are widely utilised in plastics, coatings, and adhesives. Central to un-

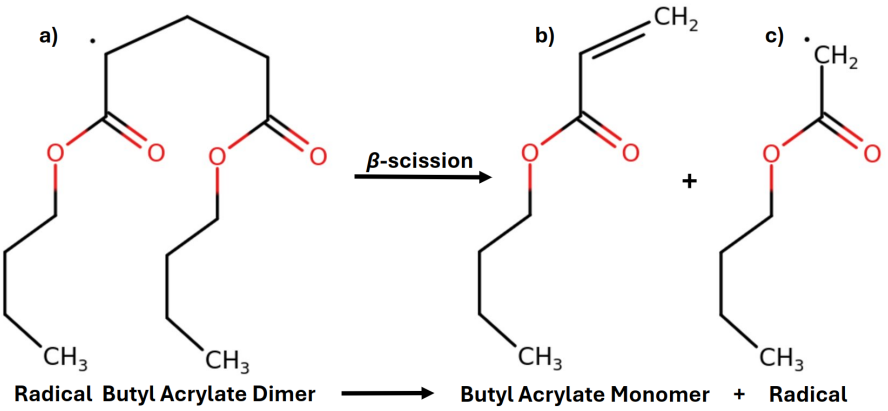


Figure 6.1: Beta-scission of butyl acrylate: a) Radical butyl acrylate dimer, b) butyl acrylate monomer, and c) radical product.

derstanding and optimising these processes is the accurate determination of kinetic parameters for elementary reactions that significantly influence structural properties such as molecular weight distributions.

In FRP, monomers such as butyl acrylate (BA) undergo chain reactions mediated by free-radical intermediates, yielding large polymers, such as poly-butyl acrylate (PBA). This process is started by radical initiators, typically peroxides, which couple with monomers, generating end-chain radicals (ECR). These ECRs propagate the FRP through successive reactions with other monomers or terminate when two radicals combine. At temperatures above 400 K, intramolecular hydrogen abstraction in radical polymers (backbiting) becomes increasingly favourable, creating mid-chain radicals (MCR). Subsequently, the MCR may combine with another monomer to form a branched chain, or fragment via β-scission into a shorter ECR and a monomer, as illustrated in Figure 6.1. In contrast, pyrolysis is a thermal degradation process, where large polymer chains are decomposed into oligomers and monomers. Due to the higher operating temperatures ($T > 600$ K), backbiting and β-scission play an essential role in pyrolysis.

Experimental measurement of kinetic rate constants of FRP and pyrolysis processes often bears substantial challenges. Techniques such as pulsed laser polymerisation combined with size exclusion chromatography (PLP-SEC) [126, 127, 128] or semi-batch reactor studies with nuclear mag-

netic resonance analysis (NMR) [129], are restricted to temperatures below 410 K due to limited operating capabilities. Thus, experimental kinetic rate constants associated with backbiting and β-scission at elevated temperatures are scarce [129, 128, 130], and are commonly estimated empirically [131, 132, 133].

In this study, PBA was selected as a model system because numerous experimental studies [126, 127, 128, 129, 130, 134] provided a reliable benchmark for validation. In particular, the β-scission of PBA is investigated through molecular simulations at temperatures where experimental reaction rates are available ($T = 310$ K–410 K), and at an elevated temperature lacking experimental data ($T = 510$ K). Furthermore, the reaction is examined in various solvent environments, including water, xylene, BA monomer and vacuum.

Accurately describing the reactive intermediates requires computationally demanding quantum-mechanical calculations. To reduce the computational cost, BA dimers (depicted in Figure 6.1) were employed to investigate the β-scission, instead of much longer BA polymer chains. This decision was based on the assumption that the reactivity is primarily influenced by local structure rather than by chain length [135, 136, 137].

Even so, the computational demand was so great that it was not possible to sample sufficient crossings of the energy barrier within a single run to calculate a converged FES. For this reason, it was decided to adopt an alternative strategy where the simulations are divided into multiple independent, shorter runs. This allowed, on the one hand, for the parallel execution of independent simulations, thereby gaining computational efficiency. On the other hand, it enabled a more flexible scheduling of simulations, such that these could be run when computational resource allocations were available.

The independent trajectories were post-processed and merged with MFI to determine the combined FES. Lastly, generalised transition state theory was used to estimate reaction rates [138], which are validated against

available experimental data. While this chapter exclusively investigates β-scission, Serse *et al.* published a complementary analysis of the reaction kinetics of backbiting [139], employing analogous analysis tools as presented here.

6.2 Methodology

6.2.1 Computational Approach

Accurate computational predictions of reaction kinetics in condensed phases demand advanced methodologies capable of describing electronic structures explicitly. To this end, both quantum mechanics (QM) and molecular mechanics (MM) have been employed in a hybrid computational framework (QM/MM).

The QM/MM approach partitions the system into two distinct regions, significantly reducing computational effort while preserving accuracy where required. The radical BA dimer undergoing β-scission is treated using QM methods, while solvent molecules (water, xylene, or BA monomer) are described using classical molecular mechanics force fields. Explicit solvent molecules are included, capturing specific solvent-solute interactions crucial for accurately predicting kinetic parameters.

6.2.2 Simulation Details

All simulations were conducted by Serse [5] using CP2K version 9.1 [140] in a periodic box with dimensions of 23 Å $\times 23$ Å $\times 23$ Å and a time step of 0.5 fs for the MD simulations. The equilibration stage comprised an initial 100 ps simulation in the isothermal-isobaric ensemble to reach equilibrium bulk density, followed by an additional 50 ps equilibration in the canonical ensemble. The Nosé-Hoover thermostat was used in the equilibration simulations [141], with a time constant of 50 fs, while the Bussi-Parrinello thermostat [107] was used for the other simulations. The generalised Amber force field (GAFF) [68, 142] was utilised in the equilibration runs and in the production runs to describe the behaviour of the solvent molecules. Long-range electrostatic interactions were determined using the smooth particle mesh Ewald method [143, 48] with a 10 Å cutoff.

For MetaD simulations, the reactive dimer constituted the QM region. It was described by the GFN1-xTB (Geometry, Frequency, and Noncovalent

interaction force field — extended tight binding Hamiltonian) method [144], chosen due to its balance of computational efficiency and reasonable accuracy. Electrostatic interactions between QM and MM regions were described using a Coulomb potential [145] and a 1 ps equilibration was conducted to allow the QM/MM partition to stabilise.

The β-scission reaction was described with two CVs, depicted in Figure 6.2. The primary CV measures the carbon-carbon bond distance, d_{C-C}, undergoing scission (the distance between the carbon atoms labelled with 2 and 3 in Figure 6.2). The second CV provides the torsional dihedral angle, ϕ, between the molecular fragments involved (the dihedral angle involving the carbon atoms labelled with 1, 2, 3 and 4 in Figure 6.2). The bias potential was constructed using Gaussian hills with a constant height of 1 kcal/mol and widths of 0.2 Bohr and 0.2 rad, deposited every 30 fs. A non-tempered MetaD bias was chosen so that the height of the Gaussians would not be damped and the configuration space would be explored faster. Using non-tempered MetaD comes with the disadvantage that the bias grows without bounds, resulting in poorer convergence over long simulation times. However, the relatively short simulation times that were accessible rendered this drawback negligible. Post-processing non-tempered MetaD simulations and evaluating the FES with MFI poses no additional complications compared to post-processing WT-MetaD simulations. The system was simulated multiple times for 25 ps at different temperatures, using butyl acrylate, water and xylene as solvents, and vacuum (imitating gas phase conditions). The number of simulations for each temperature and solvent environment is summarised in Table 6.1.

6.2.3 Evaluation of Free Energy Barriers

The individual simulation data were analysed and combined with MFI to construct the FES. To simplify the evaluation of the free energy barrier, the two-dimensional FES was projected onto a one-dimensional representation [146] along the primary reaction coordinate d_{C-C} through:

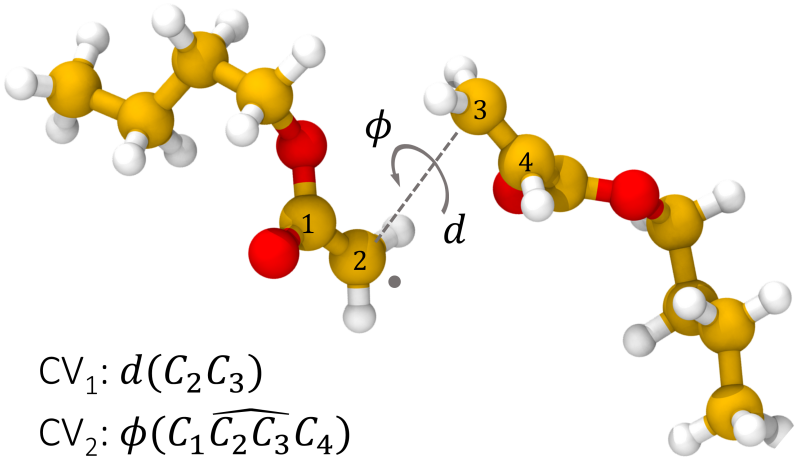


Figure 6.2: Image adapted from Serse *et al.* [5]. Collective variables used to describe the β-scission. The distance between the carbon atoms labelled with 2 and 3 is denoted by d, and the torsional dihedral angle between the molecular fragments is denoted by ϕ

Solvent	Temperature [K]	Number of Simulations
Gas (Vacuum)	310	10
	410	9
	510	10
Butyl Acrylate Monomer	410	11
Water	310	8
	410	8
	510	8
Xylene	410	10

Table 6.1: Summary of solvents used, temperature conditions and number of simulations.

$$F(d_{C-C}) = -\frac{1}{\beta} \ln \left(\int P^u(d_{C-C}, \phi) \, \mathrm{d}\phi \right), \tag{6.1}$$

where $P^u(d_{C-C}, \phi)$ is the unbiased probability density, which is obtained from the two-dimensional FES as:

$$P^u(d_{C-C}, \phi) = \frac{e^{-\beta F(d_{C-C}, \phi)}}{\int \int e^{-\beta F(d_{C-C}, \phi)} \mathrm{d}\phi \, \mathrm{d}d_{C-C}} \tag{6.2}$$

The energy barrier can be categorised as the *activation* energy barrier, indicated with a * superscript (e.g. ΔF^*), which is the energy difference between the reactant and the transition state, and the reaction energy barrier, indicated with a R superscript (e.g. ΔF^R), which is the energy difference between the reactant and the product state. Once the one-dimensional FES were available at various temperatures, the internal energy contribution, $U(d_{C-C})$, and the entropic contribution, $S(d_{C-C})$, were separated through equation 2.41, by fitting the FESs at different temperatures as follows:

$$F(d_{C-C}, T_i) = U(d_{C-C}) - T_i S(d_{C-C}), \qquad T_i \in \{310\mathrm{K}, 410\mathrm{K}, 510\mathrm{K}\} \tag{6.3}$$

The fitting was done using the Scikit-learn linear regression module [147].

Once the internal energy contribution was isolated, it was refined via higher-level density functional theory (DFT) optimisations followed by zero-point-energy (ZPE) corrections in vacuum at 0 K for five configurations, including one representing the reactant state and one the transition state. For each configuration, higher-level ωB97XD/def2-TZVPP DFT [148] calculations were conducted to optimise their geometries, followed by a harmonic frequency analysis to compute the ZPE corrections. The energy corrections were then applied uniformly across solvent conditions to correct the internal energy activation barrier as follows:

$$\Delta U^*_{\omega\mathrm{B97XD}} = (\mathrm{E}^* + \mathrm{ZPE}^*) - (\mathrm{E}^R + \mathrm{ZPE}^R) \tag{6.4}$$

where the * superscript denotes values calculated in the transition state, the R superscript denotes values computed in the reactant state, and E represents internal energies calculated with the QM scheme used during the simulations. Finally, the energy difference was used to replace the previous internal energy barrier, ΔU_{xTB}, to compute the corrected free energy difference of activation:

$$\Delta F^*_{\text{corrected}} = \Delta F^*_{\text{xTB}} + (\Delta U^*_{\omega\text{B97XD}} - \Delta U^*_{\text{xTB}}) \tag{6.5}$$

6.2.4 Histogram Test

A histogram test can be conducted to validate the dividing surface identified on the one-dimensional FES and the choice of CV (d_{C-C}) [48, 149]. This is accomplished by initialising unbiased MD simulations at the identified transition state, using various torsional angles (ϕ), and monitoring whether they evolve toward reactant or product states. The committor probability distribution obtained from these trajectories serves as a diagnostic. If it contains a sharp peak around 0.5, it can be confirmed that the chosen CV is suitable for describing the transition state accurately. If the committor probability distribution significantly deviates from a monomodal distribution centred at 0.5, the one-dimensional CV is not enough to accurately represent the energy barrier of activation.

6.2.5 Evaluation of Reaction Rates

Kinetic rate constants were calculated within the framework of generalised transition state theory (TST) [138, 150], where the transition state divides the phase space into a region for reactants (R) and one for products (P). The rate constant, k_β, is determined as the net flux between these regions using the expression:

$$k_\beta = \frac{\langle |\dot{d}_{C-C}| \rangle^*}{2l} e^{-\beta \Delta F^*} \ , \tag{6.6}$$

where ΔF^* denotes the height of the energy barrier separating the region R from P (or free energy barrier of activation), and $\langle|\dot{d}_{C-C}|\rangle^*$ denotes the ensemble average velocity of crossing the energy barrier. The expression in equation 6.6 is normalised by l, representing the reaction coordinate length that connects the reactant state to the transition state. For a more detailed explanation of the TST theory, readers are referred to [138, 150].

6.3 Results and Validation

6.3.1 Free Energy Surface Exploration

The independent trajectories, simulated at 410 K in different solvent environments (summarised in Table 6.1), were post-processed with MFI and the resulting (two-dimensional) FESs are illustrated in Figures 6.3 (a-d). All FESs share a basin elongated along $d_{C-C} \approx 3$, an energy barrier at approximately $d_{C-C} \approx 5$, followed by a roughly flat free energy landscape beyond $d_{C-C} > 6$. On the other hand, the FES only contains minor fluctuations for varying values of ϕ along constant values of d_{C-C}. While the energy barrier along d_{C-C} is around 30 kcal/mol, the energy fluctuations along ϕ are below 3 kcal/mol. This implies a rapid equilibration of transitions in the torsional angle, whereas breaking the carbon-carbon bond emerges as the rate-limiting mechanism. Because the FESs were reconstructed with MFI, local uncertainties and convergence behaviour can be evaluated quantitatively. The local bootstrap error of the combined FESs, depicted in Figures 6.3 (e-h), is below 1 kcal/mol within the basins for every solvent condition. Along the energy barrier and for higher values of the $C-C$ distance, the error is larger, but remains below 2 kcal/mol. Part of this increase stems from the convention of setting the lowest free energy of the basin to 0 kcal/mol, resulting in a lower error in that region. Conversely, the height of the energy barrier is determined from the slope of the FES, where errors can be introduced. Besides, the basin was sampled more extensively than other regions, resulting in a lower error in the basin region.

The temporal evolution of the error of the mean force (black line) and that of the bootstrap error of the FES (red line) are displayed in Figures 6.3 (i-l) with the green vertical lines indicating the end of the individual simulations. The bootstrap error stabilises at around 0.5 kcal/mol for all solvent cases, suggesting that the existing simulation data is sufficient to estimate the FES accurately. Moreover, the final error of the mean force is 1.2 kcal/(mol·Bohr·rad) for all cases. When comparing the mean force er-

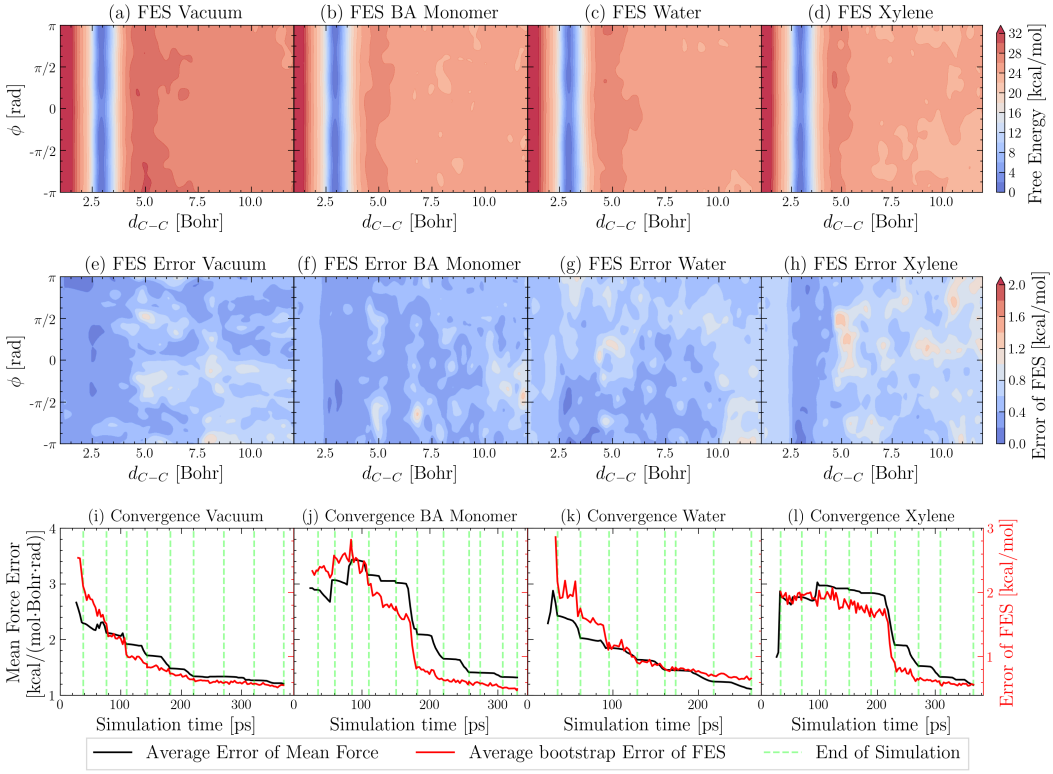


Figure 6.3: (a-d) Two-dimensional FES as a function of d_{C-C} and ϕ from MetaD simulation at 410 K, analysed with MFI, and (e-h) the bootstrap error of the FES. (i-l) Progression of the mean force error (black line, left y-axis), bootstrap error of the FES (red line, right y-axis) with the end of the individual simulations (green-dashed vertical lines). The butyl acrylate dimer was simulated in (a,e,i) vacuum, (b,f,j) xylene, (c,g,k) water, and (d,h,l) BA monomer.

ror with the bootstrap error of the FES, it can be observed that the latter appears to have converged, whereas the former has not reached convergence yet. However, this is expected since residual fluctuations of the mean force are smoothed through integration, and their impact on the FES is attenuated even while the mean force is still refining. These insights into the convergence of the FES highlight the key advantages of MFI. Not only can independent biased trajectories be merged, but the method also provides position-dependent uncertainties and convergence metrics, thereby providing quantitative confidence in the reliability of the final results.

This insight motivates the projection of the two-dimensional FES onto only the d_{C-C} dimension by integrating out ϕ (see discussion in Section

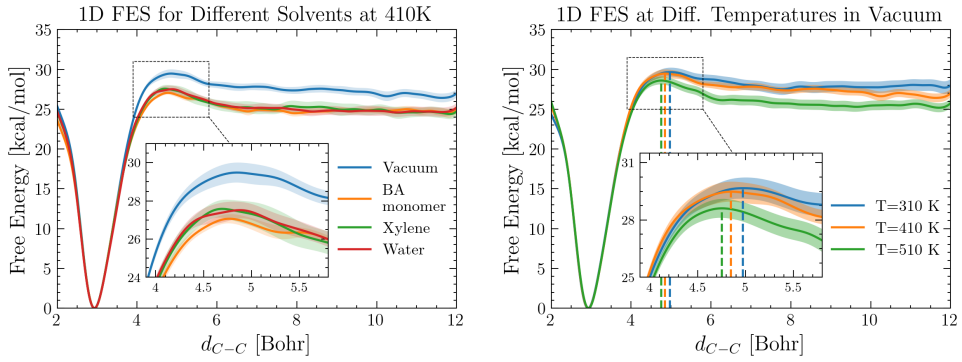
6.2.3). The marginal free energy profiles derived from these landscapes are depicted in Figure 6.4 (a), enabling a more precise assessment of the energy barrier. It is the largest when the β-scission was simulated in vacuum, with an energy barrier around 29.5 kcal/mol. The energy barrier from the simulations that used xylene and water as solvents is 2 kcal/mol lower, while that of the simulations that employed BA monomers is 2.5 kcal/mol lower. These results demonstrate a notable reduction in activation barriers due to solvent interactions. Figure 6.4 (b) shows the one-dimensional FES from the simulations of the β-scission in vacuum at 310 K, 410 K and 510 K. It can be observed that as the temperature increases, the height of the energy barrier decreases and shifts toward the reactant state. This suggests that β-scission becomes more thermodynamically favoured with increasing temperature, as reported in previous studies [131, 132, 133]. Moreover, the uncertainty in the one-dimensional FESs is also displayed in Figure 6.4 with the coloured shaded regions. These provide confidence in the accuracy of the resulting surfaces and can also be used to determine the uncertainty of the reaction rates.

6.3.2 Histogram Test

A histogram test was conducted and the committor probability is shown in Figure 6.5. It confirms the suitability of the chosen reaction coordinate (d_{C-C}) for capturing the transition state. Unbiased MD trajectories initiated at the dividing surface consistently produce a sharply peaked committor probability distribution around 0.5, verifying that no additional collective variables are required to describe the transition state accurately.

6.3.3 Determining the Energy Barrier

The FESs were decoupled to extract the internal energy and the entropy contribution through linear regression to gain a deeper thermodynamic understanding. Figure 6.6 (a) shows the contributions of the FES simulated at 510 K in vacuum. The internal energy profile starts at zero kcal/mol in the



(a) (b)

Figure 6.4: One-dimensional free energy profile of the β-scission of butyl acrylate as a function of d_{C-C} together with the bootstrap error indicated by the shaded regions. (a) FES of butyl acrylate β-scission simulated at 410 K in vacuum (blue line) and different solvents: water (red line), xylene (green line) and BA monomer (orange line). (b) FES of butyl acrylate β-scission simulated in vacuum at 310 K (blue line), 410 K (orange line) and 510 K (green line). The vertical dashed lines indicate the location of the energy barrier.

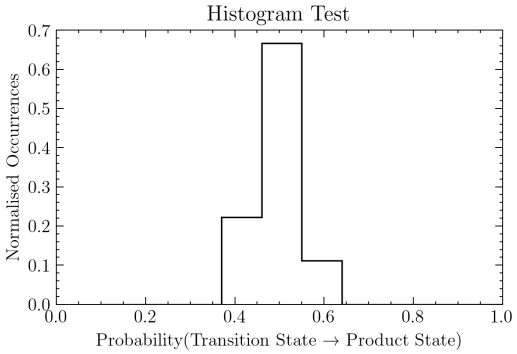


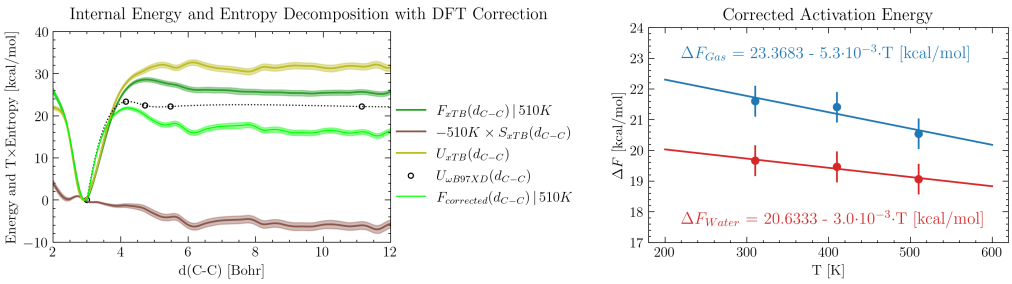
Figure 6.5: Histogram test to validate choice of CV and confirm location of energy barrier. Several simulations were initialised along the transition state. The graph shows the probability of the simulations evolving toward the product state.

stable bonded state ($d_{C-C} \approx 3$), rises to 30 kcal/mol at the transition state ($d_{C-C} \approx 4.75$), and plateaus at 31.5 kcal/mol shortly thereafter ($d_{C-C} \approx 5$), corresponding to the energy required to break the $C - C$ bond in the β-scission fully. Moreover, the entropy profile exhibits a continuous increase (illustrated as $-TS$ decrease) up to $d_{C-C} \approx 6$, which can be attributed to a rise in the number of degenerate configurations as the β-scission progresses. In vacuum, the barrier of activation of the internal energy is 31.52 ± 0.61 kcal/mol, while that of the entropy is 0.005 ± 0.002 kcal/mol/K. In water, these values decrease to 28.69 ± 0.54 kcal/mol for the internal energy and 0.005 ± 0.002 kcal/mol/K for the entropy.

In further analysis, the internal energy contribution was refined using higher-level DFT calculations, described in Section 6.2.3. This resulted in a correction of the free energy activation barrier, decreasing it by 7.92 kcal/mol and shifting it 0.45 Bohr closer to the reactant state. The corrected internal energies are represented by the black circles in Figure 6.6 (a), which also shows the shift of the previous FES (dark green line) to the corrected FES (light green line). This correction was applied correspondingly to all free energy profiles, and the resulting free energy barrier of activation and barrier of reaction are summarised in Table 6.2. Furthermore, the temperature dependence of the free energy barrier of activation in vacuum and water is shown in Figure 6.6. A linear fit of these values reveals the corrected barrier of activation of the internal energy and the entropy, where the former is the y-intercept and the latter is the slope. While the corrected activation internal energy is 23.4 kcal/mol for vacuum and 20.6 kcal/mol for water, the activation entropy is roughly the same and within the error bars of the previous values.

6.3.4 Evaluation of Reaction Rates

The reaction rate constants are computed using the generalised transition state theory (TST), described in Section 6.2.5. The velocity of crossing the transition state, $\langle|\dot{d}_{C-C}|\rangle^*$, was estimated from the simulations conducted



(a) (b)

Figure 6.6: (a) FES from the simulation in vacuum at 510 K (dark green line), decomposed into the internal energy contribution (yellow line) and the entropy contribution (brown line). The black circles represent the refined internal energy contribution obtained via high-level DFT and the corrected (refined) FES (light green line). (b) Corrected (refined) values of free energy barrier of activation (dots) with a linear fit (line), together with the equation of the linear fit (y-intercept corresponds to internal energy contribution and slope to entropy contribution). The blue data indicates the values for the system simulated in vacuum (gas), and the red data values for the system simulated in water.

Solvent	T [K]	l [Bohr]	ΔF^* [kcal/mol]	ΔF^R [kcal/mol]	k_β [s^{-1}]
Gas (Vacuum)	310	2.0	21.8 ± 0.5	20.1 ± 0.5	0.0016 ± 0.0013
	410	1.9	21.6 ± 0.5	19.1 ± 0.5	12 ± 7
	510	1.8	20.8 ± 0.5	17.9 ± 0.5	$5\,000 \pm 2\,000$
BA Monomer (Bulk)	410	1.9	19.1 ± 0.3	16.6 ± 0.4	300 ± 200
Water	310	1.8	20.1 ± 0.7	15.9 ± 0.7	0.03 ± 0.02
	410	1.7	19.6 ± 0.4	16.6 ± 0.4	200 ± 90
	510	1.6	19.5 ± 0.4	16.1 ± 0.5	$25\,000 \pm 10\,000$
Xylene	410	1.9	19.6 ± 0.4	18.1 ± 0.4	180 ± 120

Table 6.2: Summary of results in different solvent and temperature conditions (column 1 and 2): distances between reactant state and the transition state l (column 3), free energy activation barriers ΔF^* (column 4), free energy reaction barriers ΔF^R (column 5) and reaction rates k_β (column 6).

for the histogram test and was assumed to be $1.42 \cdot 10^{13}$ Bohr/s across all environments. The distances between the reactant and transition states, l, were measured from the corrected one-dimensional FESs and are reported in Table 6.2. Together with the corrected free energy barrier of activation, the reaction rate constants were computed, which are listed in Table 6.2.

At 410 K, rate constants in the BA monomer solvent (bulk) are the highest, while those in xylene and water match closely, and those in vacuum are the lowest. These findings indicate substantial solvent-dependent kinetic variations. For water and vacuum, rate constants are available across multiple temperatures, allowing for the estimation of the Arrhenius relationship:

$$\begin{aligned} k_{\beta,\text{water}} &= 2.82 \cdot 10^{13}\, e^{-\frac{21.24 \cdot 10^3}{RT}} \\ k_{\beta,\text{vacuum}} &= 4.55 \cdot 10^{13}\, e^{-\frac{23.41 \cdot 10^3}{RT}} \end{aligned} \tag{6.7}$$

where $R = 1.987$ kcal/mol/K is the gas constant. From the Arrhenius relationship, effective activation energies are determined to be $E_{a,\text{water}} = 21.24$ kcal/mol for water and $E_{a,\text{vacuum}} = 23.41$ kcal/mol for vacuum.

The Arrhenius relationship together with all computed reaction rate constants are plotted in Figure 6.7 (except the rate for xylene, as it fully overlaps with the rate of water at 410 K) and compared to reaction rate values from previous studies [130, 129, 134]. The results from this study closely match the experimental results obtained by Vir *et al.* [130] via high-temperature PLP-SEC using BA bulk as the solvent (green crosses and dotted line). The predicted reaction rate for the bulk solvent falls right on the Arrhenius relationship of Vir *et al.*. However, the computed reaction rate for the xylene solvent is 30 times larger than that measured by Peck *et al.* [129] with NMR. The most notable discrepancy exists between the reaction rates in vacuum predicted by this study and those predicted by Cuccato *et al.* [134], with a difference of three orders of magnitude. This arises because Cuccato *et al.* predicted the rates using *harmonic* TST with a *B3LYP/6-31+G(d,p)* QM scheme, where they did not account for anharmonic contributions, both

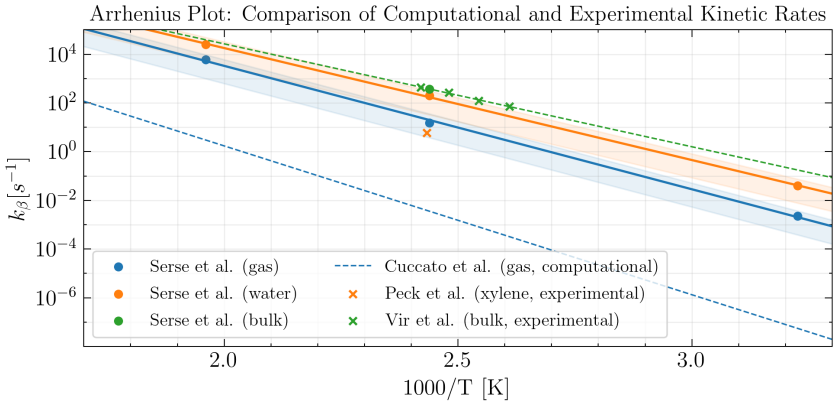


Figure 6.7: Comparison of reaction rates: Computed by Serse et al. in vacuum (gas) (blue dots and line), with water as solvent (orange dots and line), and BA monomers as solvent (bulk) (green dot). Computed by Cuccato et al. in vacuum (gas) (blue dotted line). Experimentally measured by Vir *et al.* with BA bulk as solvent (green crosses and dotted line) and measured by Peck et al. with xylene as solvent (orange cross).

from the solute and the solvent, that lower the barrier. Additionally, this study used high-level DFT corrections to refine the internal energy contributions, which resulted in a notable reduction in the energy barrier and consequently a larger reaction rate.

6.4 Conclusions

In this chapter, a collaborative research project was presented, where the FES and reaction rates of the β-scission of a BA dimer were determined under different temperature and solvent conditions. For the accurate prediction of the reaction kinetics, a QM/MM approach was employed, where the BA dimer undergoing β-scission was described with QM methods while the solvent was described with classical force fields. Additionally, higher-level DFT optimisation followed by ZPE corrections was utilised to refine the potential energy calculation of five configurations, resulting in a free energy correction of 7.92 kcal/mol across all solvent environments.

Non-tempered MetaD was employed to accelerate the sampling of the system. However, due to the extensive computational demand of the QM calculations and the limited computational resources, it was not possible to

obtain converging FESs from single trajectories. This issue was resolved by dividing the simulation into several independent runs, enabling the sampling of multiple crossings of the free energy barrier, as well as more flexible resource management. While it is not possible to merge independent trajectory data to determine the combined FES in a self-consistent manner, MFI proved invaluable for the post-processing. It enabled the estimation of the combined FES, and the bootstrap error of the FES, which served as a quantitative measure of the global convergence and the local uncertainty. On the one hand, this confirmed that the existing simulation data was sufficient to calculate the FESs accurately. On the other hand, it provided quantitative uncertainties for the free energy barriers of activation, which in turn enabled the calculation of the error bars for the reaction rates.

Rare events generalised TST was then used to determine the reaction rates of β-scission, which revealed that these increase with rising temperature. Furthermore, it was shown that the energy barrier is lowered and reaction rates increase when a solvent is present. The largest reaction rates were found when BA was used as the solvent. Crucially, the reaction rates were in good agreement with previous experimental studies, demonstrating the effectiveness and reliability of MFI in combining several short simulations to determine the FES of highly complex systems.

Chapter 7

General Conclusions and Outlook

Accurate estimation of FESs is one of the central challenges in MD simulations, particularly for systems in which metastable states are separated by substantial energy barriers [35]. Enhanced sampling methods, such as MetaD [2, 50, 49, 74] or US [1, 37, 58], have been widely adopted as a means to bias the potential energy of the system, thereby achieving ergodic sampling within accessible time scales. However, suboptimal biasing parameters often result in incomplete sampling of relevant high-energy configurations or poor convergence. Furthermore, independent MetaD simulations can not be combined in a self-consistent manner with standard MetaD estimators, thus restricting this method to single long trajectories. As a result, poorly converged MetaD simulations with inadequate biasing parameters are routinely discarded, even though their trajectory data contains valuable information. Consequently, a robust post-processing framework that permits the estimation of the FES from several independent MetaD simulations is of considerable practical significance, particularly for systems where long, continuous simulations are prohibitively expensive.

MFI [3, 77] addresses this need by initially determining the average mean force from trajectory data, which can be integrated to determine the FES. This approach has the advantage that independent mean force esti-

mates can be merged via a weighted average approach, enabling the estimation of the combined FES from independent simulations. MFI was first published by Marinova *et al.* [3] as a *proof of concept*. However, key questions regarding quantifying the uncertainty and convergence, interoperability with biasing methods other than MetaD and effectiveness in complex chemical systems remained unresolved.

The principal objectives investigated in this work were therefore:

1. Generalisation of the MFI formalism so that it can operate subject to arbitrary static or time-dependent biases, or a combination of those.
2. Development of numerical schemes suitable for efficient FES reconstruction, uncertainty estimation and convergence evaluation.
3. Demonstrate that multiple short and under-converged trajectories can be patched together to determine the combined FES accurately.
4. Validate the reliability and transferability of MFI by simulating complex chemical systems and reconstructing their FESs.
5. Provide an open-source implementation of the MFI algorithms, encourage collaborative projects and facilitate a wider adoption of MFI.

7.1 Contributions

This work's contribution started with a generalised framework of MFI [77] in Section 3.1. This allowed for the simultaneous application of multiple static and time-dependent biases within one simulation, which was subsequently post-processed with MFI, and the FES was determined. The effectiveness of this framework was demonstrated in Chapter 3 through the simulation of analytical potentials and alanine dipeptide, where a MetaD bias supplemented with a US bias resulted in accurate estimations of FESs. Chapter 5 further showed that, given an *a priori* FES estimate, a static *InvF* bias can be utilised to ”flatten” the underlying potential energy surface. Such a bias effectively transfers information from earlier simulations to subsequent ones,

facilitating prompt access to the previously sampled configuration space. This can be further augmented with a MetaD bias and customised surfaces, for example, to prioritise sampling of energy barriers, less converged regions (see Figure 7.2 a) or transition paths. Overall, this MFI framework affords additional flexibility in combining biasing strategies unavailable in existing enhanced-sampling techniques.

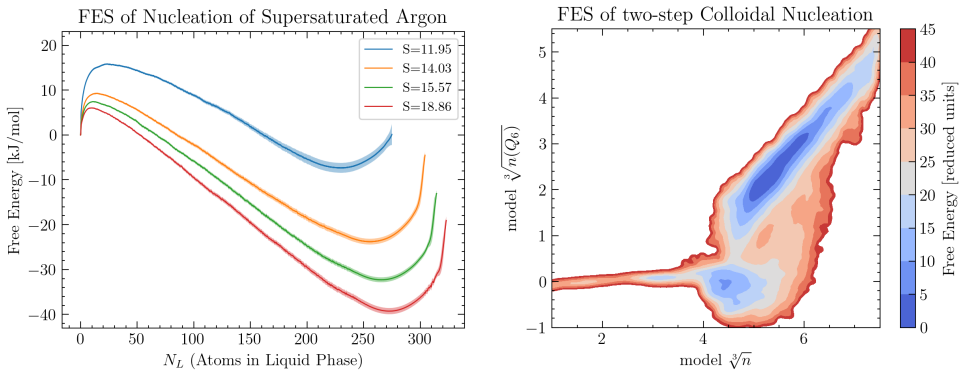
Next, two complementary convergence metrics have been developed in Section 3.2. The first quantifies the statistical uncertainty of the mean force. Since the mean force is obtained from a weighted average, its variance provides a reliable *on-the-fly* convergence estimation at negligible additional cost. However, in most applications, the uncertainty of the FES is of greater significance. For this reason, a bootstrap error analysis was implemented, where independent estimates of the mean force are randomly combined and integrated. This procedure produces an ensemble of FESs, which in turn provide an estimate of the uncertainty in the FES. Multiple examples confirmed that the average bootstrap error of the FES reproduces both the qualitative and quantitative behaviour of the average absolute deviation of the FES, while the error of the mean force provided an additional qualitative *on-the-fly* estimation of the convergence. Moreover, a reliable and fast integration algorithm was essential for accurately estimating the FES and computing the bootstrap error efficiently. To this end, a fast-Fourier-transform-based integration method was developed in Section 3.3.3, which is 100 times faster and more accurate than the Least-Squares Finite Difference integration method used in the original MFI implementation [3].

The main contribution of this thesis was validating that independent trajectories generated under diverse static and history-dependent biasing conditions can be merged, and that the combined FES can be estimated accurately and self-consistently. This enables researchers to restructure exceedingly expensive simulations into multiple shorter, independent runs. On the one side, this allows for increased computational efficiencies through

parallel execution and more flexibility to run simulations when computational allocations are available. On the other hand, it permits researchers to devise diverse biasing procedures for different regions of configurational space or simulation stages, and to iteratively improve biasing parameters without discarding under-converged trajectories. Even though other enhanced sampling methods enable parallel execution, adaptive sampling, or the combination of independent runs, MFI combines all these features. These capabilities were initially tested for analytical model potentials and alanine dipeptide in Section 3.4, then extended to three complex chemical systems.

In Section 4.2, the nucleation of argon vapour into a liquid droplet was investigated at various supersaturation conditions. The energy barrier separating the vapour state from the liquid state is markedly asymmetric (see Figure 7.1 a): the forward transition involves a steep barrier, whereas the reverse process requires escaping a deep basin with a gentler slope. A single MetaD simulation able to sample both directions with sufficient recrossings would have incurred a prohibitive cost. Instead, 50 forward and 50 backwards simulations were conducted, employing narrow Gaussian hills for the condensation and wider hills for evaporation. The condensation with the highest supersaturation level contained the largest and steepest energy barrier, motivating the use of an additional wall-like static bias potential. The simulation data was then post-processed with MFI, resulting in converged FESs. This study exemplified how heterogeneous biases can be tailored to different regions of configuration space, thereby enabling swift transitions of challenging energy barriers.

In Section 4.3, the two-step nucleation of a colloidal system from a homogeneous solution was simulated. Two CVs were used to describe this process: the number of colloidal particles in the dense phase, n, and the number of particles exhibiting crystalline order, $n(Q_6)$. Because calculating the latter CV imposed a computational bottleneck, an ML surrogate model was employed to accelerate its estimation. The system was first sim-



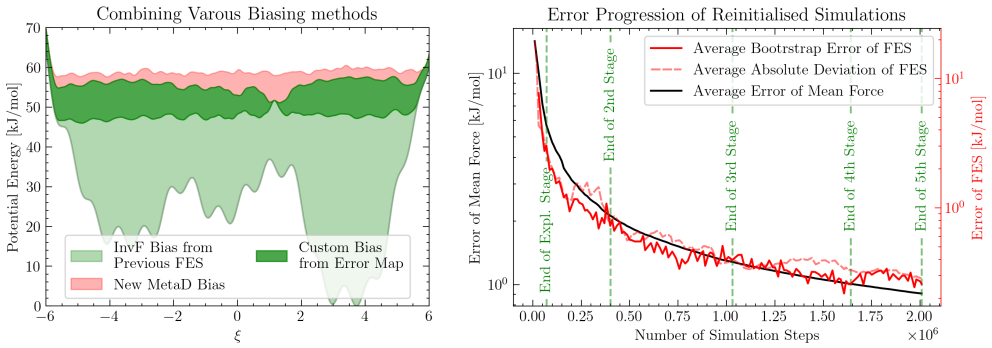
(a) (b)

Figure 7.1: (a) FES of nucleation of supersaturated argon vapour at different supersaturation levels, indicated in the legend. (b) FES of two-step nucleation of the colloidal system.

ulated with a non-tempered MetaD bias, followed by three WT-MetaD simulations, yielding markedly different sampling distributions. Post-processing with MFI produced the combined FES, illustrated in Figure 7.1 (b), which was well converged and clearly displayed a two-step nucleation pathway. This study demonstrated how data from iteratively optimised simulations can be merged and integrated rather than discarded.

In Chapter 6, the β-scission reaction in a butyl-acrylate dimer was analysed to predict reaction rates across different solvent and temperature conditions. This required expensive QM calculations to describe the electronic structure accurately, rendering simulations long enough to sample the reaction repeatedly inaccessible. Therefore, the simulation was divided into multiple shorter runs, which imposed a more accessible computational demand. Furthermore, computational efficiencies were realised through parallel execution, and aggregated results could be complemented with additional runs when required. Subsequent MFI analysis produced converged FESs, from which the reaction rates were computed via TST. Comparison with experimental reaction rates further validated the effectiveness and reliability of MFI for computing accurate FESs of complex and high-cost systems through multiple independent under-converged trajectories.

Although partitioning an expensive, long simulation into several shorter segments proved advantageous for high-cost systems, tests on simpler models revealed that the short simulations converged comparatively more slowly. Because a considerable fraction of each short run was spent revisiting low-energy basins, high-energy configurations were only sampled once the MetaD bias had accumulated sufficiently. This redundancy was mitigated by transferring information between runs via the static *InvF* bias, facilitating prompt access to the previously sampled high-energy configurations. In Chapter 5, this approach was further developed by introducing successive *biasing stages*: the system was first rapidly explored with an aggressive MetaD bias, followed by further biasing stages, employing an increasingly accurate InvF potential combined with a progressively gentler MetaD bias (see Figure 7.2). This protocol accelerated exploration of configuration space and enabled a faster transition to a less fluctuating MetaD bias, consistently improving the convergence.



(a) (b)

Figure 7.2: (a) Combination of various bias potentials: InvF bias from previous FES (light green region), custom bias constructed with error map to encourage sampling of less converged regions (dark green region), and MetaD bias (red region). (b) Error progression across multiple simulation stages: bootstrap error of FES (red line), absolute deviation of FES (red-dashed line), error of mean force (black line), and the end of each stage (green-dashed line).

This approach was further developed in Section 5.5, introducing *Serial Real-Time Reinitialisation* (SRTR) campaigns: automated criteria were

devised to initialise, terminate, and re-initialise simulations repeatedly. In Section 5.6, this concept was extended to multiple concurrent replicas through *Parallel Real-Time Reinitialisation* (PRTR) campaigns. The SRTR and PRTR campaigns were tested extensively on various analytical models, and compared to single simulations, parallel simulations and a *two-step* approach, consisting of an exploration stage followed by a simulation with a gentle MetaD. It was found that SRTR campaigns consistently converged faster than an equivalently long continuous simulation, whereas PRTR campaigns exhibited comparable convergence within reduced wall-clock time. On average, the *two-step* approach had the best accuracy, highlighting the effectiveness of starting with a fast exploration and suggesting a balance between straightforward protocols (*two-step*) and more elaborate schemes (SRTR) needs to be found. SRTR and PRTR represent a novel approach to automate the repeated reinitialisation of short simulations with iteratively improving biasing parameters. While it proved effective for cheap simulations on analytical potentials and alanine dipeptide, it still needs to be tested on complex chemical systems.

Lastly, making all algorithms developed for this work openly accessible at `github.com/mme-ucl/MFI` facilitated widespread adoption of MFI within the enhanced sampling community. The stable *pyMFI* package (`github.com/mme-ucl/pyMFI`) provides the core functionality used throughout Chapter 3, whereas the reinitialisation algorithms reside in separate modules: *MFIclass_1D* and *MFIclass_2D*. Public availability has already facilitated multiple collaborations, including the colloidal nucleation study of Section 4.3 and the β-scission investigation of Chapter 6. Additionally, numerous external researchers started using MFI in their work and utilised the pyMFI library to post-process their simulations.

7.2 Limitations and Future Work

Despite the progress achieved, several limitations remain. For instance, combining MetaD with US proved robust only when moderate force constants were employed, whereas very stiff restraints distorted the resulting mean forces. A possible remedy would be to analyse the data biased with the US on a separate, high–resolution grid. Then, the mean force would be interpolated onto the working grid and combined with the other data. Employing a smaller bandwidth would also be advantageous when sampling is concentrated in a small region of CV-space. Moreover, the quality of the reconstructed FES is sensitive to the kernel bandwidth chosen for the probability density calculation (see equation 2.92). The bandwidth was chosen heuristically throughout this thesis, whereas adaptive bandwidth selection [76, 151, 152] would reduce user bias and improve the final result and reproducibility.

Other limitations involve the SRTR and PRTR frameworks, which operate with relatively simple rules and criteria. Simulations are terminated solely based on the global uncertainty estimations without considering how effectively high-uncertainty regions are explored. Incorporating information, such as the residence time within basins, could expedite reinitialisation when the simulation remains trapped in a metastable state. Conversely, considering the accessibility to poorly converged regions could prevent premature termination when corrections to the mean force result in transient inflations of the uncertainty. Additionally, the initialisation of new simulations could be improved by estimating how much the biased potential energy surface (U+ *InvF*) deviates from a flat distribution. This would enable a better choice of the MetaD height and bias factor, such that the remaining energy barrier can be overcome with a minimally fluctuating MetaD bias. Also, a periodic reevaluation of F_{lim} would reduce reliance on initial parameter selection.

Furthermore, a terminal stage could be activated at the end of the cam-

paign, where the sampling is prioritised in important and less converged regions, such as the energy barrier. On the other hand, further investigations are required to determine the balance between straightforward (*two-step* approach) and complex (SRTR) sequential simulations. This vast space of parameter selection, rules and criteria, and the accompanying safeguards and exceptions, is incredibly challenging to implement effectively. Instead, an ML model could be trained to make these decisions. Given enough training data, such a model could, in principle, identify intricate relations between sampling, results and convergence. During his doctoral studies, the writer explored this idea and implemented, trained, and tested a deep-Q-learning reinforcement ML model [117, 153, 154]. While ML-steered decisions produced better results than random decision-making, the simple rules presented in this work performed far better. However, larger training datasets, longer optimisation windows and improved neural architectures could improve the performance. Whether an ML model could execute enhanced sampling methods better than some simple rules, a junior researcher, or come close to an expert in the field remains a question for future work.

Nevertheless, further work is not limited to the MFI methodology, but also to the publicly accessible MFI implementations. Crucially, embedding core routines of MFI directly into PLUMED [94, 93] would further facilitate access to the method. Besides, the MFI library could deliver a more user-friendly experience by providing more tutorials with additional use cases. Also, MFI currently enables the launch of MD simulations with GROMACS [71], which could be extended to LAMMPS [112], CP2K [140], OpenMM [155] and other popular MD software. More aspirational, coupling ML-derived force fields [156, 157], automatic CV discovery [158, 159, 160], and real-time MFI bias optimisation would create an end-to-end framework that would prepare the way for automated studies.

7.3 Concluding Remark

MFI has been advanced from a proof-of-concept to a versatile framework that unifies heterogeneous biased trajectories, delivers quantitative uncertainties, enables serial or parallel simulation campaigns and accommodates dynamic reinitialisation strategies. It provides a flexible blueprint for tackling free-energy problems that would otherwise be beyond reach. Continued refinements in automation, scalability and software integration will further broaden its impact across chemistry, biophysics and materials science.

Bibliography

[1] G M Torrie and J P Valleau. Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. *JOURNAL OF COMPUTATIONAL PHYSICS*, 23:187–199, 1977.

[2] Alessandro Laio and Michele Parrinello. Escaping free-energy minima. *Proceedings of the National Academy of Sciences*, 99(20):12562–12566, 2002.

[3] Veselina Marinova and Matteo Salvalaglio. Time-independent free energies from metadynamics via mean force integration. *Journal of Chemical Physics*, 151, 10 2019.

[4] Max Bonomi. Plumed masterclass 21.4: Metadynamics. `https://www.plumed.org/doc-v2.9/user-doc/html/masterclass-21-4.html`, 2021. [Accessed 23-01-2025].

[5] Serse F., Bjola A., Salvalaglio M., and Pelucchi M. Unveiling solvent effects on β-scissions through metadynamics and mean force integration. *Journal of Chemical Theory and Computation*, 20(14):6253–6262, 2024.

[6] Ludwig Boltzmann. Über die mechanische bedentung des zweiten hauptsatzes der wärmetheorie. *Sitzungsberichte der Kaiserlichen Akademie der Wissenschaften*, 53:195–220, 1927.

[7] John G. Kirkwood. Statistical mechanics of fluid mixtures. *The Journal of Chemical Physics*, 3:300–313, 1935.

[8] Robert W. Zwanzig. High-temperature equation of state by a perturbation method. i. nonpolar gases. *The Journal of Chemical Physics*, pages 1420–1426, 1954.

[9] Chfristophe Chipot and Andrew Pohorille. *Free Energy Calculations-Theory and Applications in Chemistry and Biology*, volume 86. Springer, 2007.

[10] P. Debye. Zur theorie der spezifischen wärmen. *Annalen der Physik*, 344(14):789–839, 1912.

[11] M. Born and R. Oppenheimer. Zur quantentheorie der molekeln. *Annalen der Physik*, 389(20):457–484, 1927.

[12] Rudolf Ernst Peierls. *Quantum Theory of Solids*. Clarendon Press Oxford, 1955.

[13] John C. Wheeler and Carl Blumstein. Modified moments for harmonic solids. *Physical Review B*, 6:4380, 12 1972.

[14] J. D. van der Waals. *On the Continuity of the Gaseous and Liquid States*. PhD thesis, University of Leiden, 1873.

[15] Rudiger Lichtenthaler Edmundo G. de Azevedo, John Prausnitz. *Molecular Thermodynamics of Fluid-Phase Equilibria*. Prentice-Hall international series in the physical and chemical engineering sciences. Pearson, 1969.

[16] H. Kamerlingh Onnes. Expression of the equation of state of gases and liquids by means of series. In *Proceedings of the KNAW*, volume 4, pages 125–147, Amsterdam, 1902. 1901–1902.

[17] Otto. Redlich and J. N. S. Kwong. On the thermodynamics of solutions. v. an equation of state. fugacities of gaseous solutions. *Chemical Reviews*, 44(1):233–244, 1949. PMID: 18125401.

[18] Ding-Yu Peng and Donald B. Robinson. A new two-constant equation of state. *Industrial and Engineering Chemistry Fundamentals*, 15:59–64, 1976.

[19] W.G. Chapman, K.E. Gubbins, G. Jackson, and M. Radosz. Saft: Equation-of-state solution model for associating fluids. *Fluid Phase Equilibria*, 52:31–38, 1989.

[20] Joachim Gross and Gabriele Sadowski. Perturbed-chain saft: An equation of state based on a perturbation theory for chain molecules. *Industrial & Engineering Chemistry Research*, 40(4):1244–1260, 2001.

[21] Simon Dufal, Vasileios Papaioannou, Majid Sadeqzadeh, Thomas Pogiatzis, Alexandros Chremos, Claire S. Adjiman, George Jackson, and Amparo Galindo. Prediction of thermodynamic properties and phase behavior of fluids and mixtures with the saft-γ mie group-contribution equation of state. *Journal of Chemical & Engineering Data*, 59(10):3272–3288, 2014.

[22] François Gygi and Giulia Galli. Ab initio simulation in extreme conditions. *Materials Today*, 8:26–32, 11 2005.

[23] Nathan A. Bernhardt and Ulrich H.E. Hansmann. Multifunnel landscape of the fold-switching protein rfah-ctd. *Journal of Physical Chemistry B*, 122:1600–1607, 2 2018.

[24] Jingfang Wang, Xintian Xu, Xinbo Zhou, Ping Chen, Huiying Liang, Xuan Li, Wu Zhong, and Pei Hao. Molecular simulation of sars-cov-2 spike protein binding to pangolin ace2 or human ace2 natural variants reveals altered susceptibility to infection. *The Journal of General Virology*, 101:921, 2020.

[25] Aaron R. Finney and Matteo Salvalaglio. Multiple pathways in nacl homogeneous crystal nucleation. *Faraday Discussions*, 235:56–80, 11 2021.

[26] Finney A. and Salvalaglio M. A variational approach to assess reaction coordinates for two-step crystallization. *J. Chem. Phys.*, 158:56–80, 3 2023.

[27] K. Nordlund. Historical review of computer simulation of radiation effects in materials. *Journal of Nuclear Materials*, 520:273–295, 7 2019.

[28] Paramvir Ahlawat, M. Ibrahim Dar, Pablo Piaggi, Michael Grätzel, Michele Parrinello, and Ursula Rothlisberger. Atomistic mechanism of the nucleation of methylammonium lead iodide perovskite from solution. *Chemistry of Materials*, 32:529–536, 1 2020.

[29] Paramvir Ahlawat, Alexander Hinderhofer, Essa A. Alharbi, Haizhou Lu, Amita Ummadisingu, Haiyang Niu, Michele Invernizzi, Shaik Mohammed Zakeeruddin, M. Ibrahim Dar, Frank Schreiber, Anders Hagfeldt, Michael Grätzel, Ursula Rothlisberger, and Michele Parrinello. A combined molecular dynamics and experimental study of two-step process enabling low-temperature formation of phase-pure alpha-fapbi3. *Science Advances*, 7(17):3326–3349, 2021.

[30] Matteo Salvalaglio, Claudio Perego, Federico Giberti, Marco Mazzotti, and Michele Parrinello. Molecular-dynamics simulations of urea nucleation from aqueous solution. *Proceedings of the National Academy of Sciences*, 112(1):E6–E14, 2015.

[31] Björn M. Burmann, Stefan H. Knauer, Anastasia Sevostyanova, Kristian Schweimer, Rachel A. Mooney, Robert Landick, Irina Artsimovitch, and Paul Rösch. An α helix to β barrel domain switch transforms the transcription factor rfah into a translation factor. *Cell*, 150:291–303, 7 2012.

[32] Christopher L. Burcham, Michael F. Doherty, Baron G. Peters, Sarah L. Price, Matteo Salvalaglio, Susan M. Reutzel-Edens, Louise S. Price, Ravi Kumar Reddy Addula, Nicholas Francia, Vikram Khanna, and Yongsheng Zhao. Pharmaceutical digital design: From chemical structure through crystal polymorph to conceptual crystallization process. *Crystal Growth & Design*, 24(13):5417–5438, 2024.

[33] David E. Shaw, Paul Maragakis, Kresten Lindorff-Larsen, Stefano Piana, Ron O. Dror, Michael P. Eastwood, Joseph A. Bank, John M. Jumper, John K. Salmon, Yibing Shan, and Willy Wriggers. Atomic-level characterization of the structural dynamics of proteins. *Science*, 330(6002):341–346, 2010.

[34] Thomas J Lane, Diwakar Shukla, Kyle A Beauchamp, and Vijay S Pande. To milliseconds and beyond: challenges in the simulation of protein folding. *Current Opinion in Structural Biology*, 23(1):58–65, 2013. Folding and binding / Protein-nucleic acid interactions.

[35] Carsten Hartmann, Ralf Banisch, Marco Sarich, Tomasz Badowski, and Christof Schütte. Characterization of rare events in molecular dynamics. *Entropy*, 16:350–376, 2014.

[36] Federico Giberti, Matteo Salvalaglio, and Michele Parrinello. Metadynamics studies of crystal nucleation. *IUCrJ*, 2:256–266, 2 2015.

[37] Beno t Roux. The calculation of the potential of mean force using computer simulations. *Computer Physics Communications*, 91:275–282, 1995.

[38] Lizhe Zhu, Fu Kit Sheong, Siqin Cao, Song Liu, Ilona C Unarta, and Xuhui Huang. Taps: A traveling-salesman based automated path searching method for functional conformational changes of biological macromolecules. *The Journal of Chemical Physics*, 150:124105, 2019.

[39] Phillip L. Geissler, Christoph Dellago, and David Chandler. Kinetic pathways of ion pair dissociation in water. *Journal of Physical Chemistry B*, 103:3706–3710, 5 1999.

[40] Peter G. Bolhuis, David Chandler, Christoph Dellago, and Phillip L. Geissler. Transition path sampling: Throwing ropes over rough mountain passes, in the dark. *Annual Review of Physical Chemistry*, 53:291–318, 2002.

[41] Kim S Huber GA. Weighted-ensemble brownian dynamics simulations for protein association reactions. *Biophysical Journal*, 70(1):97–110, 1 1996.

[42] Anton K. Faradjian and Ron Elber. Computing time scales from reaction coordinates by milestoning. *The Journal of Chemical Physics*, 120(23):10880–10889, 06 2004.

[43] Rosalind J. Allen, Patrick B. Warren, and Pieter Rein ten Wolde. Sampling rare switching events in biochemical networks. *Phys. Rev. Lett.*, 94:018104, Jan 2005.

[44] Gregory R. Bowman, Daniel L. Ensign, and Vijay S. Pande. Enhanced modeling via network theory: Adaptive sampling of markov state models. *Journal of Chemical Theory and Computation*, 6(3):787–794, 2010. PMID: 23626502.

[45] Jerome Henin, Tony Lelievre, Michael R Shirts, Omar Valsson, and Lucie Delemotte. Enhanced sampling methods for molecular dynamics simulations. *Living Journal of Computational Molecular Science*, 4(1), 2022.

[46] Edgar Olehnovics, Yifei Michelle Liu, Nada Mehio, Ahmad Y. Sheikh, Michael R. Shirts, and Matteo Salvalaglio. Assessing the accuracy and efficiency of free energy differences obtained from reweighted

flow-based probabilistic generative models. *Journal of Chemical Theory and Computation*, 20(14):5913–5922, 2024. PMID: 38984825.

[47] Johannes Kästner and Walter Thiel. Bridging the gap between thermodynamic integration and umbrella sampling provides a novel analysis method: ”umbrella integration”. *Journal of Chemical Physics*, 123, 10 2005.

[48] Mark Tuckerman. *Statistical Mechanics : Theory and Molecular Simulation*. Oxford University Press, 2010.

[49] Alessandro Barducci, Giovanni Bussi, and Michele Parrinello. Well-tempered metadynamics: A smoothly converging and tunable free-energy method. *Physical Review Letters*, 100, 1 2008.

[50] Alessandro Barducci, Massimiliano Bonomi, and Michele Parrinello. Metadynamics. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1:826–843, 9 2011.

[51] Eric Darve and Andrew Pohorille. Calculating free energies using average force. *The Journal of Chemical Physics*, 115(20):9169–9183, 11 2001.

[52] Eric Darve, Michael A. Wilson, and Andrew Pohorille and. Calculating free energies using a scaled-force molecular dynamics algorithm. *Molecular Simulation*, 28(1-2):113–144, 2002.

[53] Jeffrey Comer, James C. Gumbart, Jérôme Hénin, Tony Lelièvre, Andrew Pohorille, and Christophe Chipot. The adaptive biasing force method: Everything you always wanted to know but were afraid to ask. *The Journal of Physical Chemistry B*, 119(3):1129–1151, 2015. PMID: 25247823.

[54] Bob Robey and Yuliana Zamora. *Parallel and High Performance Computing*. Manning Publications, version 1 edition, 2019.

[55] Dušanka Janežič, Urban Borštnik, and Matej Praprotnik. *Parallel approaches in molecular dynamics simulations*. Springer London, 2009.

[56] David J. Earl and Michael W. Deem. Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7:3910–3916, 12 2005.

[57] Kumar S., Bouzida D., Swendsen R., Kollman P., and Rosenbergl J. The weighted histogram analysis method for free-energy calculations on biomolecules. i. the method. *J. Comp. Chem.*, 13:1011–1021, 1992.

[58] Johannes Kästner. Umbrella sampling. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1:932–942, 2011.

[59] Charles H Bennett. Efficient estimation of free energy differences from monte carlo data. *Journal of Computational Physics*, 22(2):245–268, 1976.

[60] Michael R. Shirts and John D. Chodera. Statistically optimal analysis of samples from multiple equilibrium states. *The Journal of Chemical Physics*, 129(12):124105, 09 2008.

[61] Johannes Kästner and Walter Thiel. Analysis of the statistical error in umbrella sampling simulations by umbrella integration. *Journal of Chemical Physics*, 124, 6 2006.

[62] Johannes Kästner. Umbrella integration in two or more reaction coordinates. *Journal of Chemical Physics*, 131, 2009.

[63] de Las Heras D. and Schmidt M. Better than counting: Density profiles from force sampling. *Phys. Rev. Lett.*, 120:218001, 2018.

[64] Rotenberg B. Use the force! reduced variance estimators for densities, radial distribution functions, and local mobilities in molecular simulations. *J. Chem. Phys.*, 153:150902, 2020.

[65] Isaac Newton. *Philosophiae naturalis principia mathematica*. Jussu Societatis Regiae ac Typis Josephi Streater; prostat apud plures Bibliopolas Londini, 1687.

[66] Paul K. Weiner and Peter A. Kollman. Amber: Assisted model building with energy refinement. a general program for modeling molecules and their interactions. *Journal of Computational Chemistry*, 2, 1981.

[67] Scott J. Weiner, Peter A. Kollman, David A. Case, U. Chandra Singh, Caterina Ghio, Guliano Alagona, Salvatore Profeta, and Paul Weiner. A new force field for molecular mechanical simulation of nucleic acids and proteins. *Journal of the American Chemical Society*, 106(3):765–784, 1984.

[68] Junmei Wang, Romain M Wolf, James W Caldwell, Peter A Kollman, and David A Case. Development and testing of a general amber force field. *J Comput Chem*, 25:1157–1174, 2004.

[69] Viktor Hornak, Robert Abel, Asim Okur, Bentley Strockbine, Adrian Roitberg, and Carlos Simmerling. Comparison of multiple amber force fields and development of improved protein backbone parameters. *Proteins: Structure, Function and Genetics*, 65:712–725, 11 2006.

[70] Daan Frenkel and Berend Smit. *Understanding molecular simulation: From algorithms to applications*. Academic Press, 1996.

[71] Lindahl, Abraham, Hess, and van der Spoel. Gromacs 2021.5 manual. *GROMACS*, 1 2022.

[72] Tamar Schlick. *Molecular Modeling and Simulation: An Interdisciplinary Guide*. Springer, 2nd edition edition, 2010.

[73] Pratyush Tiwary and Michele Parrinello. A time-independent free energy estimator for metadynamics. *Journal of Physical Chemistry B*, 119:736–742, 1 2015.

[74] M. Bonomi, A. Barducci, and M. Parrinello. Reconstructing the equilibrium boltzmann distribution from well-tempered metadynamics. *Journal of Computational Chemistry*, 30:1615–1621, 8 2009.

[75] Omar Valsson and Michele Parrinello. Variational approach to enhanced sampling and free energy calculations. *Phys. Rev. Lett.*, 113:090601, Aug 2014.

[76] Invernizzi M. and Parrinello M. Rethinking metadynamics: From bias potentials to probability distributions. *J. Phys. Chem. Lett.*, 11(7):2731–2736, 2020.

[77] Antoniu Bjola and Matteo Salvalaglio. Estimating free energy surfaces and their convergence from multiple, independent static and history-dependent biased molecular-dynamics simulations with mean force integration. *ChemRxiv*, 2024.

[78] Awasthi S., Kapil V., and Nair N. Sampling free energy surfaces as slices by combining umbrella sampling and metadynamics. *J. Comput. Chem.*, 37:1413–1424, 6 2016.

[79] Philip B. Bevington and D. Keith Robinson. *Data Reduction and Error Analysis for the Physical Sciences*. McGraw-Hill, 1969.

[80] Leslie Kish. *Survey sampling*. John Wiley and Sons, 1965.

[81] James Kirchner. Weighted averages and their uncertainties, 2006.

[82] K.A. Stroud and Dexter Booth. *Engineering Mathematics*. Palgrave Macmillan, 2013.

[83] L. V. Blake. A modified simpson's rule and fortran subroutine for cumulative numerical integration of a function defined by data points. Technical report, Naval Research Lab Washington DC, 4 1971. NTIS Issue Number: 197113.

[84] J D'Errico. Inverse (integrated) gradient, 2006. Accessed on 06 March, 2025.

[85] D. C.-L. Fong and M. A. Saunders. Lsmr: An iterative algorithm for sparse least-squares problems. *SIAM Journal on Scientific Computing*, 33:2950–2971, 2011. arXiv:1006.0758.

[86] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[87] Steven L Brunton and Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 1 edition, 2 2019.

[88] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Comput.*, 19:297–301, 1965.

[89] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser,

Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[90] Robert T Frankot and Rama Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 10:439–451, 6 1988.

[91] F Huhn, D Schanz, S Gesemann, P Manovski, and A Schröder. Pressure reconstruction from lagrangian particle tracking with fft integration. In *18th International Symposium on the Application of Laser and Imaging Techniques to Fluid Mechanics*, pages 1–15, 7 2016.

[92] Branduardi D., Bussi G., and Parrinello M. Metadynamics with adaptive gaussians. *J. Chem. Theory Comput.*, 8:2247–2254, 6 2012.

[93] Gareth A. Tribello, Massimiliano Bonomi, Davide Branduardi, Carlo Camilloni, and Giovanni Bussi. Plumed 2: New feathers for an old bird. *Computer Physics Communications*, 185(2):604–613, 2014.

[94] Massimiliano Bonomi, Giovanni Bussi, Gareth A. Tribello, Carlo Camilloni, Pavel Banáš, Alessandro Barducci, Mattia Bernetti, Peter G. Bolhuis, Sandro Bottaro, Davide Branduardi, Riccardo Capelli, Paolo Carloni, Michele Ceriotti, Andrea Cesari, Haochuan Chen, Wei Chen, Francesco Colizzi, Sandip De, Marco De La Pierre, Davide Donadio, Viktor Drobot, Bernd Ensing, Andrew L. Ferguson, Marta Filizola, James S. Fraser, Haohao Fu, Piero Gasparotto, Francesco Luigi Gervasio, Federico Giberti, Alejandro Gil-Ley, Toni Giorgino, Gabriella T. Heller, Glen M. Hocky, Marcella Iannuzzi, Michele Invernizzi, Kim E. Jelfs, Alexander Jussupow, Evgeny Kirilin, Alessandro Laio, Vittorio Limongelli, Kresten Lindorff-Larsen, Thomas Löhr, Fabrizio Marinelli, Layla Martin-Samos, Mat-

teo Masetti, Ralf Meyer, Angelos Michaelides, Carla Molteni, Tetsuya Morishita, Marco Nava, Cristina Paissoni, Elena Papaleo, Michele Parrinello, Jim Pfaendtner, Pablo Piaggi, Giovanni Maria Piccini, Adriana Pietropaolo, Fabio Pietrucci, Silvio Pipolo, Davide Provasi, David Quigley, Paolo Raiteri, Stefano Raniolo, Jakub Rydzewski, Matteo Salvalaglio, Gabriele Cesare Sosso, Vojtěch Spiwok, Jiří Šponer, David W.H. Swenson, Pratyush Tiwary, Omar Valsson, Michele Vendruscolo, Gregory A. Voth, and Andrew White. Promoting transparency and reproducibility in enhanced molecular simulations. *Nature Methods 2019 16:8*, 16:670–673, 7 2019.

[95] Dimo Kashchiev. *Nucleation: Basic Theory with Applications*. Butterworth-Heinemann, Oxford, 2000.

[96] Aaron R. Finney and Matteo Salvalaglio. Molecular simulation approaches to study crystal nucleation from solutions: Theoretical considerations and computational challenges. *WIREs Computational Molecular Science*, 14(1):e1697, 2024.

[97] B. Vonnegut and Henry Chessin. Ice nucleation by coprecipitated silver iodide and silver bromide. *Science*, 174(4012):945–946, 1971.

[98] Alexei Kiselev, Felix Bachmann, Philipp Pedevilla, Stephen J. Cox, Angelos Michaelides, Dagmar Gerthsen, and Thomas Leisner. Active sites in heterogeneous ice nucleation—the example of k-rich feldspars. *Science*, 355(6323):367–371, 2017.

[99] Paul J. M. Smeets, Aaron R. Finney, Wouter J. E. M. Habraken, Fabio Nudelman, Heiner Friedrich, Jozua Laven, James J. De Yoreo, P. Mark Rodger, and Nico A. J. M. Sommerdijk. A classical view on nonclassical nucleation. *Proceedings of the National Academy of Sciences*, 114(38):E7882–E7890, 2017.

[100] Lunna Li, Matteo Paloni, Aaron R. Finney, Alessandro Barducci, and Matteo Salvalaglio. Nucleation of biomolecular condensates from finite-sized simulations. *The Journal of Physical Chemistry Letters*, 14(7):1748–1755, 2023. PMID: 36758221.

[101] Juliane Weber, Jacquelyn N. Bracco, Ke Yuan, Vitalii Starchenko, and Andrew G. Stack. Studies of mineral nucleation and growth across multiple scales: Review of the current state of research using the example of barite (baso4). *ACS Earth and Space Chemistry*, 5(12):3338–3361, 2021.

[102] Fengchang Yang, Ke Yuan, Andrew G. Stack, and Vitalii Starchenko. Numerical study of mineral nucleation and growth on a substrate. *ACS Earth and Space Chemistry*, 6(7):1655–1665, 2022.

[103] Matteo Salvalaglio, Pratyush Tiwary, Giovanni Maria Maggioni, Marco Mazzotti, and Michele Parrinello. Overcoming time scale and finite size limitations to compute nucleation rates from small scale well tempered metadynamics simulations. *The Journal of chemical physics*, 145(21):211925, 2016.

[104] Guram Chkonia, Judith Wölk, Reinhard Strey, Jan Wedekind, and David Reguera. Evaluating nucleation rates in direct simulations. *The Journal of Chemical Physics*, 130(6):064505, 02 2009.

[105] Diemand J., Angélil R., Tanaka K., and Tanaka H. Large scale molecular dynamics simulations of homogeneous nucleation. *The Journal of Chemical Physics*, 139:074309, 2013.

[106] J E Lennard-Jones. Cohesion. *Proceedings of the Physical Society*, 43(5):461, sep 1931.

[107] Giovanni Bussi, Davide Donadio, and Michele Parrinello. Canonical sampling through velocity rescaling. *The Journal of Chemical Physics*, 126(1):014101, 01 2007.

[108] Pieter Rein ten Wolde and Daan Frenkel. Computer simulation study of gas–liquid nucleation in a lennard-jones system. *The Journal of Chemical Physics*, 109(22):9901–9918, 12 1998.

[109] Dietrich F., Advincula X., Gobbo G., Bellucci M., and Salvalaglio M. Machine learning nucleation collective variables with graph neural networks. *J. Chem. Theory Comput.*, 10 2023.

[110] Paul J. Steinhardt, David R. Nelson, and Marco Ronchetti. Bond-orientational order in liquids and glasses. *Phys. Rev. B*, 28:784–805, Jul 1983.

[111] Gareth A. Tribello, Federico Giberti, Gabriele C. Sosso, Matteo Salvalaglio, and Michele Parrinello. Analyzing and driving cluster formation in atomistic simulations. *Journal of Chemical Theory and Computation*, 13(3):1317–1327, 2017. PMID: 28121147.

[112] Thompson A., Aktulga H., Berger R., Bolintineanu D., Brown W., Crozier P., Veld P., Kohlmeyer A., Moore S., and et al T., Nguyen. Lammps - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comput. Phys. Commun.*, 271, 2 2022.

[113] Derjaguin B. and Landau L. The theory of stability of highly charged lyophobic sols and coalescence of highly charged particles in electrolyte solutions. *Acta Physicochim. URSS*, 14:58, 1941.

[114] Verwey E. and Overbeek J. *Theory of the Stability of Lyophobic Colloids: The Interaction of Sol Particles Having an Electric Double Layer*. Elsevier, 1962.

[115] Loeb A., Overbeek J., Wiersema P., and King C. The electrical double layer around a spherical colloid particle. *J. Electrochem. Soc.*, 108:269, 1961.

[116] Florian M. Dietrich and Matteo Salvalaglio. On the reproducibility of free energy surfaces in machine-learned collective variable spaces. *ChemRxiv*, 2025.

[117] Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., Killeen T., Lin Z., Gimelshein N., Antiga L., and et al. Pytorch: An imperative style, high-performance deep learning library. In H Wallach, H Larochelle, A Beygelzimer, F d Alché-Buc, E Fox, and R Garnett, editors, *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, volume 32. Curran Associates, Inc., 2019.

[118] Michaud-Agrawal N., Denning E., Woolf T., and Beckstein O. Mdanalysis: A toolkit for the analysis of molecular dynamics simulations. *J. Comput. Chem.*, 32:2319–2327, 7 2011.

[119] Richard J. Gowers, Max Linke, Jonathan Barnoud, Tyler John Edward Reddy, Manuel N. Melo, Sean L. Seyler, Jan Domanski, David L. Dotson, Sebastien Buchoux, Ian M. Kenney, et al. Mdanalysis: A python package for the rapid analysis of molecular dynamics simulations. In *Proceedings of the 15th PYTHON in science conference*, pages 98 – 105. Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2016.

[120] McGibbon R., Beauchamp K., Harrigan M., Klein C., Swails J., Hernández C., Schwantes C., Wang L., Lane T., and Pande V. Mdtraj: A modern open library for the analysis of molecular dynamics trajectories. *Biophys. J.*, 109:1528–1532, 10 2015.

[121] Bradbury J., Frostig R., Hawkins P., Johnson M., Leary C., Maclaurin D., Necula G., Paszke A., VanderPlas J., and et al S., Wanderman-Milne. Jax: composable transformations of python +numpy programs. `http://github.com/google/jax`, 2018. Accessed: 2023-08-18.

[122] Heek J., Levskaya A., Oliver A., Ritter M., Rondepierre B., Steiner A., and van Zee M. Flax: A neural network library and ecosystem for jax. `http://github.com/google/flax`, 2020. Accessed: 2023-08-18.

[123] Giorgino T. Pycv: a plumed 2 module enabling the rapid prototyping of collective variables in python. *J. Open Source Software*, 4:1773, 10 2019.

[124] Luca Maragliano, Alexander Fischer, Eric Vanden-Eijnden, and Giovanni Ciccotti. String method in collective variables: Minimum free energy paths and isocommittor surfaces. *The Journal of Chemical Physics*, 125(2):024106, 07 2006.

[125] Peter Astrand. *Subprocess Management*. Python Software Foundation, 3.12 edition, 2024. Accessed: 2025-05-07.

[126] Michael Buback, Robert G. Gilbert, Robin A. Hutchinson, Bert Klumperman, Frank-Dieter Kuchta, Bart G. Manders, Kenneth F. O'Driscoll, Gregory T. Russell, and Johannes Schweer. Critically evaluated rate coefficients for free-radical polymerization, 1. propagation rate coefficient for styrene. *Macromolecular Chemistry and Physics*, 196(10):3267–3280, 1995.

[127] Sabine Beuermann, Michael Buback, Thomas P. Davis, Robert G. Gilbert, Robin A. Hutchinson, Atsushi Kajiwara, Bert Klumperman, and Gregory T. Russell. Critically evaluated rate coefficients for free-radical polymerization, 3. propagation rate coefficients for alkyl methacrylates. *Macromolecular Chemistry and Physics*, 201(12):1355–1364, 2000.

[128] Anil B. Vir, Y. W. Marien, Paul H. M. Van Steenberge, Christopher Barner-Kowollik, Marie-Françoise Reyniers, Guy B. Marin, and Dagmar R. D'hooge. Access to the β-scission rate coefficient in acrylate

radical polymerization by careful scanning of pulse laser frequencies at elevated temperature. *React. Chem. Eng.*, 3:807–815, 2018.

[129] Adam N. F. Peck and Robin A. Hutchinson. Secondary reactions in the high-temperature free radical polymerization of butyl acrylate. *Macromolecules*, 37(16):5944–5951, 2004.

[130] Anil B. Vir, Yoshi W. Marien, Paul H. M. Van Steenberge, Christopher Barner-Kowollik, Marie-Françoise Reyniers, Guy B. Marin, and Dagmar R. D'hooge. From n-butyl acrylate arrhenius parameters for backbiting and tertiary propagation to beta-scission via stepwise pulsed laser polymerization. *Polym. Chem.*, 10:4116–4125, 2019.

[131] Alessandro Marongiu, Tiziano Faravelli, Giulia Bozzano, Mario Dente, and Eliseo Ranzi. Thermal degradation of poly(vinyl chloride). *Journal of Analytical and Applied Pyrolysis*, 70(2):519–553, 2003.

[132] Onur Dogu, Matteo Pelucchi, Ruben Van de Vijver, Paul H. M. Van Steenberge, Dagmar R. D'hooge, Alberto Cuoci, Marco Mehl, Alessio Frassoldati, Tiziano Faravelli, and Kevin M. Van Geem. The chemistry of chemical recycling of solid plastic waste via pyrolysis and gasification: State-of-the-art, challenges, and future directions. *Progress in Energy and Combustion Science*, 84:100901, 2021.

[133] Andrea Locaspi, Matteo Pelucchi, Marco Mehl, and Tiziano Faravelli. Towards a lumped approach for solid plastic waste gasification: Polyethylene and polypropylene pyrolysis. *Waste Management*, 156:107–117, 2023.

[134] Danilo Cuccato, Evangelos Mavroudakis, Marco Dossi, and Davide Moscatelli. A density functional theory study of secondary reactions in n-butyl acrylate free radical polymerization. *Macromolecular Theory and Simulations*, 22(2):127–135, 2013.

[135] Noy Cohen and Sydney William Benson. Estimation of heats of formation of organic compounds by additivity methods. *Chemical Reviews*, 93(7):2419–2438, 1993.

[136] H. K. Eigenmann, D. M. Golden, and Sydney William Benson. Revised group additivity parameters for the enthalpies of formation of oxygen-containing organic compounds. *The Journal of Physical Chemistry*, 77(13):1687–1691, 1973.

[137] Maarten K. Sabbe, Mark Saeys, Marie-Françoise Reyniers, Guy B. Marin, Veronique Van Speybroeck, and Michel Waroquier. Group additive values for the gas phase standard enthalpy of formation of hydrocarbons and hydrocarbon radicals. *The Journal of Physical Chemistry A*, 109(33):7466–7480, 2005. PMID: 16834116.

[138] Baron Peters. *Reaction Rate Theory and Rare Events*. Elsevier Science, 1 edition, 3 2017.

[139] Serse, Francesco and Salvalaglio, Matteo and Pelucchi, Matteo. First principles assessment of solvent induced cage effects on intramolecular hydrogen transfer in the free radical polymerization of acrylates. *Phys. Chem. Chem. Phys.*, 2024.

[140] Thomas D. Kühne, Marcella Iannuzzi, Mauro Del Ben, Vladimir V. Rybkin, Patrick Seewald, Frederick Stein, Teodoro Laino, Rustam Z. Khaliullin, Ole Schütt, Florian Schiffmann, Dorothea Golze, Jan Wilhelm, Sergey Chulkov, Mohammad Hossein Bani-Hashemian, Valéry Weber, Urban Borštnik, Mathieu Taillefumier, Alice Shoshana Jakobovits, Alfio Lazzaro, Hans Pabst, Tiziano Müller, Robert Schade, Manuel Guidon, Samuel Andermatt, Nico Holmberg, Gregory K. Schenter, Anna Hehn, Augustin Bussy, Fabian Belleflamme, Gloria Tabacchi, Andreas Glöß, Michael Lass, Iain Bethune, Christopher J. Mundy, Christian Plessl, Matt Watkins, Joost VandeVondele, Matthias

Krack, and Jürg Hutter. Cp2k: An electronic structure and molecular dynamics software package - quickstep: Efficient and accurate electronic structure calculations. *The Journal of Chemical Physics*, 152(19):194103, 05 2020.

[141] Shuichi Nosé. A unified formulation of the constant temperature molecular dynamics methods. *The Journal of Chemical Physics*, 81(1):511–519, 07 1984.

[142] Sprenger, K. G. and Jaeger, Vance W. and Pfaendtner, Jim. The general amber force field (gaff) can accurately predict thermodynamic and transport properties of many ionic liquids. *The Journal of Physical Chemistry B*, 119(18):5882–5895, 2015. PMID: 25853313.

[143] Ulrich Essmann, Lalith Perera, Max L. Berkowitz, Tom Darden, Hsing Lee, and Lee G. Pedersen. A smooth particle mesh ewald method. *The Journal of Chemical Physics*, 103(19):8577–8593, 11 1995.

[144] Stefan Grimme, Christoph Bannwarth, and Philip Shushkov. A robust and accurate tight-binding quantum chemical method for structures, vibrational frequencies, and noncovalent interactions of large molecular systems parametrized for all spd-block elements (z = 1–86). *Journal of Chemical Theory and Computation*, 13(5):1989–2009, 2017. PMID: 28418654.

[145] Teodoro Laino, Fawzi Mohamed, Alessandro Laio, and Michele Parrinello. An efficient linear-scaling electrostatic coupling for treating periodic boundary conditions in qm/mm simulations. *Journal of Chemical Theory and Computation*, 2(5):1370–1378, 2006. PMID: 26626844.

[146] Ilaria Gimondi, Gareth A Tribello, and Matteo Salvalaglio. Building maps in collective variable space. *The Journal of chemical physics*, 149(10):104104, 2018.

[147] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[148] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. V. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Jr. Montgomery, J. E. Peralta, F. Ogliaro, M. J. Bearpark, J. J. Heyd, E. N. Brothers, K. N. Kudin, V. N. Staroverov, T. A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. P. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman, and D. J. Fox. Gaussian 16 revision c.01, 2016. Gaussian Inc. Wallingford CT.

[149] Ilaria Gimondi and Matteo Salvalaglio. CO2 packing polymorphism under pressure: Mechanism and thermodynamics of the i-iii polymorphic transition. *The Journal of Chemical Physics*, 147(11):114502, 2017.

[150] Dhiman Ray and Michele Parrinello. Kinetics from metadynamics: Principles, applications, and outlook. *Journal of Chemical Theory and Computation*, 19(17):5649–5670, 2023. PMID: 37585703.

[151] Ian Abramson. On bandwidth variation in kernel estimates-a square root law. *Annals of Statistics*, 10:1217–1223, 1982.

[152] George R. Terrell and David W. Scott. Variable kernel density estimation. *The Annals of Statistics*, 20(3):1236–1265, 1992.

[153] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2 2015.

[154] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arxiv.org*, 12 2013.

[155] Peter Eastman, Raimondas Galvelis, Raúl P. Peláez, Charlles R. A. Abreu, Stephen E. Farr, Emilio Gallicchio, Anton Gorenko, Michael M. Henry, Frank Hu, Jing Huang, Andreas Krämer, Julien Michel, Joshua A. Mitchell, Vijay S. Pande, João PGLM Rodrigues, Jaime Rodriguez-Guerra, Andrew C. Simmonett, Sukrit Singh, Jason Swails, Philip Turner, Yuanqing Wang, Ivy Zhang, John D. Chodera, Gianni De Fabritiis, and Thomas E. Markland. Openmm 8: Molecular dynamics simulation with machine learning potentials. *The Journal of Physical Chemistry B*, 128(1):109–116, 2024. PMID: 38154096.

[156] Volker L. Deringer, Miguel A. Caro, and Gábor Csányi. A general-purpose machine-learning force field for bulk and nanostructured phosphorus. *Nature Communications*, 11, 12 2020.

[157] Dávid Péter Kovács, Ilyes Batatia, Eszter Sára Arany, and Gábor Csányi. Evaluation of the mace force field architecture: From medicinal chemistry to materials science. *The Journal of Chemical Physics*, 159(4):044118, 07 2023.

[158] Mary A. Rohrdanz, Wenwei Zheng, Mauro Maggioni, and Cecilia Clementi. Determination of reaction coordinates via locally scaled diffusion map. *The Journal of Chemical Physics*, 134(12):124116, 03 2011.

[159] Guillermo Pérez-Hernández, Fabian Paul, Toni Giorgino, Gianni De Fabritiis, and Frank Noé. Identification of slow molecular order parameters for markov model construction. *The Journal of Chemical Physics*, 139(1):015102, 07 2013.

[160] Pratyush Tiwary and B. J. Berne. Spectral gap optimization of order parameters for sampling complex molecular systems. *Proceedings of the National Academy of Sciences*, 113(11):2839–2844, 2016.

Appendix A

Additional Results

A.1 Synthetic-Noise Benchmark for the Integration of the Force

To quantify how robust the three integration schemes of Section 3.3 (FFT–based, Simpson, and `intgrad1`) are to statistical scatter, an ensemble of artificial "simulation" gradients was generated and integrated. The procedure is summarised below.

A smooth analytical force $f_{\mathrm{ref}}(\xi)$ was obtained by differentiating an analytical potential $F_{\mathrm{ref}}(\xi)$ on an equidistant grid $\{\xi_i\}_{i=1}^{N}$ with spacing $\Delta\xi$.

Each replica adds a zero–mean Gaussian perturbation η_i to the reference force,

$$f_i^{\mathrm{noisy}} = f_{\mathrm{ref}}(\xi_i) + \eta_i,$$

where the random field $\boldsymbol{\eta}$ is produced by the helper `generate_noise`:

```
def generate_noise(shape, noise_level=1.0, correlation_length=None):
    if noise_level == 0:
        return np.zeros(shape)

    # uncorrelated noise
    noise = np.random.normal(0, noise_level, shape)

    # impose spatial correlations if requested
```

```
    if correlation_length is not None:
        noise = gaussian_filter(noise, sigma=correlation_length)

    # rescale so that the average absolute amplitude equals `noise_level`
    return noise * noise_level / (np.sum(np.abs(noise)) / len(noise))
```

Two control parameters are varied systematically:

- *Noise amplitude* $n_{\mathrm{rms}} \in \{0.5,\ 1,\ 1.5,\ 2,\ 2.5,\ 3,\ 4,\ 5\}$;
- *Correlation length* $l_c \in \{\text{None},\ 1,\ 2,\ 3,\ 4\}$ grid points, implemented through a Gaussian filter of width l_c.

The full factorial combination yields $8 \times 5 = 40$ sets of noise statistics. For each pair (n_{rms}, l_c) 100 statistically independent noisy gradients are produced, giving $40 \times 100 = 4\,000$ test cases

After numerical integration the result $F^{\mathrm{num}}(\xi)$ is compared with the analytical reference via the average absolute deviation

$$\mathrm{AAD} = \frac{1}{N} \sum_{i=1}^{N} \left| F_i^{\mathrm{num}} - F_i^{\mathrm{ref}} \right|$$

and the corresponding percentage AAD, $\mathrm{AAD}_{\%} = 100\,\mathrm{AAD}/\Delta F_{\mathrm{range}}$, where $\Delta F_{\mathrm{range}}$ is the range of the reference surface.

The core of the benchmark is sketched below (simplified variable names; see the repository for the full script):

```
for noise_level in noise_levels:
    for corr_len in corr_lengths:
        for _ in range(100):
            dy_noisy = dy_ref + generate_noise(dy_ref.shape, noise_level=noise_level, correla
            error_fft.append(  AAD(FFT_intg_1D(dy_noisy, dx), F_ref) )
            error_simps.append(AAD(intg_1D(dy_noisy, dx), F_ref) )
            error_grad.append( AAD(intgrad1(dy_noisy, dx), F_ref) )
```

Finally, the mean and standard error of each metric are reported over the entire ensemble:

$$\langle \mathrm{AAD} \rangle = \frac{1}{M} \sum_{k=1}^{M} \mathrm{AAD}_k, \qquad \sigma_{\langle \mathrm{AAD} \rangle} = \frac{\sqrt{\frac{1}{M-1} \sum_k (\mathrm{AAD}_k - \langle \mathrm{AAD} \rangle)^2}}{\sqrt{M}},$$

with M equal to the number of replicas. These values constitute the data discussed in Section 3.3.4.

A.2 Estimating the Height in the Exploration Stage

For a two–dimensional collective-variable space, the goal of the exploration stage is identical to that described in Section 5.3.1: rapidly cover every thermodynamically relevant basin up to a predefined free-energy ceiling, F_{lim}, while preventing the walkers from being trapped in local minima. The procedure again relies on an aggressive well-tempered MetaD bias (short Gaussian deposition stride and a bias factor much bigger than F_{lim}), but the estimate of the initial Gaussian height, ω_0, must now account for the *area* that has to be filled.

Assuming that the portion of the FES that needs to be compensated resembles an inverted pyramid of height F_{lim} over the rectangular domain

$$[\mathrm{CV1}_{\mathrm{min}}, \mathrm{CV1}_{\mathrm{max}}] \times [\mathrm{CV2}_{\mathrm{min}}, \mathrm{CV2}_{\mathrm{max}}],$$

the bias “volume” to inject is approximated by

$$A = \frac{1}{3} F_{\mathrm{lim}} (\mathrm{CV1}_{\mathrm{max}} - \mathrm{CV1}_{\mathrm{min}}) (\mathrm{CV2}_{\mathrm{max}} - \mathrm{CV2}_{\mathrm{min}}).$$

A two-dimensional Gaussian kernel of widths $\sigma_{H,x}$ and $\sigma_{H,y}$ contributes a volume $2\pi\sigma_{H,x}\sigma_{H,y}\,\omega$ to the bias. If N_{hills} Gaussians are to be deposited

during the exploration time slice, $t_{\text{exploration}}$, the required height is therefore

$$\omega_0 = \frac{\frac{1}{3} F_{\text{lim}} (\text{CV1}_{\text{max}} - \text{CV1}_{\text{min}})(\text{CV2}_{\text{max}} - \text{CV2}_{\text{min}})}{2\pi \, \sigma_{H,x}\sigma_{H,y} \, \frac{t_{\text{exploration}}}{\tau_G}}, \tag{A.1}$$

where τ_G is the time between two consecutive Gaussian depositions. As in one dimension, a lower bound $\omega \geq F_{\text{lim}}/5$ is imposed to prevent unrealistically small hills.

The exploration trajectory is terminated once the sum of the instantaneous MetaD bias and the current FES estimate exceeds F_{lim} everywhere on the sampled grid. Although the resulting FES will still contain sizeable statistical errors, its qualitative shape suffices for constructing the inverse-force (InvF) bias used in the subsequent refinement stage.

A.3 Gaussian Fitting for MetaD Bandwidth estimation in SRTR and PRTR

In the two–dimensional variants of SRTR and PRTR the local bandwidths $\boldsymbol{\sigma} = (\sigma_x, \sigma_y)$ that control the adaptive reinitialisation kernels are obtained by fitting an Gaussian kernel to each basin of the FES detected on the current surface estimate. The procedure is implemented in `Gaus_fitting_to_fes_2D` and proceeds as follows.

1. Periodic extension (optional).

If either collective variable is periodic, the surface is replicated and concatenated so that basins straddling a boundary are treated as single connected objects. After the fit the indices are mapped back onto the original grid.

2. Pre-processing.

A Gaussian smoothing filter ($\sigma = 3$ grid points by default) removes high-frequency fluctuations to avoid false basin detection. A maximum-filter of size max_filter_size $= n/10$ ($n = \min(N_x, N_y)$) is then applied to the *inverted*

surface, turning minima into peaks. Pixels fulfilling $\tilde{F} = \max(\tilde{F})$ after this operation mark potential basin centres; candidates at the boundary or above $0.95\, F_{\max}$ are discarded.

3. Basin segmentation.

Starting from each remaining centre, a flood-fill algorithm follows the surface gradient "uphill" (four–way connectivity) until no neighbouring grid point has a larger value. The visited points define the footprint of that basin. Basins covering fewer than $1\,\%$ of the total grid area are ignored.

4. Local surface reorientation.

For every accepted basin the local surface is mirrored around its minimum so that the basin floor becomes the peak of an upright hill; values outside the footprint are set to zero. The resulting positive function $B_i(x, y)$ is the target for the Gaussian fit.

5. Gaussian regression.

Along the grid lines crossing the basin centre (x_c, y_c) the function

$$g(x, y) = h\, \exp\!\Big[-\frac{(x - x_c)^2}{2\sigma_x^2}\Big] + h\, \exp\!\Big[-\frac{(y - y_c)^2}{2\sigma_y^2}\Big]$$

is fitted to B_i with `scipy.optimize.curve_fit`. Bounds are:

- $h \in [0,\, B_i^{\max}]$,
- $\sigma_x \in [\Delta x,\, L_x/4]$,
- $\sigma_y \in [\Delta y,\, L_y/4]$,

where $\Delta x, \Delta y$ are the grid spacings and L_x, L_y the box lengths. The parameter triple (h, σ_x, σ_y) is stored for later use by the SRTR and PRTR schedulers; in particular σ_x and σ_y serve as kernel bandwidths, whereas h controls the initial height of the local bias.

6. Output.

The routine returns

- the indices of the basin centres on the original grid;

- a list of the basin masks $B_i(x, y)$;
- the fitted parameter sets $(h, \sigma_x, \sigma_y)_i$.

A one–dimensional analogue, used when a single CV is biased, follows exactly the same logic but operates on line profiles and fits a one–parameter Gaussian $g(x) = h \exp[-(x - x_c)^2/(2\sigma^2)]$.

Illustrative example

Figure A.1 shows two basins extracted from a quartic double-well potential (blue and orange) together with the corresponding Gaussian fits (purple and red). The fitted widths σ_x, σ_y capture the local curvature of each well and are subsequently used to tailor the SRTR/PRTR kernels to the intrinsic length scale of the landscape.

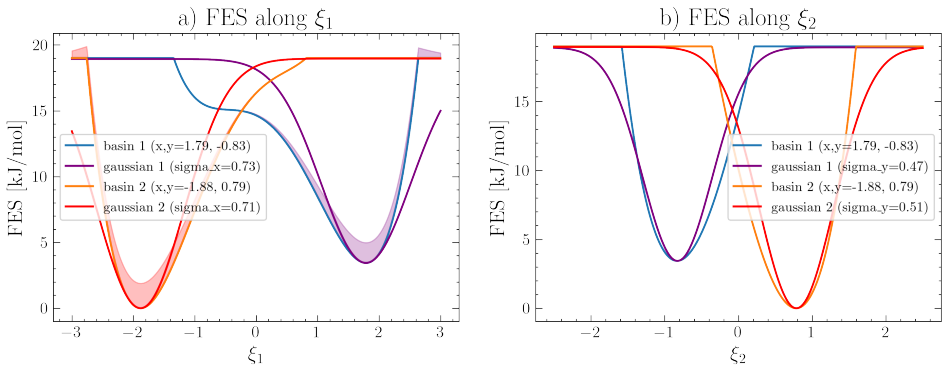


Figure A.1: Gaussian regression on two basins of a synthetic two-dimensional free-energy surface. (a) slices along the x axis ($y = y_c$) and (b) slices along the y axis ($x = x_c$). The blue and orange line represent the identified basins, and the purple and red line the Gaussian fits. Legends report the basin centres and the fitted standard deviations.

A.4 Definition of One-Dimensional Surfaces

In Section 5.8 , the reinitialisation protocols introduced in Sections 5.4–5.6.3 were benchmarked against twenty analytical model potentials. They were selected to span a broad spectrum of topographies, ranging from nearly smooth double wells to highly corrugated landscapes with multiple competing minima. This appendix gathers the explicit functional forms of those test

surfaces and the associated error progressions, thereby keeping the main text uncluttered.

Figure A.2 visualises the 20 one-dimensional potentials employed. Moving from left to right, the surfaces become progressively rougher, whereas moving from top to bottom, the central barrier height increases. All potentials are dimensionless and are expressed as functions of the collective coordinate ξ.

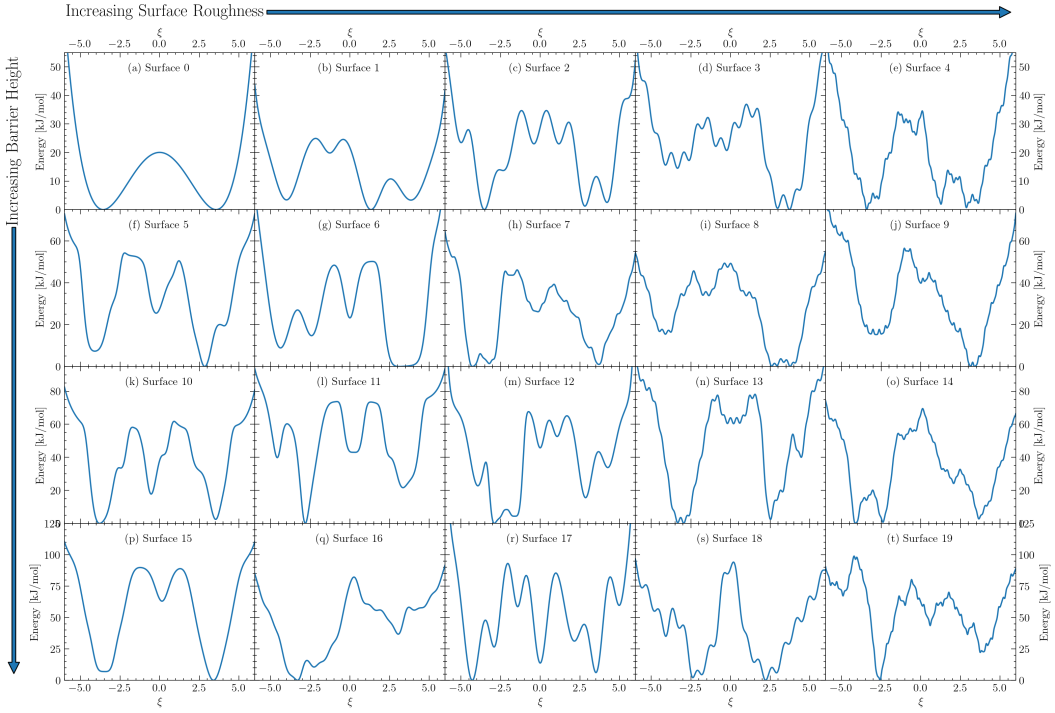


Figure A.2: The figure shows a range of one-dimensional analytical potential surfaces. Moving from left to right, the surface roughness increases, while moving from top to bottom, the energy barrier height rises.

For completeness, the mathematical expressions of the twenty potentials are reported below. They are labelled *Surface 0* – *Surface 19* in the same order as Figure A.2.

Surface 0:

$$U_0(\xi) = \frac{2}{3}\left(3(0.5\xi)^4 - 19(0.5\xi)^2\right) . \tag{A.2}$$

Surface 1:

$$U_1(\xi) = \frac{5}{6}\Big(-30e^{-0.33(1.5\xi+6.0)^2} - 10e^{-(1.5\xi+1.9)^2} - 32.5e^{-0.4(1.5\xi-1.9)^2} - 30e^{-0.2(1.5\xi-5.8)^2} + e^{-3\xi-15} + e^{3\xi-15}\Big) . \tag{A.3}$$

Surface 2:

$$U_2(\xi) = \frac{7}{6}\Big(-20e^{-0.25(\xi+3.5)^4} - 20e^{-0.25(\xi-3.5)^4} + 5\sin(4\xi) + 10e^{-5(\xi+4.5)^2} + e^{-2\xi-9} + e^{2\xi-8.8}\Big) . \tag{A.4}$$

Surface 3:

$$U_3(\xi) = \frac{4}{3}\Big(-14e^{-0.25(\xi+3.5)^4} - 25e^{-0.25(\xi-3.5)^4} - 10e^{-(\xi+0.5)^2} - 2\sin(-8\xi) + e^{-2\xi-9} + e^{2\xi-9}\Big) . \tag{A.5}$$

Surface 4:

$$U_4(\xi) = 1.5\Big(-22e^{-1.6(\xi-1)^2} - 10e^{-4.0(\xi+0.5)^2} - 30e^{-0.332(\xi-3.2)^2} - 30e^{-0.344(\xi+3.2)^2} + 2\sin\big(-8(\xi-1.1)\big)^3\Big) + e^{-2\xi-9} + e^{2\xi-9} . \tag{A.6}$$

Surface 5:

$$\begin{aligned} U_5(\xi) = 0.89\Big(&22e^{-4.7(\xi-3.6)^2} + 11e^{-2.0(\xi-0.09)^2} + 29e^{-3.8(\xi+3.0)^2} \\ &- 11e^{-5.0(\xi-3.2)^4} - 28e^{-2.2(\xi+0.13)^2} - 6e^{-4.4(\xi+3.1)^2} \\ &- 12.4e^{-1.6(\xi-0.14)^6} - 51e^{-0.21(\xi-3.2)^4} - 52e^{-0.361(\xi+3.7)^6}\Big) \\ &+ e^{-2\xi-9} + e^{2\xi-9} . \end{aligned} \tag{A.7}$$

Surface 6:

$$U_6(\xi) = 1.8\Big(-28e^{-0.1(\xi+3.5)^4} + 15e^{-1.5(\xi+3.3)^2} - 15e^{-3.5\xi^2} - 28e^{-0.1(\xi-3.5)^6}\Big) + e^{-2\xi-9} + e^{2\xi-8.8} . \tag{A.8}$$

Surface 7:

$$\begin{aligned} U_7(\xi) =& 0.88\Big(12e^{-2.2(\xi-3.0)^2} + e^{-4.8(\xi+0.02)^4} + 18e^{-3.0(\xi+3.5)^2} \\ & - 17e^{-2.5(\xi-3.3)^4} - 19e^{-3.9(\xi+0.29)^4} - 11e^{-3.0(\xi+3.4)^2} \\ & - 37e^{-0.208(\xi-3.6)^2} - 53e^{-0.386(\xi+3.7)^6} + 3\sin\big(-4(\xi-1.3)\big)^6\Big) \\ & + e^{-2\xi-9} + e^{2\xi-9} \,. \end{aligned} \tag{A.9}$$

Surface 8:

$$\begin{aligned} U_8(\xi) =& 0.65\Big(9e^{-2.9(\xi-3.3)^2} + 21e^{-3.3(\xi+0.21)^4} + 28e^{-3.5(\xi+2.8)^4} \\ & - 58e^{-0.278(\xi-3.3)^4} - 29e^{-0.188(\xi+3.7)^6} + 3\sin\big(-8(\xi-1.0)\big)^2\Big) \\ & + e^{-2\xi-9} + e^{2\xi-9} \,. \end{aligned} \tag{A.10}$$

Surface 9:

$$\begin{aligned} U_9(\xi) =& 0.83\Big(5e^{-4.6(\xi-2.7)^2} + 3e^{-3.3(\xi-0.22)^2} + 21e^{-3.6(\xi+2.9)^4} \\ & - 15e^{-4.9(\xi-3.4)^4} - 16e^{-4.3(\xi-0.0)^2} - 21e^{-4.0(\xi+2.9)^4} \\ & - 59e^{-0.197(\xi-3.0)^2} - 53e^{-0.274(\xi+2.8)^4} + 3\sin\big(-8(\xi-1.9)\big)^6\Big) \\ & + e^{-2\xi-9} + e^{2\xi-9} \,. \end{aligned} \tag{A.11}$$

Surface 10:

$$\begin{aligned} U_{10}(\xi) =& 1.3\Big(11e^{-3.3(\xi-2.8)^4} + 21e^{-4.1(\xi-0.13)^2} + 16e^{-4.3(\xi+2.7)^2} \\ & - 20e^{-4.4(\xi-3.4)^2} - 28e^{-3.4(\xi+0.13)^4} - 8e^{-3.8(\xi+3.5)^4} \\ & - 10e^{-1.2(\xi+0.26)^6} - 32e^{-0.3(\xi-3.2)^6} - 38e^{-0.304(\xi+3.3)^6} \\ & + 2\sin\big(-2(\xi-1.1)\big)\Big) + e^{-2\xi-9} + e^{2\xi-9} \,. \end{aligned} \tag{A.12}$$

Surface 11:

$$\begin{aligned} U_{11}(\xi) =& 1.53\Big(21e^{-4.3(\xi-2.7)^2} + e^{-3.6(\xi-0.04)^4} + 29e^{-3.5(\xi+3.7)^4} \\ & - 15e^{-2.3(\xi-2.8)^2} - 21e^{-4.8(\xi-0.23)^4} - 17e^{-4.9(\xi+3.0)^2} \\ & - 30e^{-0.366(\xi-3.3)^6} - 38e^{-0.187(\xi+3.4)^4}\Big) + e^{-2\xi-9} + e^{2\xi-9} \ . \end{aligned} \tag{A.13}$$

Surface 12:

$$\begin{aligned} U_{12}(\xi) =& 2.3\Big(-20e^{-0.25(2.8\xi+11)^2} - 28e^{-0.15(2.8\xi+7.5)^4} - 28e^{-0.15(2.8\xi+4.5)^4} \\ & - 10e^{-0.8(2.8\xi)^2} - 7e^{-0.8(2.8\xi-3.0)^2} - 23e^{-0.3(2.8\xi-8.0)^2} \\ & - 15e^{-0.3(2.8\xi-12)^2}\Big) + e^{-5.6\xi-30} + e^{5.6\xi-30} \ . \end{aligned} \tag{A.14}$$

Surface 13:

$$\begin{aligned} U_{13}(\xi) =& 1.81\Big(29e^{-1.4(\xi-3.4)^4} + 5e^{-3.8(\xi-0.07)^4} + 0e^{-3.8(\xi+2.9)^4} \\ & - 22e^{-2.5(\xi-2.9)^4} - 5e^{-3.7(\xi-0.05)^4} - 12e^{-3.2(\xi+3.1)^4} \\ & - 8.6e^{-2.3(\xi-0.22)^6} - 40e^{-0.35(\xi-3.4)^4} - 30e^{-0.212(\xi+3.0)^4} \\ & + 2\sin\big(-8(\xi-0.2)\big)^2\Big) + e^{-2\xi-9} + e^{2\xi-9} \ . \end{aligned} \tag{A.15}$$

Surface 14:

$$\begin{aligned} U_{14}(\xi) =& 1.54\Big(22e^{-1.6(\xi-2.8)^2} + 10e^{-4.0(\xi-0.19)^2} + 23e^{-3.2(\xi+3.2)^4} \\ & - 41e^{-0.332(\xi-3.2)^2} - 47e^{-0.344(\xi+3.2)^4} + 2\sin\big(-8(\xi-1.1)\big)^3\Big) \\ & + e^{-2\xi-9} + e^{2\xi-9} \ . \end{aligned} \tag{A.16}$$

Surface 15:

$$\begin{aligned} U_{15}(\xi) =& 1.39\Big(-25e^{-1.9(\xi-3.4)^2} - 12e^{-3.2(\xi-0.17)^2} - 16e^{-3.7(\xi+3.5)^4} \\ & - 7.6e^{-2.2(\xi-0.11)^2} - 40e^{-0.24(\xi-3.4)^4} - 44e^{-0.227(\xi+3.5)^4}\Big) \\ & + e^{-2\xi-9} + e^{2\xi-9} \ . \end{aligned} \tag{A.17}$$

Surface 16:

$$\begin{aligned} U_{16}(\xi) =& 1.2\Big(10e^{-4.1(\xi-3.6)^2} + 28e^{-3.3(\xi-0.13)^2} + 6e^{-4.6(\xi+2.7)^2} \\ & - 12e^{-4.2(\xi-3.1)^2} - 3e^{-4.4(\xi-0.07)^2} - 13e^{-4.7(\xi+3.7)^4} \\ & - 23e^{-0.227(\xi-3.3)^2} - 57e^{-0.155(\xi+2.6)^2} + 2\sin\big(-4(\xi-1.6)\big)^2\Big) \\ & + e^{-2\xi-9} + e^{2\xi-9} \,. \end{aligned} \tag{A.18}$$

Surface 17:

$$\begin{aligned} U_{17}(\xi) =& 3.5\Big(-15e^{-0.45(2.8\xi-14.0)^2} - 28e^{-0.35(2.8\xi-10.0)^2} - 23e^{-0.15(2.8\xi-6.0)^2} \\ & - 28e^{-0.3(2.8\xi)^2} - 15e^{-0.6(2.8\xi+4.0)^2} - 23e^{-0.4(2.8\xi+8.0)^2} \\ & - 32e^{-0.2(2.8\xi+12.0)^2}\Big) + e^{-5.6\xi-30} + e^{5.6\xi-30} \,. \end{aligned} \tag{A.19}$$

Surface 18:

$$\begin{aligned} U_{18}(\xi) =& 1.69\Big(5e^{-1.6(\xi-3.6)^2} + 26e^{-4.4(\xi+0.01)^4} + 21e^{-4.6(\xi+3.2)^4} \\ & - 42e^{-0.249(\xi-2.5)^2} - 44e^{-0.281(\xi+2.5)^2} + 3\sin\big(-8(\xi-2.0)\big)\Big) \\ & + e^{-2\xi-9} + e^{2\xi-9} \,. \end{aligned} \tag{A.20}$$

Surface 19:

$$\begin{aligned} U_{19}(\xi) =& 1.46\Big(21e^{-4.3(\xi-3.4)^2} + 15e^{-4.5(\xi+0.14)^4} + 28e^{-3.7(\xi+3.7)^4} \\ & - 20e^{-2.4(\xi-3.6)^2} - 15e^{-3.6(\xi-0.3)^2} - 24e^{-3.4(\xi+2.7)^2} \\ & - 8.9e^{-0.5(\xi-0.06)^2} - 23e^{-0.3(\xi-3.7)^4} - 22e^{-0.337(\xi+2.8)^6} \\ & + 3\sin\big(-8(\xi-0.4)\big)^7\Big) + e^{-2\xi-9} + e^{2\xi-9} \,. \end{aligned} \tag{A.21}$$

A.5 Reinitialisation Results for One-Dimensional Surfaces

Section 5.8 compares five reinitialisation protocols on the 20 analytical potentials defined in Section A.4. The protocols are:

- Long-intuitive simulation: A single simulation is run with a MetaD bias using an initial Gaussian height of 5 k/mol and a bias factor of 50 for every surface.
- Long-adjusted simulation: A single simulation is run with a MetaD bias that is adjusted to the height of the largest energy barrier (parameters are provided in each figure).
- SRTR (Serial Real-Time Reinitialisation) campaigns: A simulation starts with a big initial Gaussian height and a bias factor. Automated rules are used to terminate and reinitialise the simulation repeatedly with a more conservative MetaD bias.
- PRTR (Parallel Real-Time Reinitialisation) campaigns: Similar to SRTR, but with four simulations running in parallel, and their results are combined, producing the results of the campaign.
- Two-step approach: A simpler version of SRTR. A short exploration simulation with a agressive MetaD bias is followed by a longer, more conservative MetaD simulation.

Each figure reports the ensemble-averaged error as a function of the accumulated simulation steps (log–linear axes).

- Panel **a** displays the average magnitude of the instantaneous mean-force error.
- Panel **b** shows the average absolute deviation of the reconstructed FES.
- Panel **c** gives the bootstrap estimate of the FES error.

Curves are coloured as in the legend: blue dotted (long-intuitive), blue solid (long-adjusted), red (PRTR), orange (SRTR), and green (two-step protocol; stage 1 and stage 2). Parameter values for the Gaussian hill height H and bias factor BF are listed in the legend of each plot, except the dynamically reinitialised SRTR and PRTR protocols..

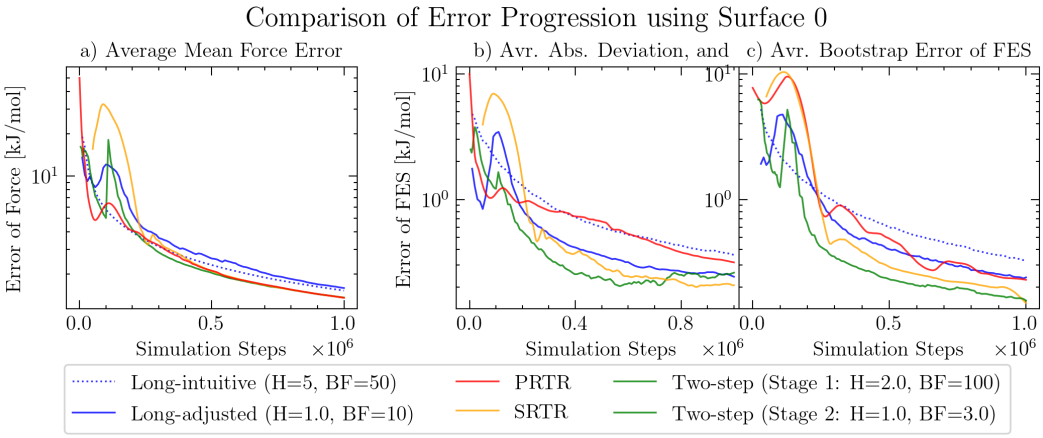


Figure A.3: Convergence behaviour on Surface 0. (a) average mean-force error, (b) average absolute deviation of the FES, and (c) bootstrap estimate of the FES error. All curves represent averages over 100 independent trajectories. Legends report the Gaussian hill height H and bias factor BF used by each protocol, except the dynamically reinitialised SRTR and PRTR protocols.

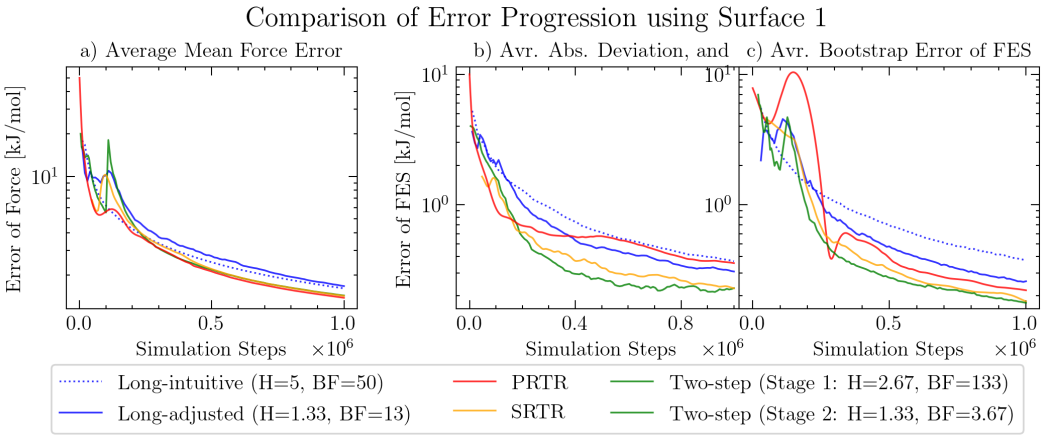


Figure A.4: Convergence behaviour on Surface 1. See the caption of Figure A.3 for a description of the panels and colour coding.

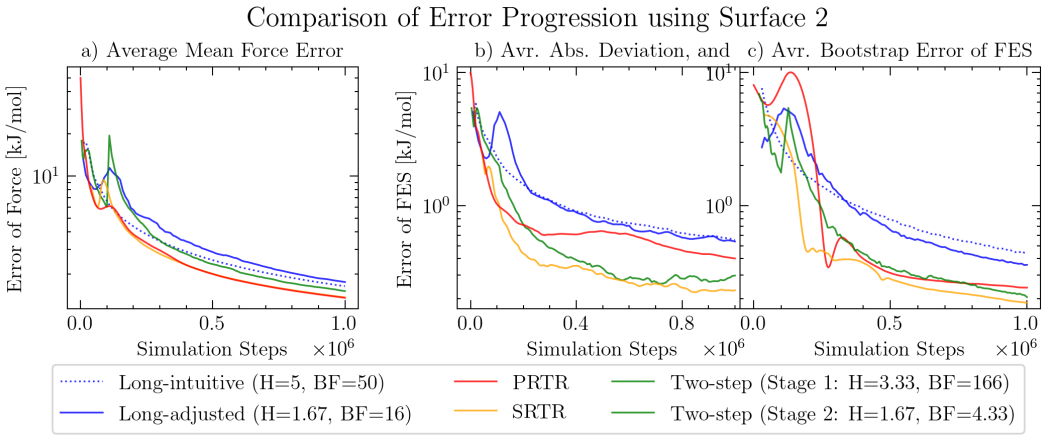


Figure A.5: Convergence behaviour on Surface 2. See the caption of Figure A.3 for a description of the panels and colour coding.

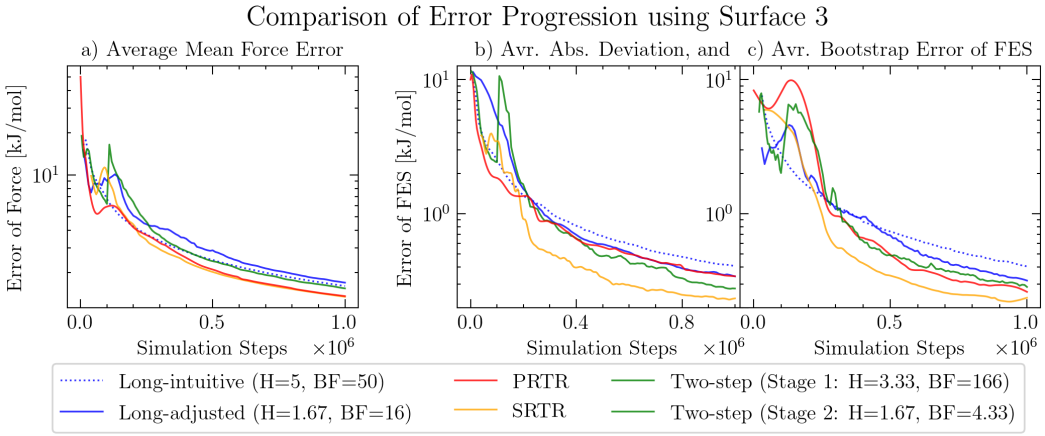


Figure A.6: Convergence behaviour on Surface 3. Panels and colour code as in Figure A.3.

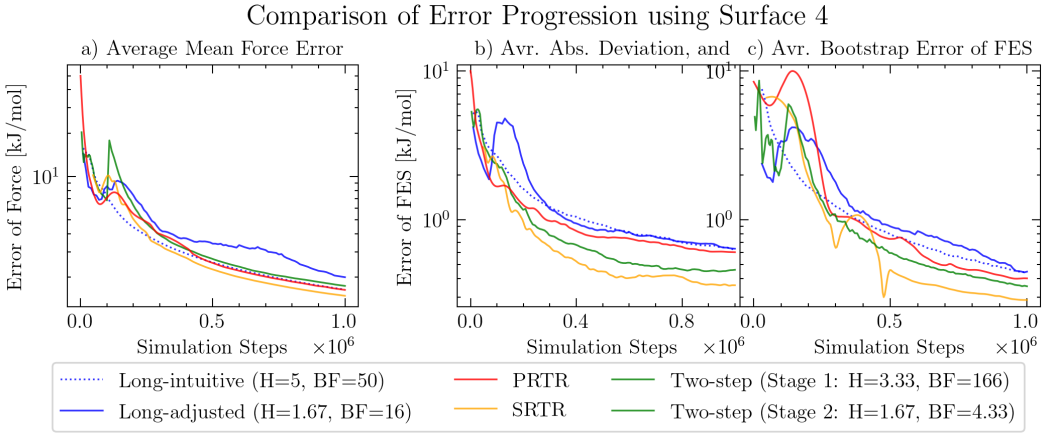


Figure A.7: Convergence behaviour on Surface 4. See Figure A.3 for details.

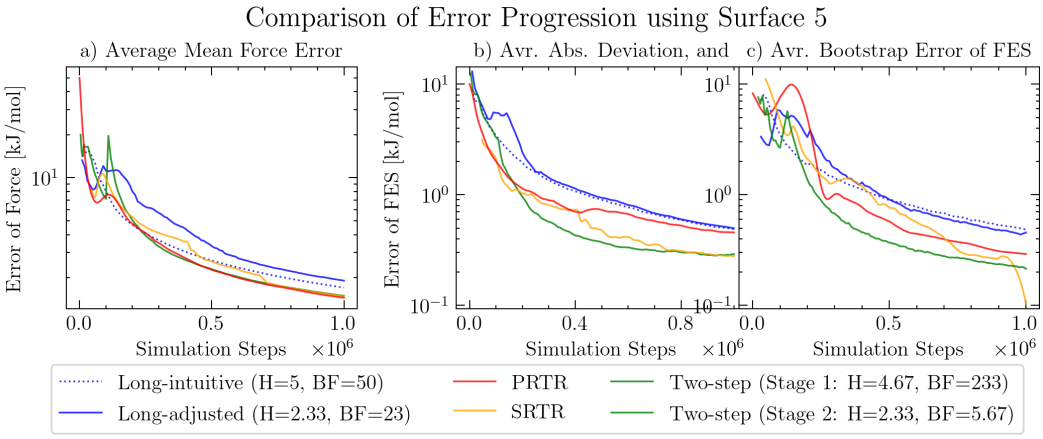


Figure A.8: Convergence behaviour on Surface 5. Panels and colour code as in Figure A.3.

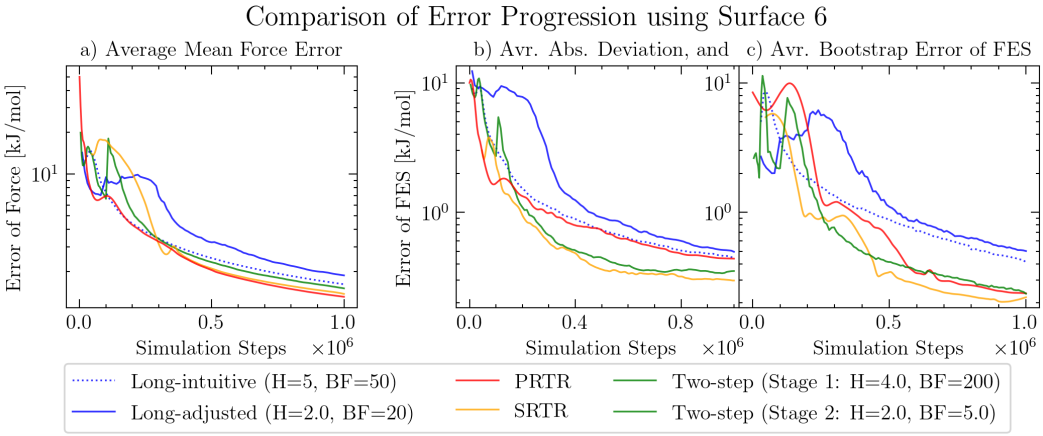


Figure A.9: Convergence behaviour on Surface 6. See Figure A.3 for details.

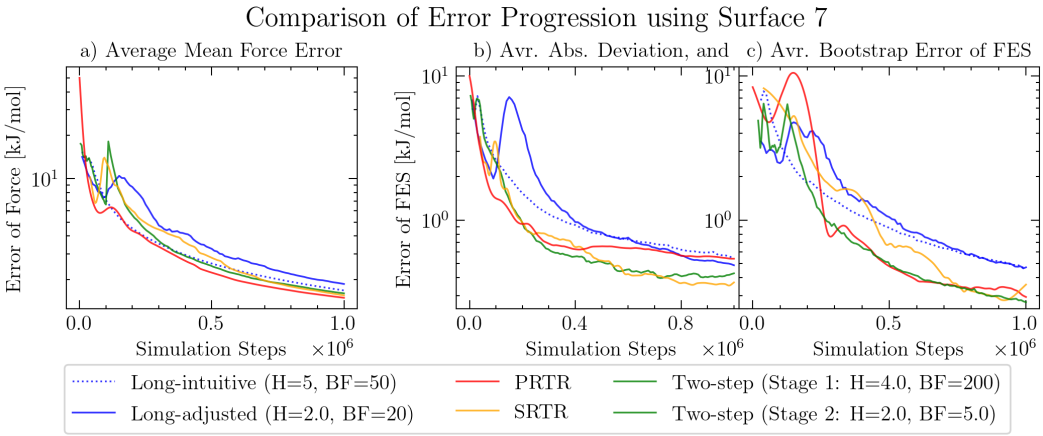


Figure A.10: Convergence behaviour on Surface 7. Panels and colour code as in Figure A.3.

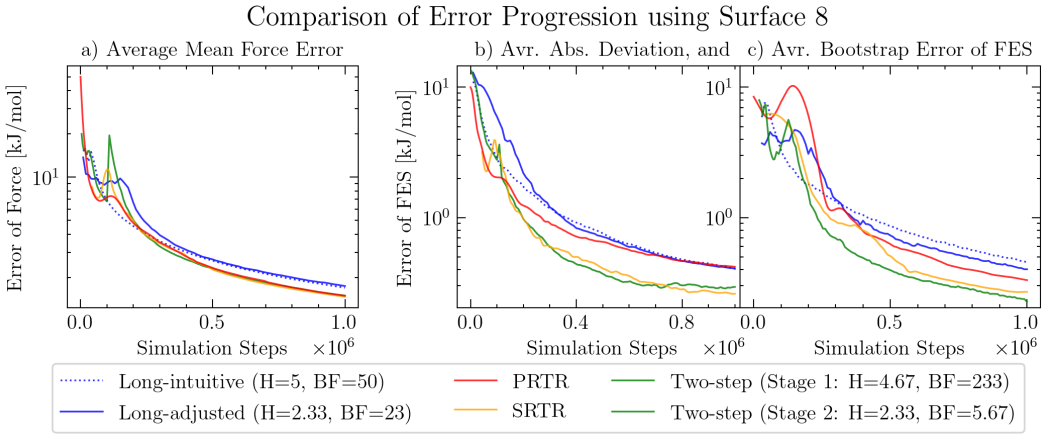


Figure A.11: Convergence behaviour on Surface 8. See Figure A.3 for details.

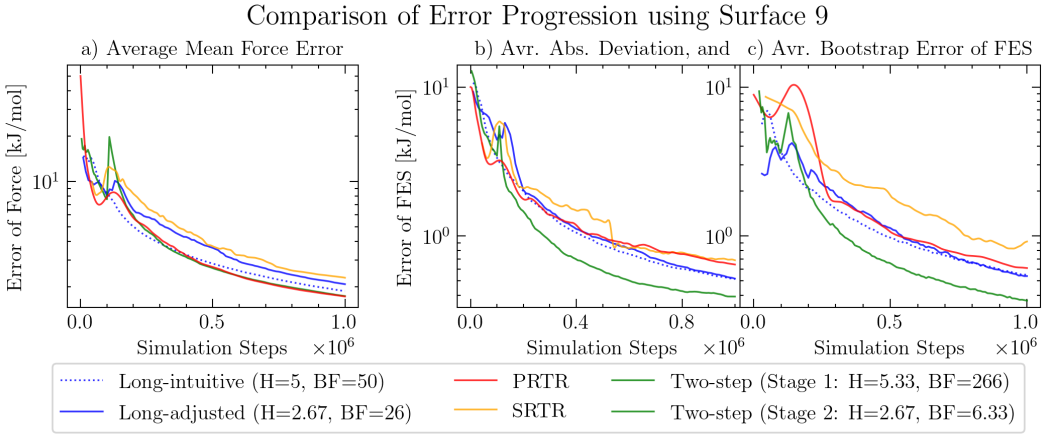


Figure A.12: Convergence behaviour on Surface 9. Panels and colour code as in Figure A.3.

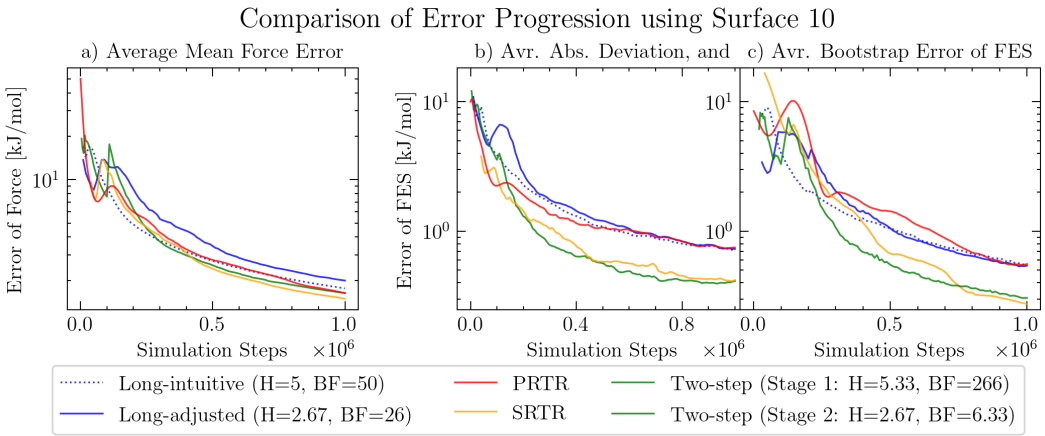


Figure A.13: Convergence behaviour on Surface 10. See Figure A.3 for details.

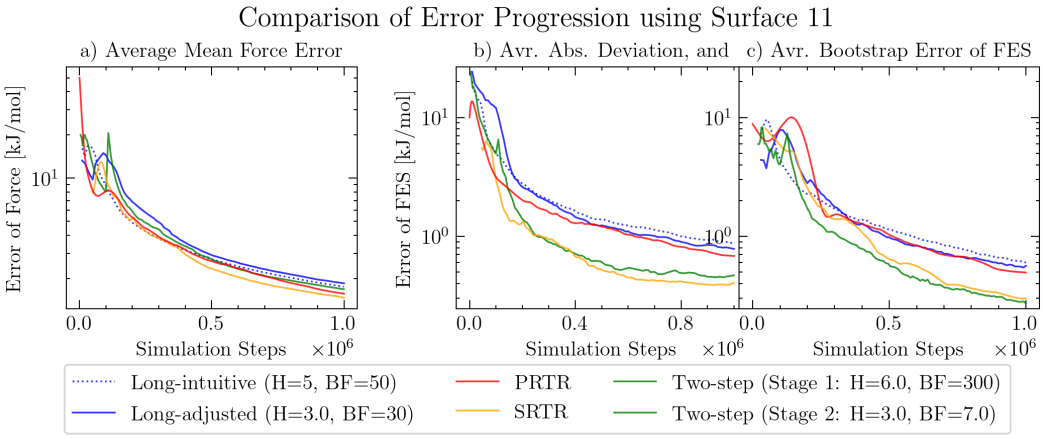


Figure A.14: Convergence behaviour on Surface 11. Panels and colour code as in Figure A.3.

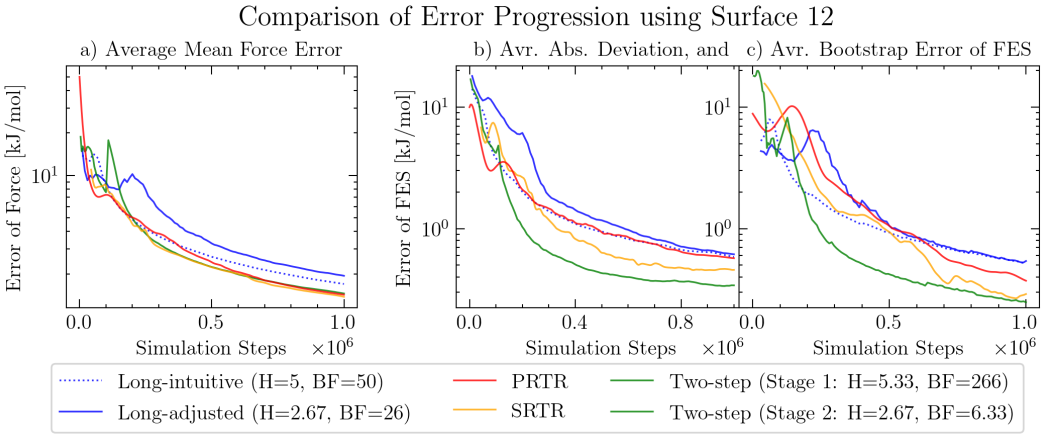


Figure A.15: Convergence behaviour on Surface 12. See Figure A.3 for details.

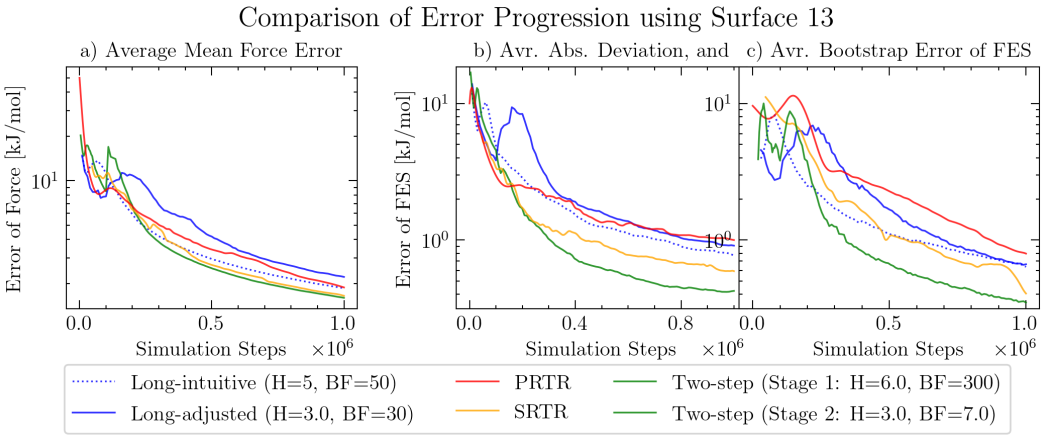


Figure A.16: Convergence behaviour on Surface 13. Panels and colour code as in Figure A.3.

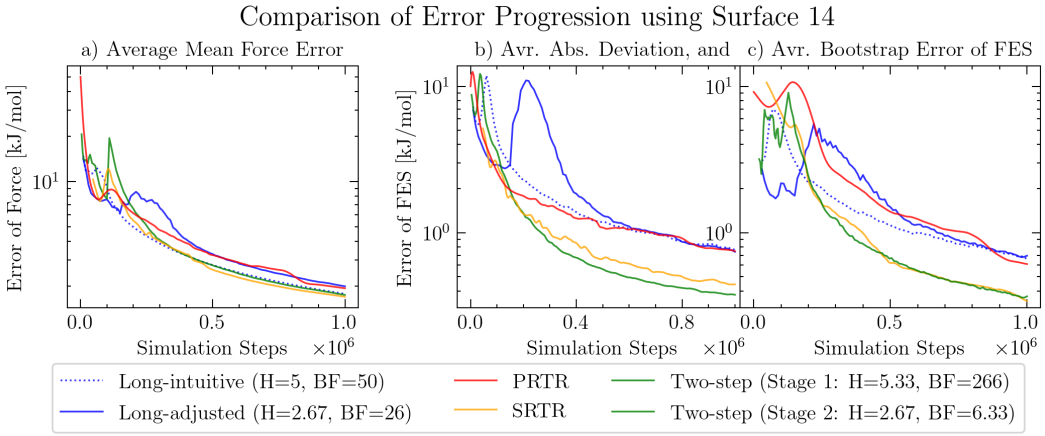


Figure A.17: Convergence behaviour on Surface 14. See Figure A.3 for details.

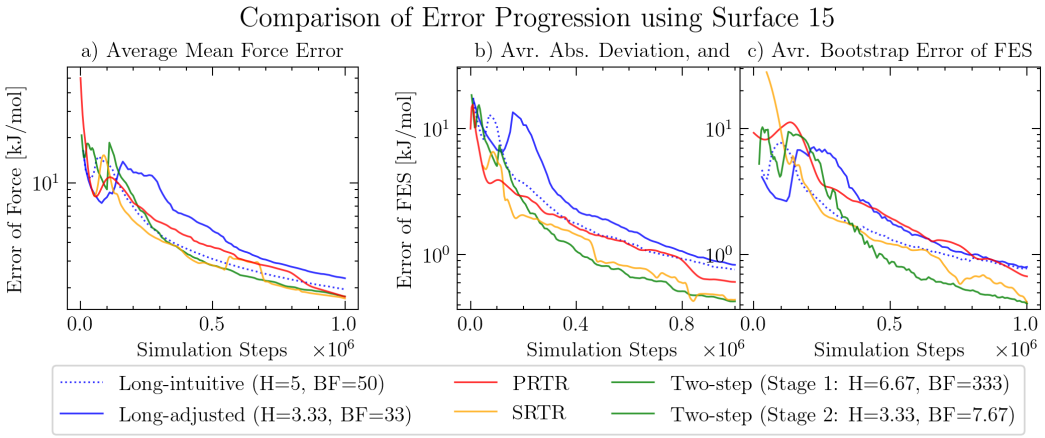


Figure A.18: Convergence behaviour on Surface 15. Panels and colour code as in Figure A.3.

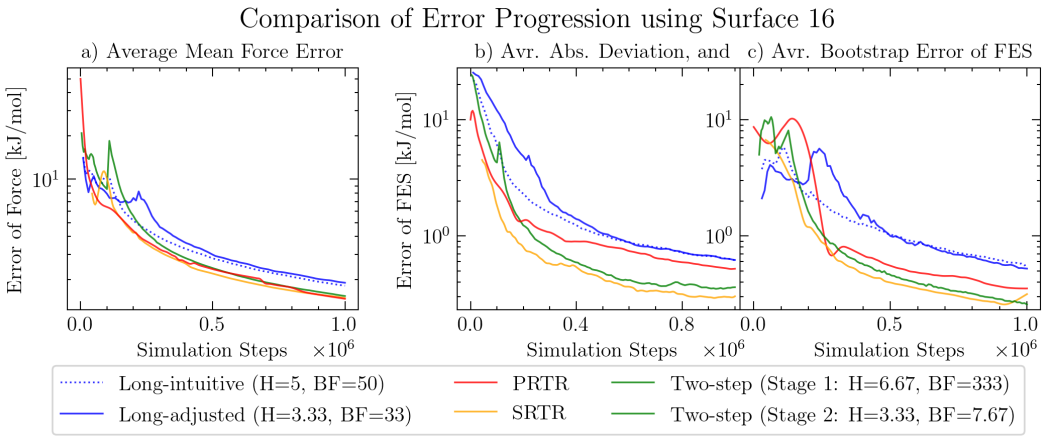


Figure A.19: Convergence behaviour on Surface 16. See Figure A.3 for details.

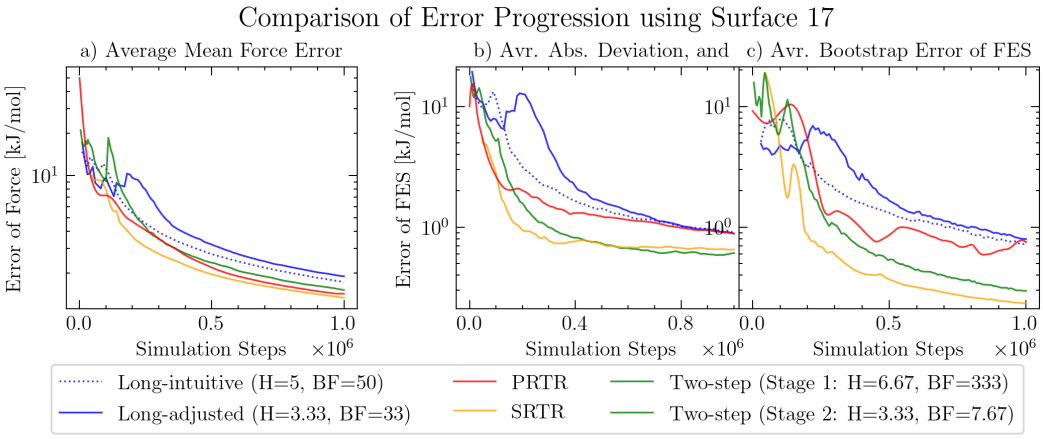


Figure A.20: Convergence behaviour on Surface 17. Panels and colour code as in Figure A.3.

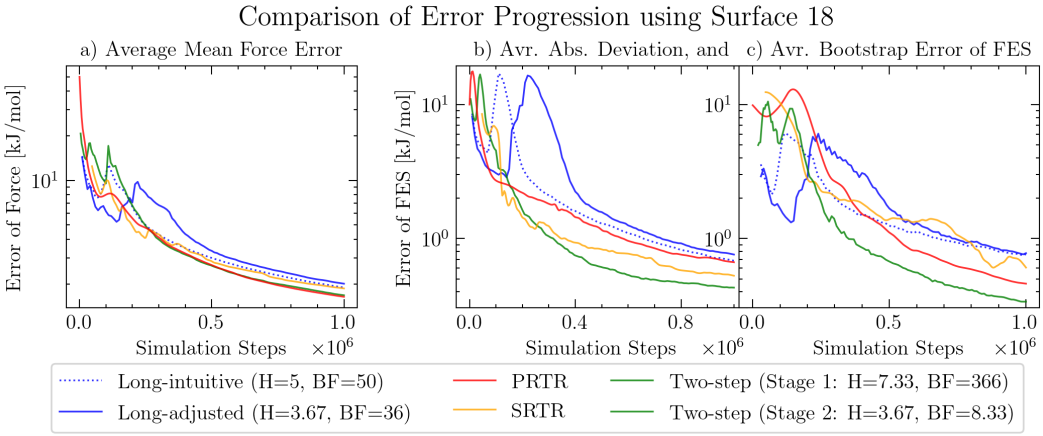


Figure A.21: Convergence behaviour on Surface 18. See Figure A.3 for details.

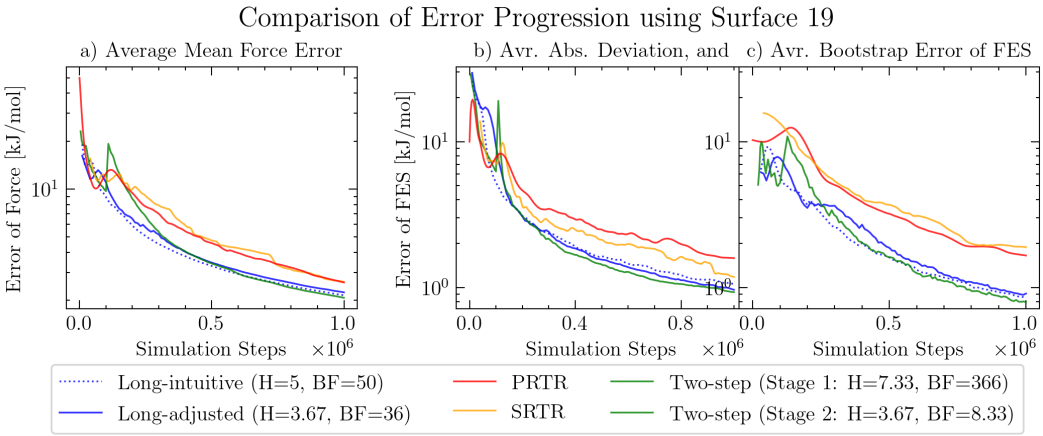


Figure A.22: Convergence behaviour on Surface 19. Panels and colour code as in Figure A.3.

A.6 Definition of Two-Dimensional Surfaces

Four analytical two-dimensional potentials are considered: the Invernizzi benchmark surface [76] and three custom double-well landscapes of increasing ruggedness (labelled *Potential 1–3*). Figure A.23 provides a qualitative overview; explicit functional forms are listed below for reproducibility.

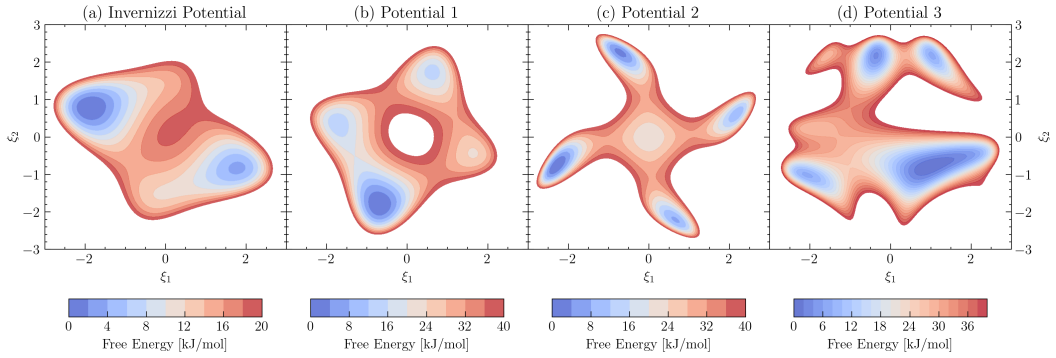


Figure A.23: Four two-dimensional analytical potential surfaces, increasing in complexity from left to right.

Invernizzi Potential:

$$\begin{aligned} U_{\text{Inve}}(\xi_1,\xi_2) = {} & 1.34549\,\xi_1^4 + 1.90211\,\xi_1^3\xi_2 + 3.92705\,\xi_1^2\xi_2^2 - 6.44246\,\xi_1^2 \\ & - 1.90211\,\xi_1\xi_2^3 + 5.58721\,\xi_1\xi_2 + 1.33481\,\xi_1 + 1.34549\,\xi_2^4 \\ & - 5.55754\,\xi_2^2 + 0.904586\,\xi_2 + 18.5598\ . \end{aligned} \tag{A.22}$$

Potential 1:

$$\begin{aligned} U_{\text{Pot1}}(\xi_1,\xi_2) = {} & 3.9\,\xi_1^4 + 3.9\,\xi_2^4 - 19.2\,\xi_1^2 - 19.2\,\xi_2^2 + 5.7\,\xi_1^3\xi_2 \\ & - 5.7\,\xi_1\xi_2^3 + 12\,\xi_1^2\xi_2^2 - 6\,\xi_1\xi_2 + 2.7\,\xi_2 + 3.6\,\xi_1\ . \end{aligned} \tag{A.23}$$

Potential 2:

$$\begin{aligned} U_{\text{Pot2}}(\xi_1,\xi_2) = {} & 1.84\,\xi_1^6 + 1.84\,\xi_2^6 - 11.5\,\xi_1^4 - 11.5\,\xi_2^4 + 17.94\,\xi_1^2 + 17.94\,\xi_2^2 \\ & - 2.3\,\xi_1^5\xi_2 + 2.3\,\xi_1\xi_2^5 - 3.7\,\xi_1^3\xi_2 + 3.7\,\xi_1\xi_2^3 + 20\,\xi_1^2\xi_2^2 \\ & + 2.1\,\xi_1^2 + 2.1\,\xi_2^2 + \xi_1^2\xi_2 + \xi_1\xi_2^2 + 24.70075155\ . \end{aligned} \tag{A.24}$$

Potential 3:

$$U_{\text{Pot3}}(\xi_1,\xi_2) = \left(e^{\xi_1-2}\right)^5 + \left(e^{-\xi_1-2}\right)^5 + \left(e^{\xi_2-2}\right)^5 + \left(e^{-\xi_2-2}\right)^5 + 13\,\sin\!\left(2\,\xi_1\xi_2\right) + 17\,\sin\!\left(2\,\xi_2\right) + 0.137\,\xi_1^4\xi_2^4\ . \tag{A.25}$$

A.7 Reinitialisation Results for Two-Dimensional Surfaces

Section 5.8 benchmarked four simulation protocols—the long reference run, PRTR, SRTR, and the two-step approach—on each two-dimensional surface. Every protocol was repeated 100 times, and three error metrics were monitored during the trajectory:

(a) average mean-force error,

(b) average absolute deviation of the FES,

(c) bootstrap estimate of the FES error.

The ensemble-averaged progressions are plotted below. Curves are coloured consistently with the one-dimensional study: blue (long), red (PRTR), orange (SRTR), and green (two-step).

Invernizzi Potential:

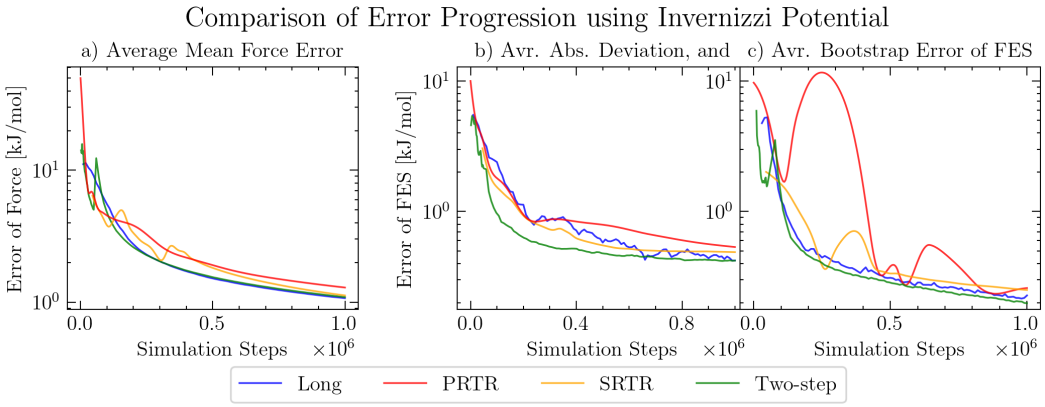


Figure A.24: Error progression on the Invernizzi surface: (a) average mean-force error, (b) average absolute deviation of the FES, and (c) bootstrap estimate of the FES error. Each curve is averaged over 100 trajectories; colours denote the long (blue), PRTR (red), SRTR (orange), and two-step (green) protocols.

Potential 1

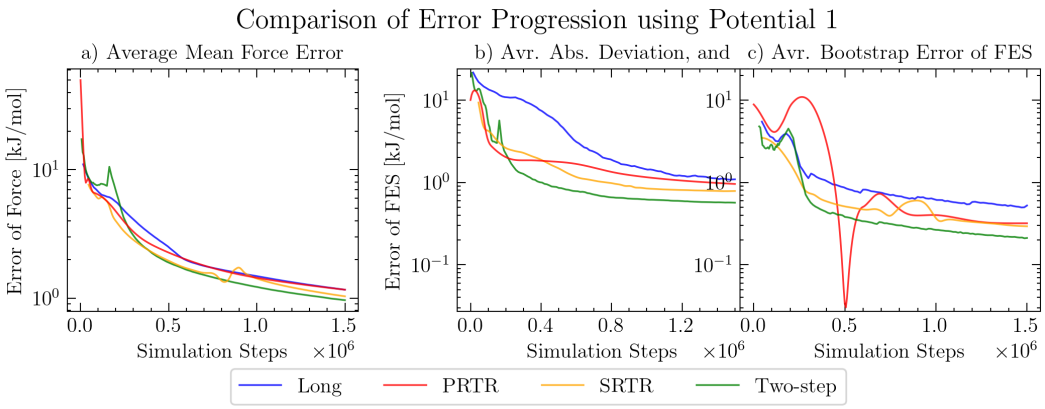


Figure A.25: Error progression on Potential 1. Panels and colour code as in Figure A.24.

Potential 2

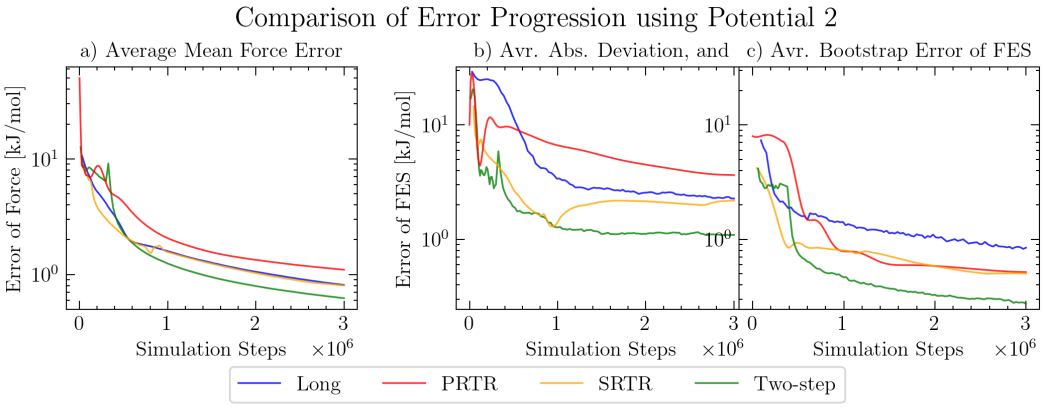


Figure A.26: Error progression on Potential 2. Panels and colour code as in Figure A.24.

Potential 3

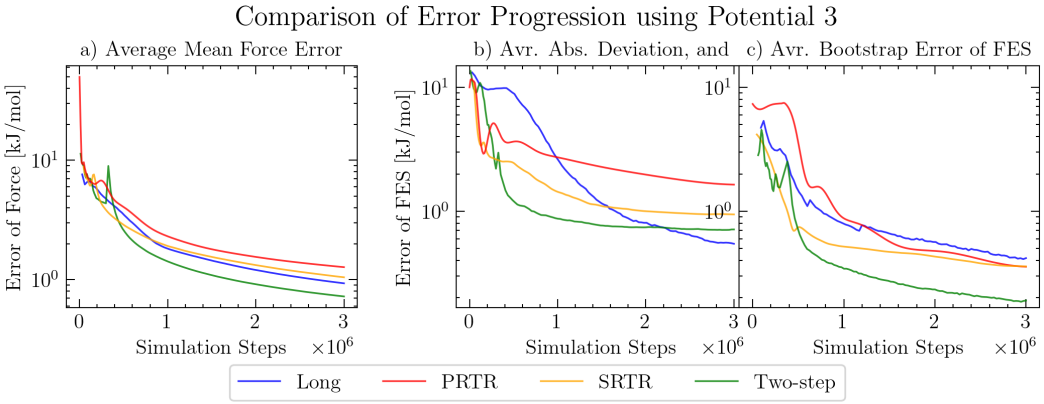


Figure A.27: Error progression on Potential 3. Panels and colour code as in Figure A.24.

Bootstrap error diagnostics: The bootstrap estimate in panel (c) occasionally displays pronounced oscillations for the dynamically reinitialised PRTR and SRTR simulations. Nevertheless, it provides a reliable indication of the absolute FES error, especially for the long reference trajectories. The discrepancy for the reinitialised schemes is most likely attributed to the current bootstrap implementation rather than an inherent metric limitation. Therefore, it does not compromise the error analysis presented in the applications of Chapter 4 and Chapter 6.

Appendix B

Python Code

The algorithms developed for this thesis are distributed under an open-source licence and are actively maintained at `https://github.com/mme-ucl`. Two repositories are relevant:

- `pyMFI` — a lightweight, function-oriented library that provides the core MFI routines for one- and two-dimensional collective variables (`https://github.com/mme-ucl/pyMFI`);
- `MFI` — a research repository that contains the most recent code, object-oriented wrappers for automated reinitialisation, and Jupyter tutorials (`https://github.com/mme-ucl/MFI`).

The repository at `github.com/mme-ucl/MFI/tree/master/pyMFI` tracks the current development branch of the core library. The higher-level classes for one- and two-dimensional workflows for the reinitialisation methods reside in `github.com/mme-ucl/MFI/tree/master/MFI_class1D` and `github.com/mme-ucl/MFI/tree/master/MFI_class1D`, respectively. All scripts interface with the community-developed PLUMED package [94]; running new MD simulations further requires a patched build of PLUMED and GROMACS [71]. Comprehensive installation instructions are provided in the PLUMED user manual. The analysis routines alone can be executed without any external MD engine. Langevin dynamics simulations of analytical potentials can be executed with PLUMED.

B.1 pyMFI: Main Functionalities

B.1.1 One-dimensional workflow

The code below illustrates a complete one-dimensional workflow: generation of a Langevin trajectory on an analytical surface, calculation of the time-independent mean force with MFI1D, numerical integration to obtain the FES, and visualisation of the results.

```
# Import libraries
from pyMFI import MFI1D
from pyMFI import run_plumed

# Define surface
X = np.linspace(-2, 2, 401)
y_string = "1*x^8-50*exp(-(x-1)^2/0.1)-93*exp(-(x+0.5)^2/0.05)
 ↪  -103*exp(-(x+1.5)^2/0.07)-70*exp(-(x)^2/0.03)
 ↪  +50*exp(-(x-0.5)^2/0.03)"
y_ref = y=1*X**8-50*np.exp(-(X-1)**2/0.1)-93*np.exp(-(X+0.5)**2/0.0
 ↪  5)-103*np.exp(-(X+1.5)**2/0.07)-70*np.exp(-(X)**2/0.03)+50*np.e
 ↪  xp(-(X-0.5)**2/0.03)
y_ref = y_ref - min(y_ref)

#Simulation parameters
simulation_parameters = {"analytical_function":y_string,
 ↪  "simulation_steps":2_000_000, "temperature":1, "gaus_width":0.1,
 ↪  "gaus_height":5, biasfactor":30, "gaus_pace":100}

#Run Metadynamics simulation
run_plumed.run_langevin1D(**simulation_parameters)

#Read the HILLS and COLVAR file
HILLS = MFI1D.load_HILLS(hills_name="HILLS")
position = MFI1D.load_position(position_name="position")
```

```
20
21 # MFI parameters
22 MFI_parameters = {"HILLS":HILLS, "position":position, "bw":0.03,
   ↪ "min_grid":-2, "max_grid":2, "nbins":401,
   ↪ "use_weighted_st_dev":False}
23
24 #Compute the time-independent mean force
25 results = MFI1D.MFI_1D(**MFI_parameters)
26 X, Ftot_den, Ftot_den2, Ftot, ofv_num, FES, ofv, ofe, cutoff,
   ↪ error_evol, fes_error_cutoff_evol = results
27
28 #integration on a non-periodic domain
29 FES = MFI1D.intg_1D(Ftot, X[1]-X[0])
30
31 #Plot the results
32 MFI1D.plot_recap(X, FES, Ftot_den, ofe, error_evol[1],
   ↪ error_evol[3], y_ref, FES_lim=200)
```

B.1.2 Two-Dimensional Workflow

The next code box demonstrates how to extend the procedure to two collective variables. The script generates a metadynamics trajectory on a quartic double well, reconstructs the mean force and FES on a rectangular grid, and produces a multi-panel summary plot.

```
1 # Import libraries
2 from pyMFI import MFI
3 from pyMFI import run_plumed
4
5 # Define surface
6 grid = np.linspace(-2, 2, 201)
7 X, Y = np.meshgrid(grid, grid)
8 Z_string = "7*x^4-23*x^2+7*y^4-23*y^2"
```

```
Z = 7*X**4 - 23*X**2 + 7*Y**4 - 23*Y**2
Z = Z - np.min(Z)

#Simulation parameters
simulation_parameters = {"analytical_function":Z_string,
    "simulation_steps":1_000_000, "temperature":1,
    "gaus_width_x":0.1, "gaus_width_y":0.1, "gaus_height":0.5,
    "biasfactor":10, "gaus_pace":100, "file_extension":"_2D"}

#Run Metadynamics simulation on custom potential
run_plumed.run_langevin2D(**simulation_parameters)

#Read the HILLS and COLVAR file
HILLS = MFI.load_HILLS_2D(hills_name="HILLS_2D")
```

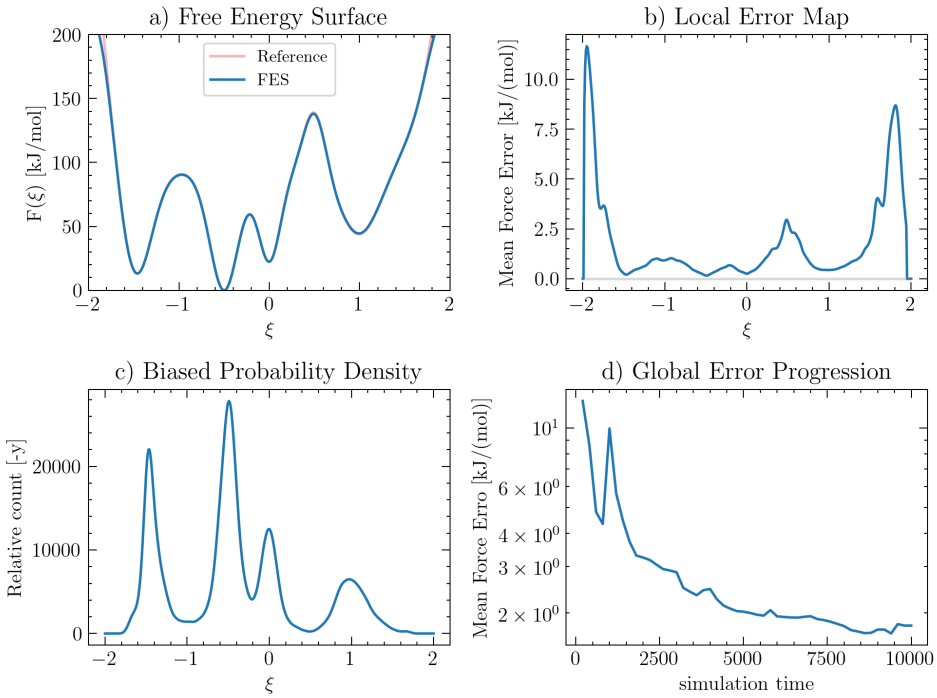


Figure B.1: Representative output of the one-dimensional metadynamics run analysed with MFI. (a) FES reconstructed with MFI (blue) compared with the reference analytical profile (red); (b) Local error of the mean force. (c) Biased probability density accumulated during the simulation. (d) Progression of the error of the mean force.

```
[position_x, position_y] =
↪ MFI.load_position_2D(position_name="position_2D")

# MFI parameters
MFI_parameters = {"HILLS":HILLS, "position_x":position_x,
↪ "position_y":position_y, "bw":[0.1,0.1], "min_grid":[-2,-2],
↪ "max_grid":[2,2], "nbins":[201,201],
↪ "use_weighted_st_dev":False}

#Compute the time-independent mean force
results = MFI.MFI_2D(**MFI_parameters)
[X, Y, PD, FX, FY, st_dev, st_err, cutoff, vol, st_err_prog,
↪ aad_prog, t_prog, PD2, ofv_num_x, ofv_num_y] = results
force_terms_2D = PD, PD2, FX, FY, ofv_num_x, ofv_num_y

#integration on a non-periodic domain
[X, Y, FES] = MFI.FFT_intg_2D(FX, FY, min_grid=[-2,-2],
↪ max_grid=[2,2])

#Plot the results
MFI.plot_recap_2D(X, Y, FES, PD, st_err, st_err_prog, t_prog,
↪ FES_lim=40, ofe_map_lim=10)
```

B.1.3 Combining Simulations with Different Biases

Finally, the code below shows how two independent simulations, a global exploration run and a harmonically restrained refinement, can be merged by patching their mean-force fields before integration. The resulting FES benefits from the broader sampling of the exploratory trajectory while retaining the precision of the local refinement.

```
#Simulation parameters with harmonic potential and MetaD
simulation_parameters = {"analytical_function":Z_string,
 ↪  "simulation_steps":1_000_000, "temperature":1,
 ↪  "gaus_width_x":0.1, "gaus_width_y":0.1, "gaus_height":0.2,
 ↪  "biasfactor":4, "hp_centre_x":0.0, "hp_centre_y":0.0,
 ↪  "hp_kappa_x":40, "hp_kappa_y":40, "gaus_pace":100,
 ↪  "file_extension":"_HP"}

#Run Metadynamics simulation on custom potential
run_plumed.run_langevin2D(**simulation_parameters)
```

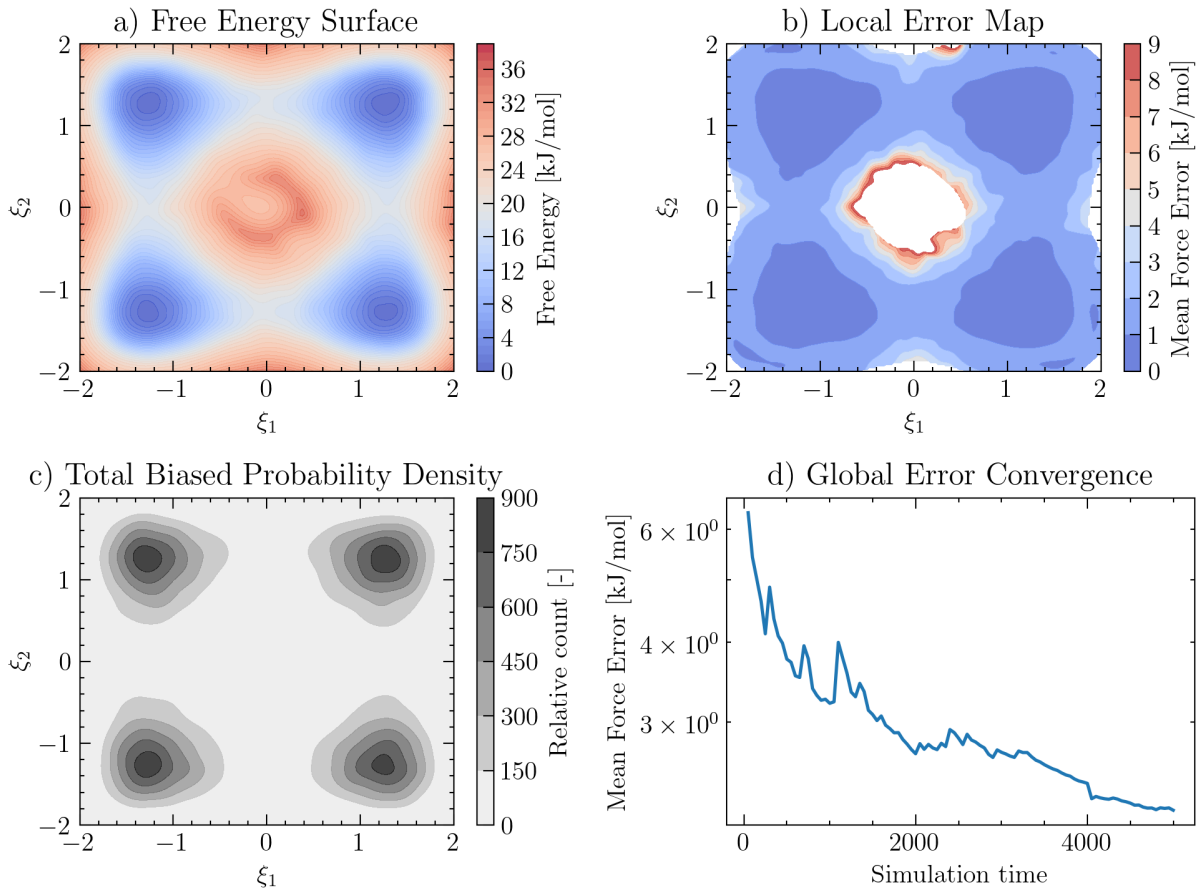


Figure B.2: Representative output of the one-dimensional metadynamics run analysed with MFI. (a) Reconstructed FES. (b) map of the mean force error. (c) Total biased probability density. (d) Global convergence of the mean force error.

```
6
7  #Read the HILLS and COLVAR file
8  HILLS = MFI.load_HILLS_2D(hills_name="HILLS_HP")
9  [position_x, position_y] =
   ↪  MFI.load_position_2D(position_name="position_HP")
10
11 # MFI parameters
12 MFI_parameters = {"HILLS":HILLS, "position_x":position_x,
   ↪  "position_y":position_y, "bw":[0.05,0.05], "min_grid":[-2,-2],
   ↪  "max_grid":[2,2], "nbins":[201,201],
   ↪  "base_terms":force_terms_2D, "hp_centre_x":0.0,
   ↪  "hp_centre_y":0.0, "hp_kappa_x":40, "hp_kappa_y":40,
   ↪  base_terms":force_terms_2D, "use_weighted_st_dev":False}
13
14 #Compute the time-independent mean force
15 results = MFI.MFI_2D(**MFI_parameters)
16 [X, Y, PD, FX, FY, st_dev, st_err, cutoff, vol, st_err_prog,
   ↪  aad_prog, t_prog, PD2, ofv_num_x, ofv_num_y] = results
17 force_terms_HP = PD, PD2, FX, FY, ofv_num_x, ofv_num_y
18
19 # patch force terms of the two simulations
20 PD_patch, _, FX_patch, FY_patch, _, _ =
   ↪  MFI.patch_2D([force_terms_2D, force_terms_HP])
21
22 #integration on a non-periodic domain
23 [X, Y, FES_patch] = MFI.FFT_intg_2D(FX_patch, FY_patch,
   ↪  min_grid=[-2,-2], max_grid=[2,2])
24
25 #Plot the results
26 MFI.plot_recap_2D(X, Y, FES_patch, PD_patch, st_err, st_err_prog,
   ↪  t_prog, FES_lim=40, ofe_map_lim=10)
```

B.2 MFIclass 1D: Manual Reinitialisation

This example demonstrates how the individual exploration, MetaD, flat and umbrella-sampling (US) stages introduced in Section 5.3 can be reproduced step-by-step with the object-oriented implementation `MFI_class1D`. All simulations are run on the analytical potential defined in the code below. The same grid and default parameters are reused throughout and modified only where explicitly stated.

```
# Import libraries
from MFI_class1D import MFI_class1D
from MFI_class1D import MFI_lib1D as lib1
```

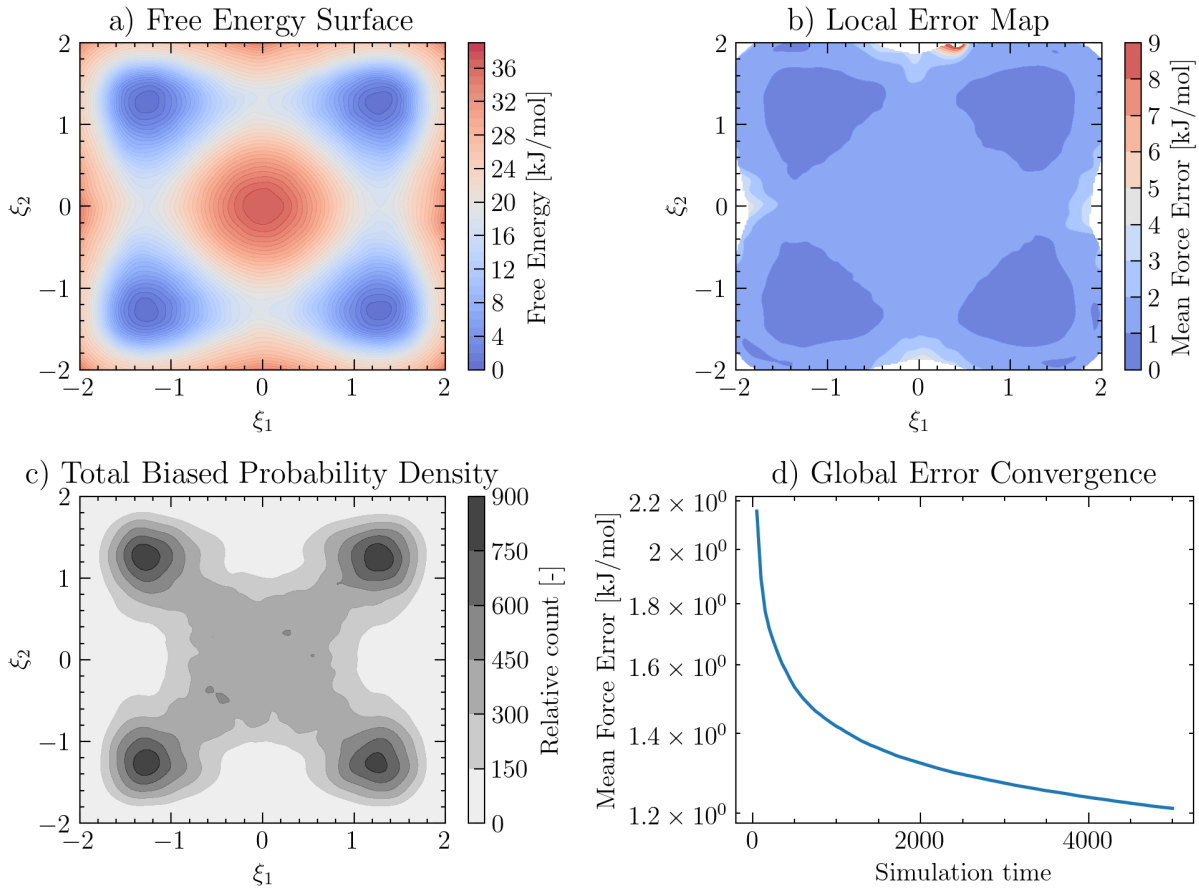


Figure B.3: Representative output of the result obtained after patching the MetaD and MetaD + harmonically restrained simulations. (a) Combined FES. (b) Map of the combined mean force error. (c) Combined biased probability density. (d) Global convergence of the mean force error for the second simulation.

```
4
5  # Define surface
6  grid = np.linspace(-6,6, 501)
7  y_string = "4/3*(-14*exp(-0.25*(x+3.5)^4)-25*exp(-0.25*(x-3.5)^4)-1
   ↪  0*exp(-(x+0.5)^2)-2*sin(-8*x)+exp(-x-4.5)^2+exp(x-4.5)^2)"
8  y = 4/3*(-14*np.exp(-0.25*(grid+3.5)**4)-25*np.exp(-0.25*(grid-3.5)
   ↪  **4)-10*np.exp(-(grid+0.5)**2)-2*np.sin(-8*grid)+np.exp(-grid-4
   ↪  .5)**2+np.exp(grid-4.5)**2)
9  y = y - min(y)
10 FES_cutoff = 50
11
12 # Define plumed input
13 plumed_dat_text = f"p: DISTANCE ATOMS=1,2 COMPONENTS\nff: MATHEVAL
   ↪  ARG=p.x FUNC=({y_string}) PERIODIC=NO\nbb: BIASVALUE ARG=ff\n"
14 [pl_x, pl_min, pl_max, pl_n, pl_ext] = lib1.get_plumed_grid_1D(grid,
   ↪  -7, 7, print_info=True)
15
16 # Define default parameters for the simulation and MFI
17 default_params = {"grid":grid, "y":y, "bw":0.036, "periodic":False,
   ↪  "System":"Langevin", "plumed_dat_text":plumed_dat_text,
   ↪  "pl_grid":pl_x, "simulation_steps": 500_000, "position_pace":50,
   ↪  "metad_pace":500, "n_pos_per_window":10, "metad_width":0.1,
   ↪  "metad_height":20, "biasfactor":80 , "Bias_sf":1.05,
   ↪  "gaus_filter_sigma":1, "FES_cutoff":80, "bootstrap_iter":100,
   ↪  "simulation_folder_path": path_data}
```

B.2.1 Exploration Stage

During exploration an aggressive MetaD bias (`metad_height` $= 20$ kJ mol^{-1}, `biasfactor` $= 80$) is deposited for 1.5×10^5 MD steps. The script below launches the simulation, reconstructs the mean force on the fly, and produces a multi-panel diagnostic plot. Figure B.4 summarises the resulting sampling quality and error metrics.

```
### MFI for exploration stage
new_params = {**default_params}
new_params.update({"simulation_steps":"150_000", "ID":"_expl"})

## Load MFI object
MFI = MFI_class1D.MFI1D(**new_params)

# Run the simulation
MFI.run_simulation()

# Analyse the data
MFI.analyse_data()
t1, aofe1, aad1 = MFI.Avr_Error_list[:,[0, 2, MFI.aad_index]].T

# Plot results
MFI.plot_results()
```

B.2.2 MetaD Stage

The second phase replaces the exploration bias by a gentler well-tempered MetaD (`metad_height` = 5 kJ mol^{-1}, `biasfactor` = 10) constructed on top of the exploration FES. The additional bias accelerates transitions while progressively refining the force estimate. The results are displayed in Figure B.5.

```
### MFI for MetaD stage
new_params = {**default_params}
new_params.update({"ID":"_MetaD", "metad_height":5, "biasfactor":10,
↪  "base_forces":MFI.force_terms, "base_time":MFI.sim_time})

## Load MFI object
MFI2 = MFI_class1D.MFI1D(**new_params)

```

```
## Make the external bias
MFI2.make_external_bias(MFI.FES)

# Run the simulation
MFI2.run_simulation()

# Analyse the data
MFI2.analyse_data(print_analysis=False)
t2, aofe2, aad2 = MFI2.Avr_Error_list[:,[0, 2, MFI.aad_index]].T

# Plot results
MFI2.plot_results(more_aofe=[t1, aofe1] , more_aad = [t1, aad1])
```

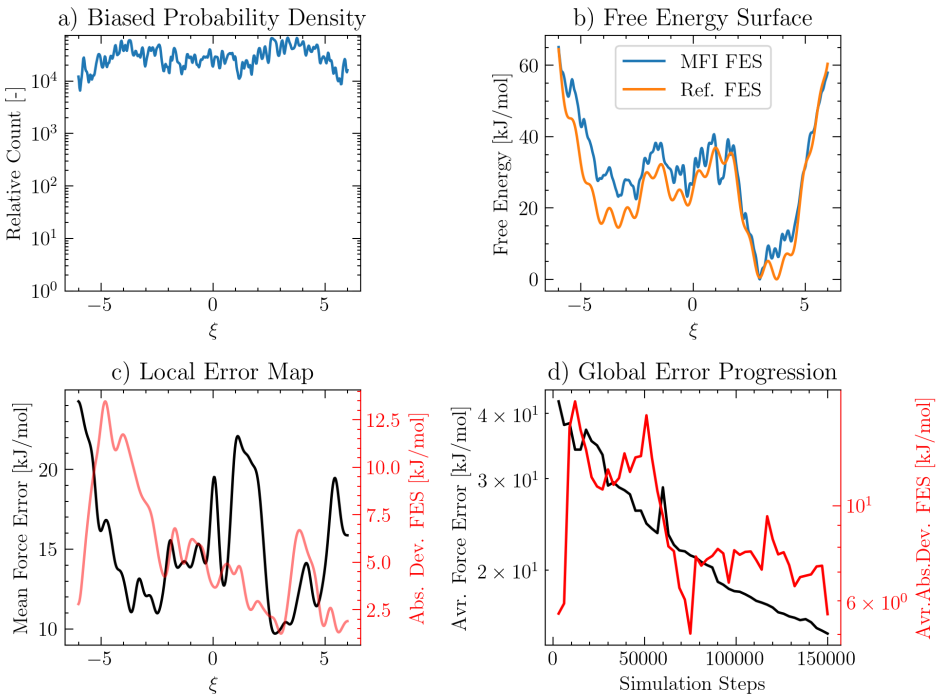


Figure B.4: Results of the exploration stage. (a) Relative biased probability density; (b) reconstructed FES (blue) compared with the analytical reference (orange); (c) local mean-force error (black, left axis) and absolute deviation of the FES (red, right axis); (d) global convergence of the average mean-force error (black, left axis) and average absolute deviation of the FES (red, right axis) as a function of simulation steps.

B.2.3 Flat Stage

A third simulation further smooths the bias landscape by using flatter Gaussians and a reduced bias factor; the external bias is regenerated from the latest FES estimate and rescaled by a factor `Bias_sf` $= 0.9$ to avoid overshoot. The results are displayed in Figure B.6.

```
### MFI for flat stage
# prepare the base forces and time
base_forces_3 = lib1.patch_forces(MFI.force_terms, MFI2.force_terms)
base_time_3 = MFI.sim_time + MFI2.sim_time

new_params = {**default_params}
```

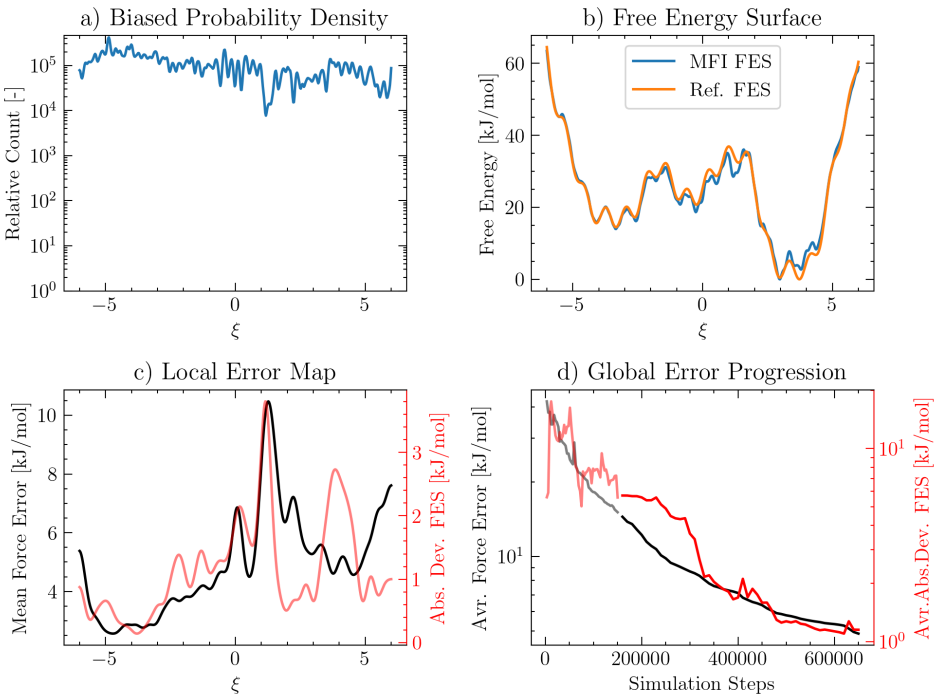


Figure B.5: Results of the MetaD stage. (a) Relative biased probability density; (b) reconstructed FES (blue) compared with the analytical reference (orange); (c) local mean-force error (black, left axis) and absolute deviation of the FES (red, right axis); (d) global convergence of the average mean-force error (black, left axis) and average absolute deviation of the FES (red, right axis) as a function of simulation steps. The transparent lines represent earlier convergence results.

```
new_params.update({"ID":"_flat", "metad_height":1, "biasfactor":5,
    "metad_width":0.05, "metad_pace":1000, "position_pace":50,
    "n_pos_per_window":20, "base_forces":base_forces_3,
    "base_time":base_time_3})

## Load MFI object
MFI3 = MFI_class1D.MFI1D(**new_params)

## Make the external bias
MFI3.make_external_bias(MFI2.FES, Bias_sf=0.9, gaus_filter_sigma=3,
    FES_cutoff=max(MFI2.FES))

# Run the simulation
MFI3.run_simulation()

# Analyse the data
MFI3.analyse_data()
t3, aofe3, aad3 = MFI3.Avr_Error_list[:,[0, 2, MFI.aad_index]].T

# Plot results
MFI3.plot_results(more_aofe=[[t1, aofe1], [t2, aofe2]] , more_aad =
    [[t1, aad1], [t2, aad2]])
```

B.2.4 US Stage

An umbrella potential centred at $\xi = 1$ with force constant $\kappa = 5$ kJ mol^{-1} rad^{-2} is combined with the current MetaD bias. This hybrid set-up allows targeted refinement of selected regions without discarding information gathered in previous stages. The results are displayed in Figure B.7.

```
### MFI with harmonic restraint (Umbrella sampling)
# prepare the base forces and time
```

```
base_forces_4 = lib1.patch_forces(base_forces_3, MFI3.force_terms)
base_time_4 = base_time_3 + MFI3.sim_time

new_params = {**default_params}
new_params.update({"ID":"_4", "bw":MFI.bw/2, "metad_height":1,
    "biasfactor":5, "metad_width":0.05, "metad_pace":500,
    "position_pace":50, "n_pos_per_window":10, "initial_position":1,
    "hp_centre":1, "hp_kappa":5, "base_forces":base_forces_4,
    "base_time":base_time_4})

## Load MFI object
MFI4 = MFI_class1D.MFI1D(**new_params)

```

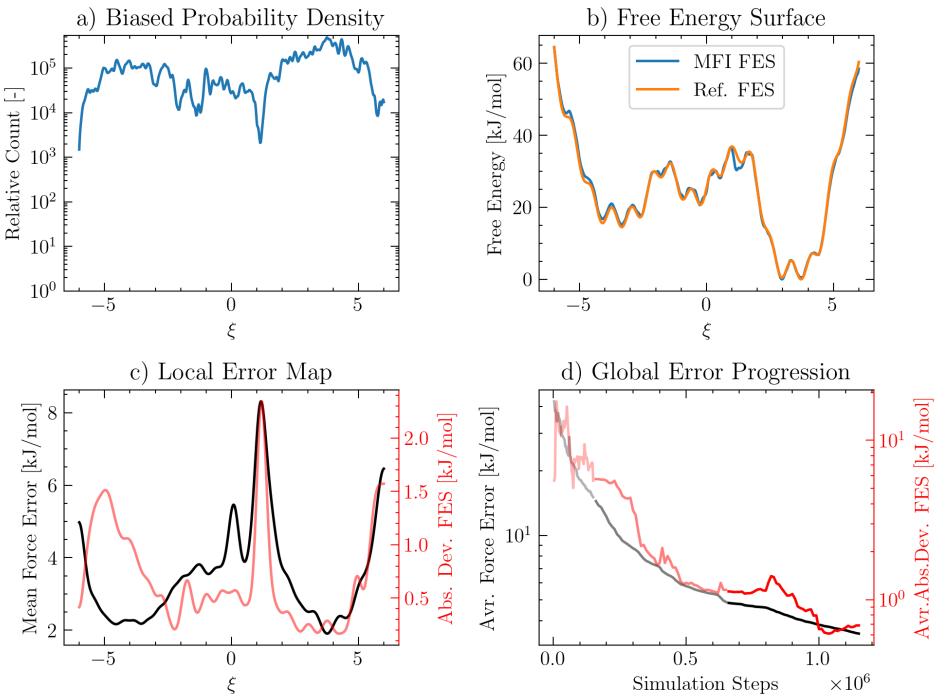


Figure B.6: Results of the flat stage. (a) Relative biased probability density; (b) reconstructed FES (blue) compared with the analytical reference (orange); (c) local mean-force error (black, left axis) and absolute deviation of the FES (red, right axis); (d) global convergence of the average mean-force error (black, left axis) and average absolute deviation of the FES (red, right axis) as a function of simulation steps. The transparent lines represent earlier convergence results.

```
## Make the external bias
MFI4.make_external_bias(MFI3.FES, gaus_filter_sigma=None,
↪  FES_cutoff=max(MFI3.FES))

# Run the simulation
MFI4.run_simulation()

# Analyse the data
MFI4.analyse_data(print_analysis=False)

# Plot results
more_aofe = [[t1, aofe1], [t2, aofe2], [t3, aofe3]]
more_aad = [[t1, aad1], [t2, aad2], [t3, aad3]]
MFI4.plot_results(more_aofe=more_aofe , more_aad = more_aad)
```

B.2.5 Error Based Sampling Stage

In a final demonstration the static bias is derived from the local uncertainty of the mean force rather than from the FES itself. High-error regions are thus preferentially revisited and refined. The code below loads the force-error map obtained after the flat stage, converts it into a bias potential (option `bias_type="error"`) and launches an additional simulation. Figure B.8 documents the outcome.

```
### MFI for special stage - error bias
new_params = {**default_params}
new_params.update({"ID":"_5", "metad_height":0.5, "biasfactor":5,
↪  "metad_width":0.1, "metad_pace":1000, "position_pace":50,
↪  "n_pos_per_window":20, "base_forces":base_forces_4,
↪  "base_time":base_time_4})

## Load MFI object
MFI5 = MFI_class1D.MFI1D(**new_params)
```

```
7
8  ## Make the external bias
9  MFI5.make_external_bias(MFI3.FES, gaus_filter_sigma=None,
   ↪  FES_cutoff=FES_cutoff, bias_type="error", error=MFI3.ofe)
10 ## Alternative for probability density based bias:
   ↪  MFI5.make_external_bias(MFI3.FES, bias_type="PD",
   ↪  error=MFI3.PD)
11
12
13 # Run the simulation
14 MFI5.run_simulation()
15
16 # Analyse the data
```

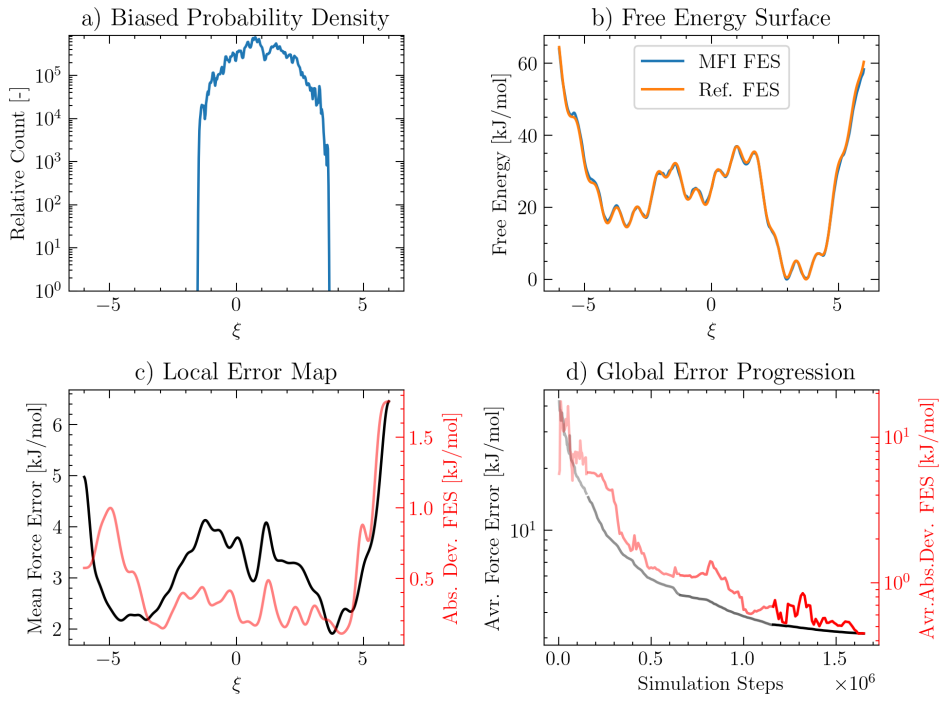


Figure B.7: Results of the US stage. (a) Relative biased probability density; (b) reconstructed FES (blue) compared with the analytical reference (orange); (c) local mean-force error (black, left axis) and absolute deviation of the FES (red, right axis); (d) global convergence of the average mean-force error (black, left axis) and average absolute deviation of the FES (red, right axis) as a function of simulation steps. The transparent lines represent earlier convergence results.

```
MFI5.analyse_data(print_analysis=False)


# Plot results
MFI5.plot_results(more_aofe=more_aofe , more_aad = more_aad)
```

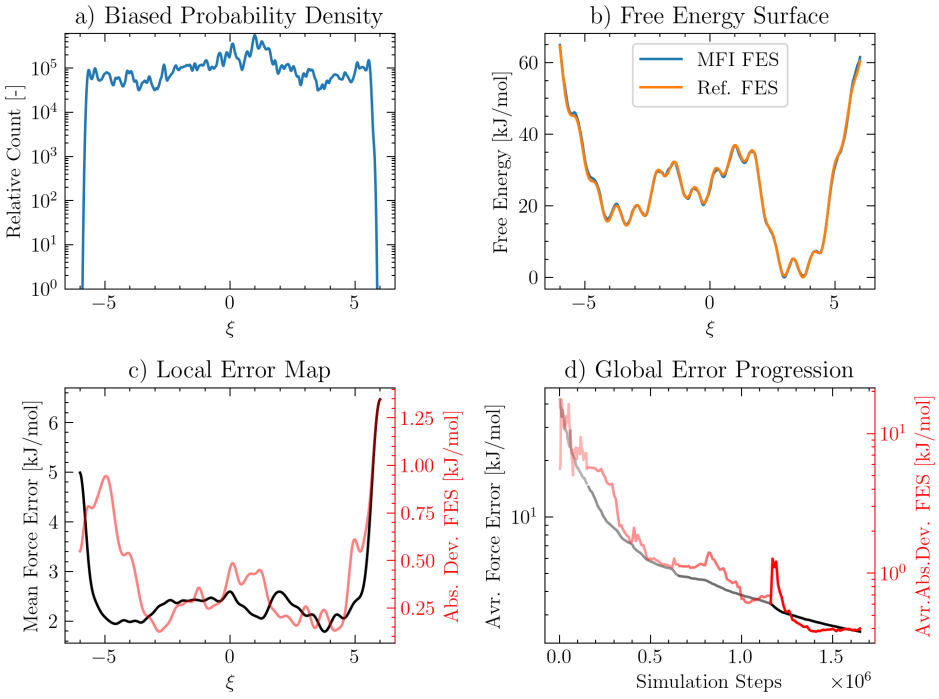


Figure B.8: Results of the error-based sampling stage. (a) Relative biased probability density; (b) reconstructed FES (blue) compared with the analytical reference (orange); (c) local mean-force error (black, left axis) and absolute deviation of the FES (red, right axis); (d) global convergence of the average mean-force error (black, left axis) and average absolute deviation of the FES (red, right axis) as a function of simulation steps. The transparent lines represent earlier convergence results.

B.3 MFIclass 1D: Automatic Reinitialisation

The manual workflow of Section B.2 can be fully automated.

B.3.1 Serial Real-Time Reinitialisation

In a serial real-time reinitialisation (SRTR) campaign each simulation is analysed on the fly and, once the target error is reached or the time budget exhausted, a new simulation is spawned with optimised parameters. The

code below launches such a campaign; the final diagnostics are plotted in Figure B.9.

```
default_params = {"grid":grid, "y":y, "bw":0.036, "periodic":False,
↪ "System":"Langevin", "plumed_dat_text":plumed_dat_text,
↪ "pl_grid":pl_x, "position_pace":20, "metad_pace":200,
↪ "n_pos_per_window":10, "Bias_sf":1.05, "gaus_filter_sigma":1,
↪ "FES_cutoff":50,"bootstrap_iter":50, "simulation_folder_path":
↪ path_data}

# Prepare the MFI object
MFI = MFI_class1D.MFI1D(**default_params)

# Run simulation with real time analysis and re-initialisation
MFI.MFI_real_time_ReInit(ID="_SRTR", goal=0.01,
↪ main_error_type="ST_ERR", n_pos_before_analysis=200,
↪ guaranteed_sim_time=1, max_sim_time=2, time_budget=5) #
↪ optional: reset_forces_after="Exploration"

# Plot the final results
MFI.plot_results()
```

In Section 5.4, it was shown that ignoring the data gathered in the exploration stage results in a lower final error. This can be arranged resetting the force terms after the exploration stage to zero by using an additional argument in the MFI_real_time_ReInit function, specifying: reset_forces_after="Exploration". Additionally, an existing simulation campaign can be reanalysed and restarted. Also here the force terms can be reset after an arbitrary simulation.

In the fist example, a previous SRTR campaign is restarted with an extended time budget. The restart_SRTR argument specifies a "fast" restart, where existing force terms are loaded, and the "ignore_1" argument specifies that the force terms should be reset after analysing the first simulation.

In the second example, a previous SRTR campaign is only reloaded. The restart_SRTR argument specifies a ”full” analysis, where the CV and hills data is analysed from scratch and the force terms are reevaluated. The ”ignore_3” argument specifies that the force terms should be reset after analysing the first three simulations.

```
# Fast Restart of previous SRTR campaign, ignoring the first (1) simulation
MFI.MFI_real_time_ReInit(ID="_SRTR", goal=0.01, main_error_type="ST_ERR", n_pos_before_analysis=500, guaranteed_sim_time=1, max_sim_time=2, time_budget=10, restart_SRTR="fast_ignore_1")


```

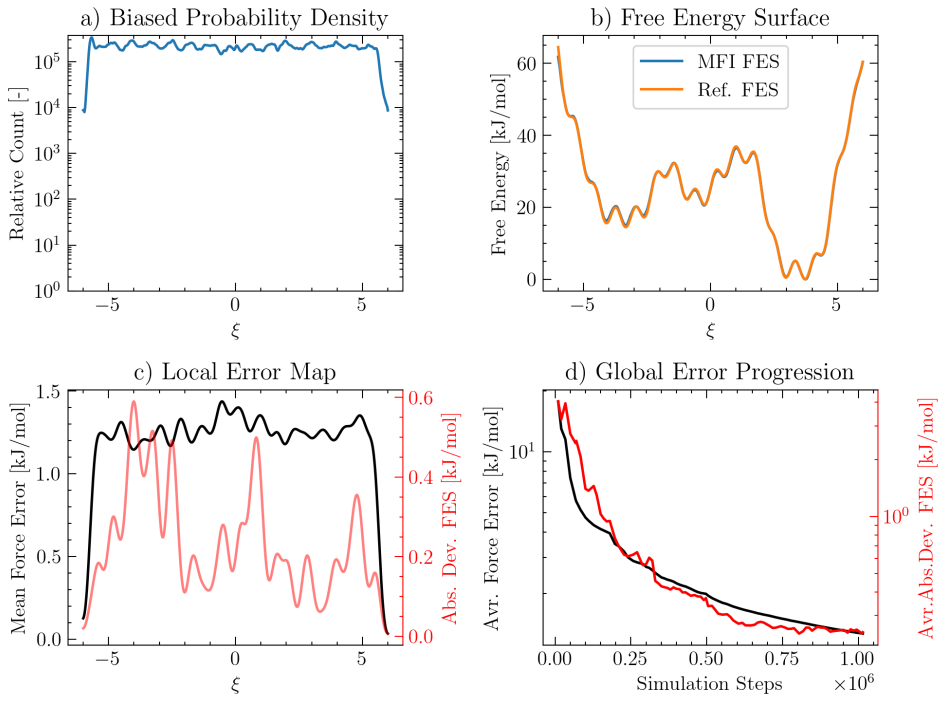


Figure B.9: Results of the SRTR campaing. (a) Relative biased probability density; (b) reconstructed FES (blue) compared with the analytical reference (orange); (c) local mean-force error (black, left axis) and absolute deviation of the FES (red, right axis); (d) global convergence of the average mean-force error (black, left axis) and average absolute deviation of the FES (red, right axis) as a function of simulation steps.

```
# Reanalyse previous SRTR campaing, ignoring the first 3 simulations
MFI.MFI_real_time_ReInit(ID="_SRTR", restart_SRTR="full_reload_ignore_3")
```

B.3.2 Parallel Real-Time Reinitialisation

A parallel real-time reinitialisation (PRTR) campaign distributes the workload over multiple workers that exchange force information after every analysis step. The launcher syntax, resetting of the force terms, reanalysis and restarting are analogous to SRTR and is illustrated in the code below. Figure B.10 summarises the performance of a four-worker PRTR run.

```
# Initialise the MFI object
MFI_PRTR = MFI_class1D.MFI1D(**default_params)

# Initialise the PRTR object
mfi_PRTR = MFI_PRTR.MFI_parallel_RTR(parent=MFI_PRTR, ID="_PRTR", workers=4, workers_exploration=1, goal=0.01, main_error_type="ST_ERR", n_pos_before_analysis=200, guaranteed_sim_time=1, max_sim_time=2, time_budget=10)

# Run the PRTR campaign
mfi_PRTR.run()

# Plot= the final results
mfi_PRTR.sim[0].plot_results()
```

B.4 MFIclass 2D: MFI with Reinitialisation

The two-dimensional implementation mirrors the one-dimensional workflow. For this reason, a different feature is presented: The simulation of alanine dipeptide with GROMACS. This is done by providing a starting structure, the

mdp file, and a topology file. MFI automatically creates an input.tpr file that is used to start the MD simulation.

First, the libraries are imported, the grid and the system is defined, and the other variables are set.

```
# Import libraries
from MFI_class2D import MFI_class2D
from MFI_class2D import MFI_lib2D as lib2

# Define gird
grid = np.linspace(-np.pi, np.pi, 101);
X, Y = np.meshgrid(grid, grid)
[plX, plY, pl_min, pl_max, pl_n, pl_extra] =
↪  lib2.get_plumed_grid_2D(X, Y, periodic=[True, True])
```

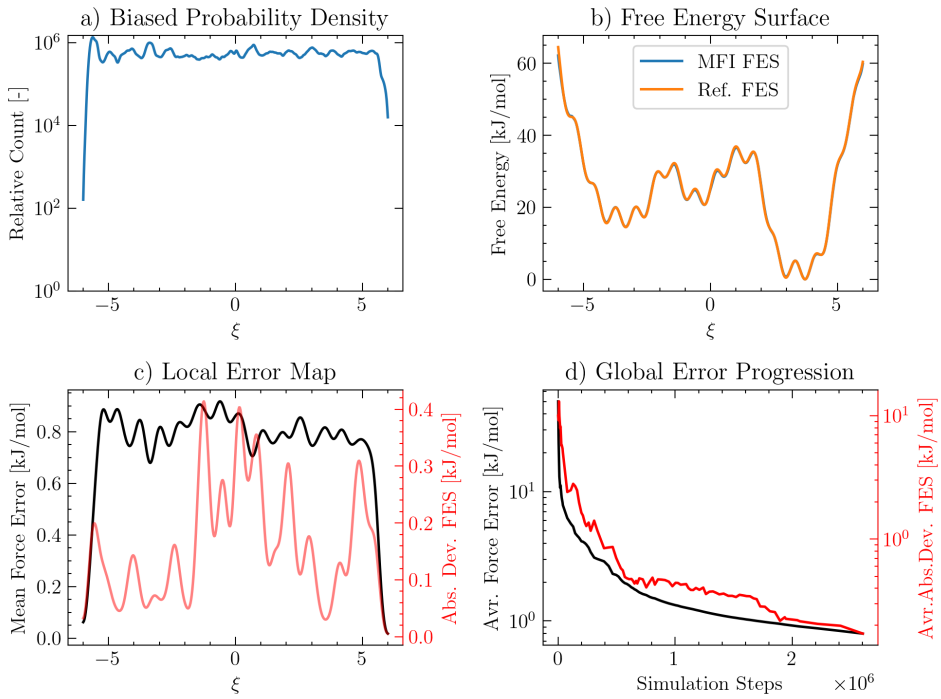


Figure B.10: Results of the PRTR campaing. (a) Relative biased probability density; (b) reconstructed FES (blue) compared with the analytical reference (orange); (c) local mean-force error (black, left axis) and absolute deviation of the FES (red, right axis); (d) global convergence of the average mean-force error (black, left axis) and average absolute deviation of the FES (red, right axis) as a function of simulation steps.

```
9
10 # Define Alanine Dipeptide system
11 path_AP = path_data + "Alanine_Dipeptide/"
12 path_input = path_AP + "input_files/"
13 plumed_dat_text = f"MOLINFO
   ↪ STRUCTURE={path_input}reference.pdb\nphi: TORSION
   ↪ ATOMS=@phi-2\npsi: TORSION ATOMS=@psi-2\n"
14 _, _, AP_ref = lib2.load_pkl(path_input + "AP_ref_101x101.pkl")
15
16 # default parameters for the MFI class
17 default_params = {"X":X, "Y":Y, "Z":AP_ref, "plX":plX, "plY":plY,
   ↪ "periodic":[True, True], "System":"gromacs",
   ↪ "cv_name":["phi","psi"], "plumed_dat_text":plumed_dat_text,
   ↪ "kT":2.49, "time_step":0.001, "simulation_steps":500_000,
   ↪ "position_pace":20, "n_pos_per_window":10, "metad_pace":200,
   ↪ "metad_width":[0.1, 0.1], "metad_height":3, "biasfactor":10,
   ↪ "Bias_sf":1, "gaus_filter_sigma":5, "bw":[0.1, 0.1],
   ↪ "FES_cutoff":50.0, "bootstrap_iter":100,
   ↪ "simulation_folder_path": path_AP, "ID":"_1",
   ↪ "make_tpr_input_file":True,
   ↪ "structure_gro_file_path":path_input+"structure.gro",
   ↪ "mdp_file_path":path_input+"gromppvac.mdp",
   ↪ "top_file_path":path_input+"topology.top",
   ↪ "pdb_file_path":path_input+"reference.pdb"}
```

Next, the MFI object is initialised, after which the simulation is started and analysed. The results are shown in Figure B.11

```
1 # Initialise the MFI object
2 MFI = MFI_class2D.MFI2D(**default_params)
3
4 # Run Alanine Dipeptide simulation
5 MFI.run_simulation()
```

```
6
7  # Analyse the data
8  MFI.analyse_data()
9  t1, v1, aofe1, aad1 = MFI.Avr_Error_list[:,0],
   ↪  MFI.Avr_Error_list[:,1], MFI.Avr_Error_list[:,2],
   ↪  MFI.Avr_Error_list[:, MFI.Avr_Error_info.index("AAD")]
10
11 # Plot the results
12 MFI.plot_results()
```

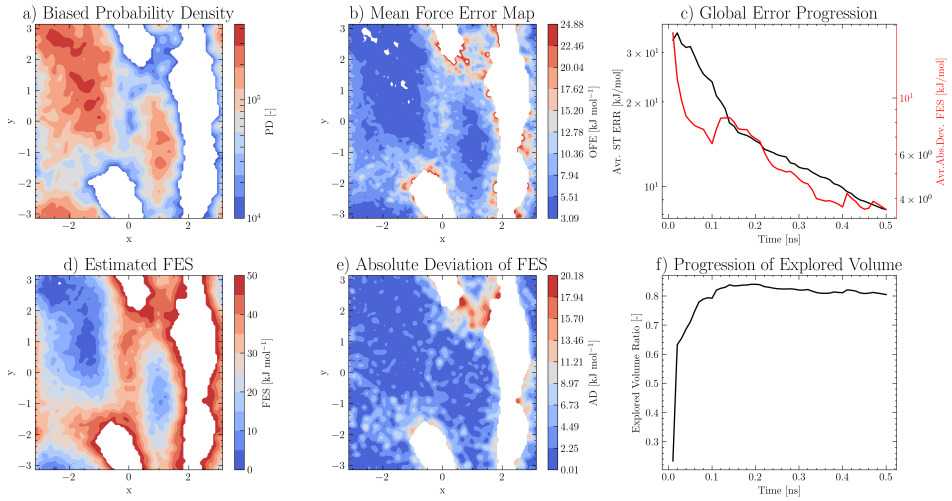


Figure B.11: First simulation on alanine dipeptide. (a) Biased probability density. (b) Map of the mean force error. (c) Global error progression of the average mean-force error (black, left axis) and average absolute deviation of the FES (red, right axis) (d) estimated FES. (e) Absolute deviation of FES. (f) Fraction of the CV space explored over time.

Next, it is shown how a new simulation can be reinitialised using the transition path bias that was introduced in Section 5.3.6. For this purpose, the previous FES is recycled to find the InvF bias and estimate the transition path, from which the transition path bias is calculated, shown in Figure 5.8 (a-b). Additionally, the previous trajectory file is used to find a new initial structure that corresponds to $\phi = 1$ and $\psi = 0$, which is the approximate location secondary basin. The results of the second simulation are provided in Figure B.12, whereas the final results are displayed in Figure 5.8 (c-d).

```
# Run the Alanine Dipeptide simulation with transition path bias
new_params = {**default_params}
new_params.update({"simulation_steps":500_000, "metad_height":0.5,
↪  "biasfactor":5, "metad_width":[0.05, 0.05], "bw":[0.02, 0.02],
↪  "initial_position":[1, 0], "ID":"_trans_path_bias_2",
↪  "find_sim_init_structure":True,
↪  "trajectory_xtc_file_path_list":[path_input+"traj_0.xtc"],
↪  "base_forces":MFI.force_terms, "base_time": MFI.sim_time})

# Initialise the MFI object
MFI_TPB = MFI_class2D.MFI2D(**new_params)

# Make the external bias
MFI_TPB.make_external_bias(FES=MFI.FES, bias_type="transition_path")

# Run Alanine Dipeptide simulation
MFI_TPB.run_simulation()

# Analyse the data
MFI_TPB.analyse_data()

# Plot the results
MFI_TPB.plot_results(more_aofe=[t1, aofe1] , more_aad = [t1, aad1],
↪  more_vol=[t1, v1])
```

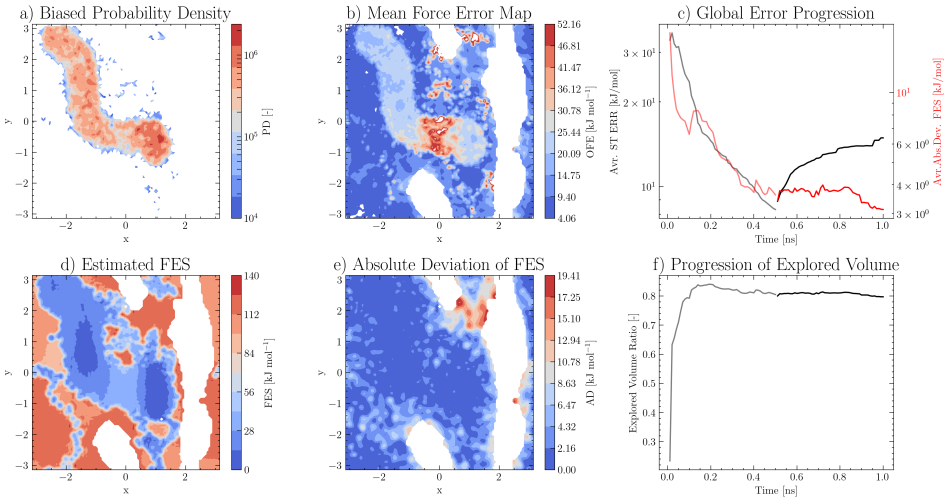


Figure B.12: Second alanine dipeptide run using the transition-path bias. (a) Biased probability density. (b) Map of the mean force error. (c) Global error progression of the average mean-force error (black, left axis) and average absolute deviation of the FES (red, right axis) (d) estimated FES. (e) Absolute deviation of FES. (f) Fraction of the CV space explored over time.