DEUTSCHES ELEKTRONEN-SYNCHROTRON
Ein Forschungszentrum der Helmholtz-Gemeinschaft

DESY 20-149
arXiv:2008.03090
August 2020

Multilevel Monte Carlo for Quantum Mechanics on a Lattice

K. Jansen

Deutsches Elektronen-Synchrotron DESY, Zeuthen

E. H. Müller

Department of Mathematical Sciences,
University of Bath, UK

R. Scheichl

Institut für Angewandte Mathematik,
Ruprecht-Karls-Universität Heidelberg

ISSN 0418–9833

PLATANENALLEE 6 - 15738 ZEUTHEN

Multilevel Monte Carlo for quantum mechanics on a lattice

Karl Jansen[a], Eike Hermann Müller[b,*], and Robert Scheichl[c]

[a] *DESY Zeuthen, Platanenallee 6, 15738 Zeuthen, Germany*
[b] *Department of Mathematical Sciences, University of Bath, Bath BA2 7AY, Bath, United Kingdom*
[c] *Institute for Applied Mathematics, Ruprecht-Karls-Universität Heidelberg, Im Neuenheimer Feld 205, 69120 Heidelberg, Germany*
[*] *Email:* `e.mueller@bath.ac.uk`

November 18, 2020

Abstract

Monte Carlo simulations of quantum field theories on a lattice become increasingly expensive as the continuum limit is approached since the cost per independent sample grows with a high power of the inverse lattice spacing. Simulations on fine lattices suffer from critical slowdown, the rapid growth of autocorrelations in the Markov chain with decreasing lattice spacing a. This causes a strong increase in the number of lattice configurations that have to be generated to obtain statistically significant results. In this paper, hierarchical sampling methods to tame this growth in autocorrelations are discussed. Combined with multilevel variance reduction techniques, this significantly reduces the computational cost of simulations for given tolerances ϵ_{disc} on the discretisation error and ϵ_{stat} on the statistical error. For an observable with lattice errors of order α and an integrated autocorrelation time that grows like $\tau_{\text{int}} \propto a^{-z}$, multilevel Monte Carlo can reduce the cost from $\mathcal{O}(\epsilon_{\text{stat}}^{-2}\epsilon_{\text{disc}}^{-(1+z)/\alpha})$ to $\mathcal{O}(\epsilon_{\text{stat}}^{-2}|\log\epsilon_{\text{disc}}|^2 + \epsilon_{\text{disc}}^{-1/\alpha})$ or $\mathcal{O}(\epsilon_{\text{stat}}^{-2} + \epsilon_{\text{disc}}^{-1/\alpha})$. Even higher performance gains are expected for non-perturbative simulations of quantum field theories in D-dimensions. The efficiency of the approach is demonstrated on two non-trivial model systems in quantum mechanics, including a topological oscillator that is badly affected by critical slowdown due to freezing of the topological charge. On fine lattices, the new methods are several orders of magnitude faster than standard, single level sampling based on Hybrid Monte Carlo. For high resolutions, multilevel Monte Carlo can be used to accelerate even the cluster algorithm for the topological oscillator. Performance is further improved through perturbative matching. This guarantees efficient coupling of theories on the multilevel lattice hierarchy, which have a natural interpretation in terms of effective theories obtained by renormalisation group transformations.

keywords: Multilevel Monte Carlo, Path Integral, Hierarchical Methods, Numerical Algorithms

1 Introduction

The Euclidean path integral formulation of quantum mechanics [1] allows the calculation of observable quantities as expectation values with respect to infinite-dimensional and highly peaked probability distributions. After discretising the theory on a lattice with finite spacing a, expectation values are computed with Markov Chain Monte Carlo methods (see for example [2] for a highly accessible introduction). This approach is elegant and attractive since it can be extended to quantum field theories, where it allows first principles-predictions for strongly interacting theories such as Quantum Chromodynamics (QCD), see e.g. [3, 4]. Ultimately, however, one is interested in the value of observables in the continuum limit of vanishing lattice spacing $a \to 0$. Since the cost of the calculation grows with a high power of a^{-1}, efficient Monte Carlo sampling techniques are crucial to obtain precise and accurate numerical predictions. Today state-of-the-art techniques [5] are routinely used to accelerate the Metropolis-Hastings algorithm [6, 7] and in particular the Hybrid Monte Carlo (HMC) method [8] has proved to be highly successful in lattice QCD simulations. However, lattice calculations with HMC methods still become prohibitively expensive as the continuum limit is approached. The reasons for this are twofold:

1. For quantum mechanical problems the cost $\mathcal{C}_{\text{sample}}$ of generating a single discretised path grows at least in proportion to the number of lattice sites, which increases with a^{-1} if the physical size of the simulation box is kept fixed (for a quantum field theory in D dimensions the growth would be even faster with a cost of a^{-D} per configuration).

2. As the theory approaches a critical point, subsequent states in the Markov chain are increasingly correlated, which requires the generation of more paths to obtain a given number of *statistically independent* samples.

Furthermore, the law of large numbers dictates that to reduce the statistical (sampling) error below a given tolerance ϵ_{stat}, at least $N_{\text{indep}} \propto \epsilon_{\text{stat}}^{-2}$ *independent* samples have to be generated. While in lattice QCD the continuum limit is usually taken by extrapolating simulations at different lattice spacings a and fixed tolerance ϵ_{stat} on the statistical error, in the multilevel Monte Carlo literature (see e.g. the classical paper [9]) it is more common to decrease ϵ_{stat} in proportion to the tolerance ϵ_{disc} on the discretisation error as the lattice spacing is reduced.

Reducing the combined statistical and discretisation error in this way would make optimal use of computational resources to obtain a result with a given *total* error for a specific fine lattice spacing.

The correlation of subsequent samples in the Markov chain is quantified by the integrated autocorrelation time τ_{int}, which grows particularly rapidly for some quantities, such as the topological susceptibility χ_t in QCD, where it has been observed that $\tau_{\text{int}} \propto a^{-z}$ with $z = 5$ [10]. This is attributed to "freezing" of the topological charge, and can lead to observable effects. Those can be both direct, since e.g. the mass of the η' meson receives important contributions from the topological susceptibility in a pure Yang-Mills theory [11, 12], and more indirect due to the coupling of slow modes with large autocorrelation times to other observables. The authors of [10] further report a milder but still significant growth with $z = 0.5 - 1.0$ for a range of other physically relevant observables. While the rapid growth of the integrated autocorrelation time for the topological susceptibility can be addressed by using open boundary conditions in time [13, 14], this introduces additional complications since it requires lattices with a very large extent in the temporal direction.

To estimate the overall growth in cost of a simulation, as the lattice spacing a is reduced, consider a quantum mechanical observable with a discretisation error that is $\mathcal{O}(a^\alpha)$, where values such as $\alpha = 1, 2$ are typical. To reduce the discretisation error below a tolerance of ϵ_{disc} and the statistical error below ϵ_{stat} incurs a cost

$$\begin{aligned} \mathcal{C}_{\text{StMC}}(\epsilon_{\text{disc}}, \epsilon_{\text{stat}}) &= N_{\text{indep}} \times \tau_{\text{int}} \times \mathcal{C}_{\text{sample}} \\ &= \mathcal{O}(\epsilon_{\text{stat}}^{-2} \epsilon_{\text{disc}}^{-(1+z)/\alpha}), \end{aligned} \tag{1}$$

with standard Monte Carlo (StMC), since $\epsilon_{\text{disc}} \propto a^\alpha$. To get an intuitive understanding of this and subsequent complexity estimates it might be instructive to consider the special case $\alpha = 2$, $z = 0$: since the discretisation error decreases quadratically with a, reducing this error by a factor of 4 can be achieved by halving the lattice spacing, which in turn doubles the cost for generating a single sample if the physical size of the simulation box is kept fixed; in other words, the cost per sample grows in proportion to $\epsilon_{\text{disc}}^{-1/2}$

In this paper, it is shown how this explosion in computational cost can be significantly reduced with hierarchical sampling [15] and Multilevel Monte Carlo (MLMC) [16, 9], which has recently been extended to a Markov chain setting [17, 18]. To generate samples, a hierarchy of $L - 1$ coarser lattices with spacings of $2a, 4a, \dots, 2^{L-1}a$ and corresponding coarse-grained versions of the original theory are constructed. Based on this hierarchy, a recursive implementation of the delayed acceptance method in [15] is proposed. Starting on the coarsest level, proposals are successively extended by additional modes and screened with a standard Metropolis-Hastings accept/reject step on increasingly finer lattices. At this point is important to stress that the coarse lattices are only used to accelerate sampling and do not introduce any additional bias because ultimately each new sample is accepted or rejected step with the correct, original action on the finest lattice. Since evaluating the action on the coarse lattices is substantially cheaper, the cost of generating a single fine level sample is not substantially higher than if a single-level sampler was used. In fact, when compared to a method such as HMC, it may be smaller since the cost of generating an HMC trajectory can be shifted to the coarsest level where it is substantially shorter. Since on each level proposals are screened with a Metropolis-Hastings accept/reject step, the method samples from the correct distribution on the original lattice with spacing a and does not introduce any additional bias, cf. [15]. Due to the convergence of the lattice theories on subsequent levels of the hierarchy with $a \to 0$, hierarchical sampling eliminates the growth in autocorrelation time, reducing the computational cost to

$$\mathcal{C}_{\text{hierarchical}}(\epsilon_{\text{disc}}, \epsilon_{\text{stat}}) = \mathcal{O}(\epsilon_{\text{stat}}^{-2} \epsilon_{\text{disc}}^{-1/\alpha}). \tag{2}$$

MLMC is a variance reduction technique, which uses the fact that the expectation value of an observable (or quantity of interest) Q on a lattice with spacing a can be written as a telescoping sum. For this assume that there is some integer $L \in \mathbb{N}$ and a constant a_0 such that $a = 2^{-L+1}a_0$. Further, let Q_ℓ be the observable measured on a lattice with spacing $2^{-\ell}a_0$. Then

$$\begin{aligned} \mathbb{E}[Q] = \mathbb{E}[Q_{L-1}] = \mathbb{E}[Q_{L-1} - Q_{L-2}] + \mathbb{E}[Q_{L-2}] = \\ = \ldots = \sum_{\ell=0}^{L-1} \mathbb{E}[Y_\ell] \approx \sum_{\ell=0}^{L-1} \widehat{Y}_\ell \end{aligned} \tag{3}$$

where

$$Y_\ell := \begin{cases} Q_0 & \text{for } \ell = 0 \\ Q_\ell - Q_{\ell-1} & \text{for } \ell = 1, 2, \dots, L-1, \end{cases}$$

$$\widehat{Y}_\ell := \frac{1}{N_\ell} \sum_{j=1}^{N_\ell} Y_\ell^{(j)}.$$

Here, the sums in $\widehat{Y}_\ell$ are taken over independent samples, labelled by the superscript "(j)". The key observation is that, except for the very coarsest level, MLMC estimates *differences* of the observable instead of the quantity of interest itself. Provided that theories on subsequent levels can be coupled efficiently and the variance of the difference $Q_\ell - Q_{\ell-1}$ decreases sufficiently rapidly as the lattice spacing a is reduced, significantly lower numbers of samples N_ℓ are sufficient on the finer levels of the grid hierarchy. The majority of the cost can be shifted to the coarser levels $\ell \ll L$, where sampling is substantially cheaper. Due to the exactness of the telescoping sum (i.e. the first equality in Eq. (3)), MLMC does not introduce any additional bias if the individual MC estimators $\widehat{Y}_\ell$ are unbiased. The algorithms described in the paper allow the construction of estimators $\widehat{Y}_\ell$ which have an arbitrarily small bias. In the numerical results presented below the size of this bias is comparable to the discretisation error on the original, fine level lattice. Compared to Eqs. (1) and (2), MLMC further reduces the computational complexity to

$$\mathcal{C}_{\text{MLMC}}(\epsilon_{\text{disc}}, \epsilon_{\text{stat}}) = \mathcal{O}(\epsilon_{\text{stat}}^{-2} |\log \epsilon_{\text{disc}}|^2 + \epsilon_{\text{disc}}^{-1}), \tag{4}$$

see below. Similar estimates have been derived in [9, 17, 18] and it has been demonstrated numerically that MLMC leads to a significant reduction in computational complexity and overall runtime for a range of applications, e.g. in uncertainty quantification (UQ) for sub-surface flow [19, 17], inverse problems [20] or material simulation [21].

While this paper focuses on the application of these new methods in quantum mechanics, the ultimate goal is to apply them in D-dimensional quantum field theories, such as lattice QCD with $D = 4$ and $\alpha = 2$. For $D > \alpha$, the expected gains are even larger, since the cost to generate a single configuration grows like a^{-D} instead of a^{-1} while the accuracy is still decreasing no faster than a^2. The predicted improvement in computational performance is summarised in the following diagram, generalising Eqs. (1), (2) and (4) to D dimensions:

$$\begin{aligned}
\mathcal{C}^{(\text{QFT})}_{\text{StMC}}(\epsilon_{\text{disc}},\epsilon_{\text{stat}}) &= \mathcal{O}(\epsilon_{\text{stat}}^{-2}\epsilon_{\text{disc}}^{-(D+z)/\alpha})\\
&\downarrow \quad \text{(hierarchical sampling)}\\
\mathcal{C}^{(\text{QFT})}_{\text{hierarchical}}(\epsilon_{\text{disc}},\epsilon_{\text{stat}}) &= \mathcal{O}(\epsilon_{\text{stat}}^{-2}\epsilon_{\text{disc}}^{-D/\alpha})\\
&\downarrow \quad \text{(multilevel Monte Carlo)}\\
\mathcal{C}^{(\text{QFT})}_{\text{MLMC}}(\epsilon_{\text{disc}},\epsilon_{\text{stat}}) &= \mathcal{O}(\epsilon_{\text{stat}}^{-2}\epsilon_{\text{disc}}^{1-D/\alpha}+\epsilon_{\text{disc}}^{-D/\alpha}).
\end{aligned} \tag{5}$$

For example, consider the prediction of the topological susceptibility ($z = 5$) in lattice QCD ($D = 4$) with improved action ($\alpha = 2$). In this case, hierarchical sampling reduces the cost of a Monte Carlo simulation from $\mathcal{O}(\epsilon_{\text{stat}}^{-2}\epsilon_{\text{disc}}^{-4.5})$ to $\mathcal{O}(\epsilon_{\text{stat}}^{-2}\epsilon_{\text{disc}}^{-2})$ and MLMC reduces the computational complexity even further to $\mathcal{O}(\epsilon_{\text{stat}}^{-2}\epsilon_{\text{disc}}^{-1}+\epsilon_{\text{disc}}^{-2})$.

To discuss this further, consider first the relative advantage of MLMC over standard Monte Carlo in the continuum limit $\epsilon_{\text{disc}} \to 0$ for fixed ϵ_{stat}. MLMC only requires the generation of a small number of samples on the finest lattice for small ϵ_{disc} (eventually only one for very small ϵ_{disc}), whereas the number of configurations that have to be generated with a standard Monte Carlo method is proportional to $\epsilon_{\text{stat}}^{-2}$. As can be seen from the final two lines of Eq. (5), MLMC is a factor of $\kappa\epsilon_{\text{stat}}^{-2}$ faster than standard Monte Carlo with hierarchical sampling for $\epsilon_{\text{disc}} \to 0$. This argument holds for general α and D; the coefficient κ depends on the relative cost of generating independent samples on the finest level and the coarser levels. Provided those costs are proportional to the number of unknowns on each level (and the constant of proportionality is independent of ϵ_{disc}) we expect κ to lie between 1 and 2.

If ϵ_{stat} is kept fixed as the continuum limit is taken, eventually the statistical error will dominate the discretisation error. To avoid this, one might consider the case where $\epsilon_{\text{disc}} = \epsilon_{\text{stat}} = \epsilon/\sqrt{2}$ and the combined root mean square error is reduced below some given tolerance ϵ. This is the common choice in the multilevel Monte Carlo literature (see e.g. [9]). In that case, the complexity estimates in Eq. (5) become

$$\begin{aligned}
\mathcal{C}^{(\text{QFT})}_{\text{StMC}}(\epsilon) &= \mathcal{O}(\epsilon^{-2-(D+z)/\alpha})\\
&\downarrow \quad \text{(hierarchical sampling)}\\
\mathcal{C}^{(\text{QFT})}_{\text{hierarchical}}(\epsilon) &= \mathcal{O}(\epsilon^{-2-D/\alpha})\\
&\downarrow \quad \text{(multilevel Monte Carlo)}\\
\mathcal{C}^{(\text{QFT})}_{\text{MLMC}}(\epsilon) &= \mathcal{O}(\epsilon^{-1-D/\alpha})
\end{aligned} \tag{6}$$

In quantum field theories, coarse grained actions are naturally obtained by integrating out high-frequency modes in a renormalisation group (RG) transformation, which results in an effective theory with less degrees of freedom. In practice, the RG transformation can be carried out either non-perturbatively (e.g. through a block spin transformation) or through perturbative matching. The latter would, in fact, be sufficient for MLMC as long as the variance of Y_ℓ decays sufficiently rapidly since the coarse levels are only used to accelerate sampling on the original, fine level. For asymptotically free theories, such as lattice QCD, Symanzik-improved actions [22, 23] can be constructed by systematically adding suitable terms which are proportional to powers of the lattice spacing a and which are multiplied by appropriate, so-called 'improvement coefficients'. These coefficients can be tuned non-perturbatively [24, 25], or they can be computed using perturbation theory for sufficiently small lattice spacing [22, 23]. In the MLMC approach, the perturbatively calculated improvement coefficients on different levels of the lattice hierarchy are in fact sufficient since the *differences* of these coefficients between subsequent levels are sufficiently small on fine lattices.

As a proof-of-concept, hierarchical sampling and multilevel Monte Carlo are applied to two problems in quantum mechanics ($D = 1$): a non-symmetric double-well potential and the topological oscillator studied in [26]. The latter case is particularly interesting since it has a topological quantum number, which freezes in the continuum limit ($a \to 0$). This results in a rapid growth of the autocorrelation time of the topological susceptibility if standard HMC sampling is used. Hierarchical sampling all but eliminates this growth, resulting in a dramatic reduction in runtime. Furthermore, the coarse-grained theories can be improved using a perturbative matching technique for this problem, which further increases the efficiency of the hierarchical approach. As demonstrated in [26], for the topological oscillator the so-called 'cluster algorithm' [27] almost entirely eliminates autocorrelations through long-range spin updates. However, this method can be further accelerated with MLMC, leading to a reduction in computational complexity and in absolute runtime for high resolutions. Similar gains are observed for the non-symmetric double-well potential problem with MLMC.

In summary, the main achievements of this work are:

1. It is described in detail how algorithms for hierarchical sampling and multilevel Monte Carlo acceleration can be applied to the path integral formulation of quantum mechanics.

2. It is shown how hierarchical sampling techniques dramatically reduce autocorrelation times.

3. It is further demonstrated that combining this with MLMC leads to an additional reduction in computational complexity and in the total runtime.

4. It is explained how perturbative matching can further improve performance for the topological oscillator.

It is stressed again that the additional gains due to MLMC accelerating are expected to be significantly larger in high-dimensional theories, such as lattice QCD (see Eq. (5)). The present paper therefore aims to lay the foundation for further work on extending the described methods to quantum field theories on a lattice.

Structure. This paper is organised as follows: after briefly reviewing the literature on related approaches in Section 1.1, the application of hierarchical sampling and multilevel Monte Carlo to the path integral formulation of quantum mechanics is discussed in Section 2. The quantum mechanical model problems that are used in this work are described in Section 3, including the construction of coarse-grained actions for those problems. Numerical results for the non-symmetric double-well potential and the topological oscillator are presented in Section 4, in particular we compare the runtime of all considered algorithms for fixed ϵ_{stat}. Section 5 contains the conclusion and outlines directions for future work. More technical topics, such as a detailed cost analysis of MLMC and a discussion of how the methods can be extended to higher dimensional problems, are relegated to the appendix where we also show results for $\epsilon_{\text{stat}} = \epsilon_{\text{disc}} = \epsilon/\sqrt{2}$.

1.1 Relationship to previous work

While hierarchical sampling techniques have been suggested previously (see e.g. [28, 29, 30, 31]), the variance reduction techniques from MLMC significantly improve on this. Eqs. (2) and (4) show that the additional acceleration will lead to a further dramatic reduction in computational complexity. The presented methods are therefore expected to be superior to the approach in [32], which uses a hierarchical method to initialise the simulation, but not for the Monte Carlo sampling. Earlier work in [30, 31] uses renormalisation group techniques to sample close to the critical point of the Ising model where the theories on the coarser levels become self-similar. Similarly, collective cluster-update algorithms [33, 27, 34] have been applied to models in solid state physics close to phase transitions (see e.g. [35]). However, the application of all those techniques is limited to spin systems. The approach here applies to general systems and delivers significant additional speedup through multilevel Monte Carlo variance reduction.

2 Methods

2.1 Path integral formulation of quantum mechanics

For completeness and to introduce the discretised path integral for non-experts, we recapitulate the key principles here. The path integral formulation of quantum mechanics [1] expresses the expectation value of physical observables as the infinite-dimensional sum over all possible configurations or paths $\{x(t)\}$, where $x(t) \in \mathcal{D} \subset \mathbb{R}$ for all times $t \in \mathbb{R}$. In this sum each path $x(t)$ is weighted by a complex amplitude $e^{\frac{i}{\hbar}S(x(t))}$, where $S(x(t))$ is the action, the integral over the Lagrangian $\mathcal{L}$ of the system. This formulation is very elegant since it allows the direct quantisation of any system which can be described by a Lagrangian. In the limit $\hbar \to 0$ fluctuations around the classical path which minimises the action cancel out, and the Euler-Lagrange equations are recovered. However, for simplicity from now on we will work in atomic units where $\hbar = 1$. To make the evaluation of the path integral tractable, two approximations are made: (1) time is restricted to a finite interval $t \in [0, T)$ and (2) the time interval is divided into d intervals of size $a = T/d$, which is known as the lattice spacing. Conditions have to be imposed on the paths at $t = 0$ and $t = T$; here we use periodic boundary conditions $x(T) = x(0)$. Each path $x(t)$, which is defined for all times $t \in [0, T)$, is replaced by a vector $\boldsymbol{x} = (x_0, x_1, \dots, x_{d-1}) \in \Omega = \mathcal{D}^d$. For each $j = 0, 1, \dots, d-1$ the quantity x_j approximates the position $x(t_j)$ of the particle at the time $t_j = aj$. Those two approximations turn the infinite dimensional integral over all paths into an integral over a finite, but high-dimensional domain $\mathcal{D}^d$. Evaluating the integral in Euclidean time converts it to the canonical ensemble average of a statistical system at a finite temperature. More specifically, the expectation value of an observable (commonly known as "Quantity of interest", QoI, in the UQ literature) which assigns a value $Q(\boldsymbol{x})$ to each discrete path $\boldsymbol{x}$ can be written as the following ratio

$$\begin{aligned}\mathbb{E}[Q] &= \frac{\int_{\mathcal{D}} \dots \int_{\mathcal{D}} Q(\boldsymbol{x}) e^{-S(\boldsymbol{x})}\, dx_0 \,\dots\, dx_{d-1}}{\int_{\mathcal{D}} \dots \int_{\mathcal{D}} e^{-S(\boldsymbol{x})}\, dx_0 \,\dots\, dx_{d-1}} \\ &= \int_{\Omega} \pi^*(\boldsymbol{x}) Q(\boldsymbol{x})\, d\boldsymbol{x}\end{aligned} \tag{7}$$

with the d-dimensional probability density π^* given by

$$\pi^*(\boldsymbol{x}) = \mathcal{Z}^{-1} e^{-S(\boldsymbol{x})}, \quad \text{for all } \boldsymbol{x} \in \Omega, \tag{8}$$

with normalisation constant $\mathcal{Z}$. The action $S(\boldsymbol{x})$ is an approximation of the continuum action

$$S(x(t)) = \int_0^T \mathcal{L}(x(t))\, dt$$

where $\mathcal{L}$ is the Lagrangian.

Physically meaningful predictions, which can be compared to experimental measurements, are obtained by extrapolating to the continuum limit $a \to 0$ and infinite volume $T \to \infty$. As d is inversely proportional to the lattice spacing, the integrals in Eq. (7) become very high dimensional in the continuum limit. In this paper we do not discuss finite volume errors (due to finite values of T). In other words, we take the continuum limit $Q^{\text{exact}} = \lim_{a\to 0} \mathbb{E}[Q]$ for finite T as the "true" value for any observables studied here.

2.2 Standard Monte Carlo

Since the distribution π^* in Eq. (8) is highly peaked, the expectation value in Eq. (7) is usually computed with importance sampling. For this, the Metropolis-Hastings algorithm [6, 7] is used to iteratively generate a sequence of samples $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(N-1)} \sim \pi^*$. The expectation value can then be approximated as the sample average

$$\mathbb{E}[Q] \approx \widehat{Q}^{\text{StMC}} := \frac{1}{N}\sum_{j=0}^{N-1} Q(\boldsymbol{x}^{(j)}). \tag{9}$$

A single Metropolis-Hastings step for computing $\boldsymbol{x}^{(t+1)}$, given $\boldsymbol{x}^{(t)}$, is written down in Alg. 1.

Algorithm 1 Standard Metropolis-Hastings step.
Input: current sample $\boldsymbol{x}^{(t)} \sim \pi^*$
Output: new sample $\boldsymbol{x}^{(t+1)} \sim \pi^*$

```
1: Pick proposal $\boldsymbol{y}$ from a probability distribution $q(\cdot|\boldsymbol{x}^{(t)})$.
2: Compute
       $\frac{\pi^*(\boldsymbol{y})}{\pi^*(\boldsymbol{x}^{(t)})} \cdot \frac{q(\boldsymbol{x}^{(t)}|\boldsymbol{y})}{q(\boldsymbol{y}|\boldsymbol{x}^{(t)})} = \exp[-\Delta S]$
   with
       $\Delta S := S(\boldsymbol{y}) - S(\boldsymbol{x}^{(t)}) + \log q(\boldsymbol{y}|\boldsymbol{x}^{(t)}) - \log q(\boldsymbol{x}^{(t)}|\boldsymbol{y})$
3: if $\Delta S < 0$ then
4:     Set $\boldsymbol{x}^{(t+1)} \mapsfrom \boldsymbol{y}$
5: else
6:     Draw uniformly distributed random number $u \in [0,1)$.
7:     if $u < \exp[-\Delta S]$ then
8:         Set $\boldsymbol{x}^{(t+1)} \mapsfrom \boldsymbol{y}$
9:     else
10:        Set $\boldsymbol{x}^{(t+1)} \mapsfrom \boldsymbol{x}^{(t)}$
11:    end if
12: end if
```

The Markov chain $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots$ is generated by starting from some $\boldsymbol{x}^{(0)}$, which is either a given vector or drawn at random. Since this $\boldsymbol{x}^{(0)}$ is not drawn from the correct distribution, all subsequent samples $\boldsymbol{x}^{(t)}$ are distributed according to some distribution $\pi^{*(t)}$ with $\lim_{t\to\infty} \pi^{*(t)} = \pi^*$. In practice, the first n_{burnin} samples are discarded, and throughout this paper we implicitly assume that $n_{\text{burnin}} \gg 1$ is chosen such that for all subsequent samples the error due to the difference between $\pi^{*(t)}$ and π^* is much smaller than the discretisation- and sampling- errors.

The law of large numbers states that in the limit of a large number of samples $N \gg 1$ the sample average $\widehat{Q}^{\text{StMC}}$ in Eq. (9) is distributed according to a Gaussian $\mathcal{N}(\mu, \sigma)$ with mean $\mu = \mathbb{E}[Q]$ and variance

$$\sigma^2 = \frac{\tau_{\text{int}} \text{Var}[Q]}{N}. \tag{10}$$

In this expression τ_{int} is the integrated autocorrelation time defined as

$$\tau_{\text{int}} = 1 + 2\sum_{s=1}^{\infty} \frac{\mathbb{E}[Q(\boldsymbol{x}^{(t_{\text{meas}})})Q(\boldsymbol{x}^{(t_{\text{meas}}+s)})]}{\mathbb{E}[Q(\boldsymbol{x}^{(t_{\text{meas}})})^2]}, \tag{11}$$

where $t_{\text{meas}} \gg n_{\text{burnin}}$ is an arbitrary point in time. As can be seen from Eq. (10), the number of samples required to reduce the statistical error below a given tolerance grows with τ_{int}, and it is therefore important to reduce the correlation between subsequent samples as far as possible. This can be achieved by carefully choosing the proposal $\boldsymbol{y}$ in line 1 of Alg. 1. In lattice QCD with dynamical fermions, Hybrid Monte Carlo [8] is very popular since it generates global updates. We therefore choose to use this method here, being aware that other algorithms, such as heat bath sampling, might be more efficient for particular applications. We nevertheless believe that HMC is representative, since τ_{int} grows with a large power of the inverse lattice spacing as the continuum limit is approached also with other sampling approaches. The only exception are some problem-specific samplers, such as the cluster algorithm [27] for the topological oscillator, which we therefore also consider in this work.

2.3 Multilevel Monte Carlo

We now describe hierarchical methods for overcoming the growth in autocorrelations and reducing the variance of the measured observable.

Lattice hierarchy

Recall that a path describes the position of the particle at the discrete points $t_j = aj$ with $j = 0, 1, 2, \dots, d-1$. More formally, define a lattice $\mathcal{T}$ as the set of points

$$\mathcal{T} = \{t_j = ja, j = 0, 1, \dots, d-1\}.$$

Paths $\boldsymbol{x}$ on this lattice are objects in the domain $\Omega = \mathcal{D}^d \subset \mathbb{R}^d$. We introduce a hierarchy of L lattices $\mathcal{T}_\ell$ for $\ell = 0, 1, \dots, L-1$, such that lattice $\mathcal{T}_\ell$ has $d_\ell = 2^{\ell-L+1}d$ points and a lattice spacing of $a_\ell = T/d_\ell = 2^{L-1-\ell}a$, i.e.

$$\mathcal{T}_\ell = \{t_j = ja_\ell : j = 0, 1, \dots, d_\ell - 1\}.$$

Here $\mathcal{T}_{L-1} = \mathcal{T}$ is the original lattice with $d_{L-1} = d$ points and a spacing of $a_{L-1} = a$. Paths on lattice $\mathcal{T}_\ell$ are represented by vectors in the domain $\Omega_\ell = \mathcal{D}^{d_\ell} \subset \mathbb{R}^{d_\ell}$, where obviously $\Omega_{L-1} = \Omega$.

Note that the lattices are nested, and the points of the lattice $\mathcal{T}_{\ell-1}$ are a subset of the points of $\mathcal{T}_\ell$, namely the points with even indices. A path on a particular level ℓ stores values at the odd and even lattice points, where the latter are also present on the next-coarser lattice. Formally this can be expressed as

$$\Omega_\ell = \Omega_{\ell-1} \oplus \Omega_{\ell-1} \tag{12}$$

such that all $\boldsymbol{x} \in \Omega_\ell$ can be written as

$$\boldsymbol{x} := [\tilde{\boldsymbol{x}}, \boldsymbol{x}'] \quad \text{with } \tilde{\boldsymbol{x}}, \boldsymbol{x}' \in \Omega_{\ell-1} \text{ and}$$

$$x_j = \begin{cases} x'_{j/2} & \text{for even } j \\ \tilde{x}_{(j-1)/2} & \text{for odd } j. \end{cases} \tag{13}$$

On each lattice we define an action $S_\ell : \Omega_\ell \to \mathbb{R}$ such that $S_{L-1} = S$ is the original action. In the simplest case the coarse-level actions are obtained by re-discretising the original action S with the appropriate lattice spacings, but other choices are possible and will be discussed below. On each level the action induces a probability distribution π_ℓ such that

$$\pi_\ell(\boldsymbol{x}) = \mathcal{Z}_\ell^{-1} \exp\left[-S_\ell(\boldsymbol{x})\right] \quad \text{for all } \boldsymbol{x} \in \Omega_\ell,$$

where $\mathcal{Z}_\ell^{-1}$ is the normalisation constant. The probability distribution π_{L-1} on the finest level is identical to π^* defined in Eq. (8). Further, introduce a conditional probability distribution $\tilde{\pi}_\ell(\cdot|\boldsymbol{x}')$ for the values at the odd points on level ℓ, given the values at the even points on the same level, namely

$$\tilde{\pi}_\ell(\tilde{\boldsymbol{x}}|\boldsymbol{x}') = \tilde{\mathcal{Z}}_\ell(\boldsymbol{x}')^{-1} \exp\left[-\tilde{S}_\ell([\tilde{\boldsymbol{x}}, \boldsymbol{x}'])\right]. \qquad (14)$$

for all $\tilde{\boldsymbol{x}}, \boldsymbol{x}' \in \Omega_{\ell-1}$. The action $\tilde{S}_\ell$ should be some approximation to S_ℓ, such that it is possible to sample from $\tilde{\pi}_\ell$ for a given $\boldsymbol{x}'$. For the quantum mechanical model problems considered in this work the construction of $\tilde{S}_\ell$ is described in Sections 3.1.1 and 3.2.1.

We stress that although in this paper we assume that the lattice can be partitioned into sets of mutually independent even and odd sites, the ideas developed here can be generalised to higher dimensions. This is outlined in Appendix A.

Hierarchical sampling

Similar to the delayed-acceptance approach in [15], we next introduce a hierarchical algorithm to efficiently construct a Markov chain on a given level ℓ using coarser levels: First we define the two-level Metropolis-Hastings step in Alg. 2. Setting $\boldsymbol{x}_\ell^{(t)} = [\tilde{\boldsymbol{x}}_\ell^{(t)}, \boldsymbol{x}_{\ell-1}^{(t)}]$ this algorithm assumes that on a given level ℓ there is a coarse level proposal distribution $q_{\ell-1}(\cdot|\boldsymbol{x}_{\ell-1}^{(t)})$ which depends on $\boldsymbol{x}_{\ell-1}^{(t)}$. Based on this, it proposes a new fine-level state which is either accepted and returned as the new state $\boldsymbol{x}_\ell^{(t+1)}$ or rejected; in the latter case the previous state $\boldsymbol{x}_\ell^{(t)}$ is returned as $\boldsymbol{x}_\ell^{(t+1)}$. It was shown in [15] that this defines a correct Metropolis-Hastings algorithm targeting π_ℓ.

Let $q_\ell^{(\mathrm{TL})}(\boldsymbol{x}_\ell^{(t+1)}|\boldsymbol{x}_\ell^{(t)})$ be the transition kernel for the process $\boldsymbol{x}_\ell^{(t)} \to \boldsymbol{x}_\ell^{(t+1)}$ implicitly defined by Alg. 2. The key idea is now to use the algorithm recursively by using $q_{\ell-1}^{(\mathrm{TL})}$ as the proposal distribution $q_{\ell-1}$ on level $\ell-1$. The process of picking $\boldsymbol{y}_{\ell-1}$ from $q_{\ell-1}(\cdot|\boldsymbol{x}_{\ell-1}^{(t)}) = q_{\ell-1}^{(\mathrm{TL})}(\cdot|\boldsymbol{x}_{\ell-1}^{(t)})$ in the first line of Alg. 2 then corresponds to a recursive call to the same algorithm on the next-coarser level. On the coarsest level ($\ell = 0$) $\boldsymbol{y}_0$ is drawn with the standard Metropolis-Hastings step in Alg. 1 with corresponding transition kernel $q_0^{(\mathrm{MH})}(\cdot|\boldsymbol{x}_0^{(t)})$; here we always assume that the proposal in this Metropolis Hastings step is generated with a symmetric method such as HMC.

Algorithm 2 Two-level Metropolis Hastings step.
Input: level ℓ, current sample $\boldsymbol{x}_\ell^{(t)} \sim \pi_\ell$, proposal distribution $q_{\ell-1}$
Output: new sample $\boldsymbol{x}_\ell^{(t+1)} \sim \pi_\ell$

```
1: Let $\boldsymbol{x}_\ell^{(t)} = [\tilde{\boldsymbol{x}}_\ell^{(t)}, \boldsymbol{x}_{\ell-1}^{(t)}]$ and pick $\boldsymbol{y}_{\ell-1}$ from $q_{\ell-1}(\cdot|\boldsymbol{x}_{\ell-1}^{(t)})$.
2: if $\boldsymbol{x}_{\ell-1}^{(t+1)} = \boldsymbol{x}_{\ell-1}^{(t)}$ (coarse level proposal rejected) then
3:     Set $\boldsymbol{x}_\ell^{(t+1)} \mapsfrom \boldsymbol{x}_\ell^{(t)}$
4: else
5:     Pick $\tilde{\boldsymbol{y}}_\ell$ from $\tilde{\pi}_\ell(\cdot|\boldsymbol{y}_{\ell-1})$ and let $\boldsymbol{y}_\ell = [\tilde{\boldsymbol{y}}_\ell, \boldsymbol{y}_{\ell-1}]$.
6:     Compute
           $$\frac{\pi_\ell(\boldsymbol{y}_\ell)}{\pi_\ell(\boldsymbol{x}_\ell^{(t)})} \cdot \frac{\tilde{\pi}_\ell(\tilde{\boldsymbol{x}}_\ell^{(t)}|\boldsymbol{x}_{\ell-1}^{(t)})}{\tilde{\pi}_\ell(\tilde{\boldsymbol{y}}_\ell|\boldsymbol{y}_{\ell-1})} \cdot \frac{\pi_{\ell-1}(\boldsymbol{x}_{\ell-1}^{(t)})}{\pi_{\ell-1}(\boldsymbol{y}_{\ell-1})} = \exp[-\Delta S_\ell]$$
       with
           $$\begin{aligned} \Delta S_\ell := & \; S_\ell(\boldsymbol{y}_\ell) - S_\ell(\boldsymbol{x}_\ell^{(t)}) \\ + & \; \tilde{S}_\ell([\tilde{\boldsymbol{x}}_\ell^{(t)}, \boldsymbol{x}_{\ell-1}^{(t)}]) - \tilde{S}_\ell([\tilde{\boldsymbol{y}}_\ell, \boldsymbol{y}_{\ell-1}]) \\ + & \; S_{\ell-1}(\boldsymbol{x}_{\ell-1}^{(t)}) - S_{\ell-1}(\boldsymbol{y}_{\ell-1}) \\ + & \; \log \tilde{\mathcal{Z}}_\ell(\boldsymbol{x}_{\ell-1}^{(t)}) - \log \tilde{\mathcal{Z}}_\ell(\boldsymbol{y}_{\ell-1}). \end{aligned}$$
7:     if $\Delta S_\ell < 0$ then
8:         Set $\boldsymbol{x}_\ell^{(t+1)} \mapsfrom \boldsymbol{y}_\ell$
9:     else
10:        Draw uniformly distributed random $u \in [0, 1)$.
11:        if $u < \exp[-\Delta S_\ell]$ then
12:            Set $\boldsymbol{x}_\ell^{(t+1)} \mapsfrom \boldsymbol{y}_\ell$
13:        else
14:            Set $\boldsymbol{x}_\ell^{(t+1)} \mapsfrom \boldsymbol{x}_\ell^{(t)}$
15:        end if
16:    end if
17: end if
```

More specifically, to construct a sequence of samples $\boldsymbol{x}_\ell^{(0)}, \boldsymbol{x}_\ell^{(1)}, \boldsymbol{x}_\ell^{(2)}, \dots \in \Omega_\ell$ distributed according to π_ℓ we use Alg. 3, which is illustrated schematically in Fig. 1.

Algorithm 3 Hierarchical delayed-acceptance sampler (recursive implementation).
Input: level ℓ, current sample $\boldsymbol{x}_\ell^{(t)} \sim \pi_\ell$
Output: new sample $\boldsymbol{x}_\ell^{(t+1)} \sim \pi_\ell$

```
1: Generate $\boldsymbol{x}_\ell^{(t+1)}$ using Alg. 2 with level $\ell$, current sample
   $\boldsymbol{x}_\ell^{(t)} \sim \pi_\ell$ and proposal distribution
       $$q_{\ell-1}(\cdot|\boldsymbol{x}_{\ell-1}^{(t)}) = \begin{cases} q_0^{(\mathrm{MH})}(\cdot|\boldsymbol{x}_0^{(t)}) & \text{for } \ell = 1 \\ q_{\ell-1}^{(\mathrm{TL})}(\cdot|\boldsymbol{x}_{\ell-1}^{(t)}) & \text{for } \ell = 2, 3, \dots, L-1 \end{cases}$$
```

Note that $\boldsymbol{x}_\ell^{(t+1)} = \boldsymbol{x}_\ell^{(t)}$ unless the proposals on all levels $0, 1, \dots, \ell$ get accepted. At first sight this seems to imply that the overall acceptance probability of Alg. 3 drops as the number of levels increases, and subsequent samples are highly correlated. However, this turns out not to be the case if the theories on subsequent level converge with $\ell \to \infty$: in this case the proposal from the two-level step in Alg. 2 is almost certainly accepted on finer levels. Our numerical experiments confirm this observation.

In practice, it is more convenient to implement Alg. 3 iteratively, starting from the coarsest level. As discussed

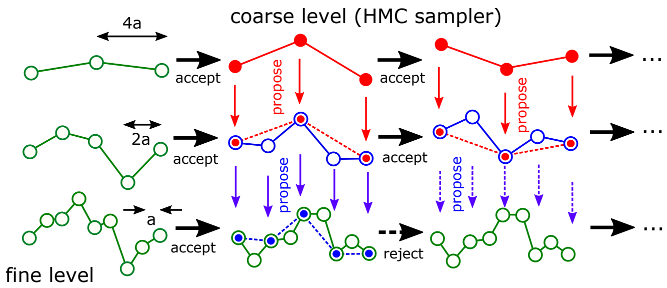

Figure 1: Hierarchical sampling, as described in Alg. 3, for $L = 3$ levels.

in Appendix B, the cost of executing Alg. 3 on level ℓ can be bounded by a constant times the number of unknowns d_ℓ on this particular level. Observe also that setting $\ell = L-1$ in Alg. 3 allows drawing a new sample $\boldsymbol{x}^{(t+1)} \sim \pi^*$ from the original fine level probability distribution defined in Eq. (8).

Relationship to the literature. The two-level step in Alg. 2 is closely related to similar algorithms in [15, 17]. If the coarse level sample is drawn with an arbitrary Metropolis-Hastings kernel $q_{\ell-1}(\cdot|\boldsymbol{x}_{\ell-1}^{(t)}) = q_{\ell-1}^{(\mathrm{MH})}(\cdot|\boldsymbol{x}_{\ell-1}^{(t)})$, then Alg. 2 above is a variant of the delayed-acceptance method in [15, Alg. 1] with proposal distribution $q(\boldsymbol{y}_\ell|\boldsymbol{x}_\ell^{(t)}) = \tilde{\pi}_\ell(\tilde{\boldsymbol{y}}_\ell|\boldsymbol{y}_{\ell-1})q_{\ell-1}^{(\mathrm{MH})}(\boldsymbol{y}_{\ell-1}|\boldsymbol{x}_{\ell-1}^{(t)})$ and approximation $f_x^*(\boldsymbol{y}_\ell) = \tilde{\pi}_\ell(\tilde{\boldsymbol{y}}_\ell|\boldsymbol{y}_{\ell-1})\pi_{\ell-1}(\boldsymbol{y}_{\ell-1})$, recalling the notation $\boldsymbol{x}_\ell^{(t)} = [\tilde{\boldsymbol{x}}_\ell^{(t)}, \boldsymbol{x}_{\ell-1}^{(t)}]$, $\boldsymbol{y}_\ell = [\tilde{\boldsymbol{y}}_\ell, \boldsymbol{y}_{\ell-1}]$.

On the other hand, if the coarse level sample is drawn from the exact coarse level distribution, i.e. if $q(\cdot|\boldsymbol{x}_{\ell-1}^{(t)}) = \pi_{\ell-1}(\cdot)$, Alg. 2 is identical to [17, Alg. 2].

Multilevel Monte Carlo algorithm

As discussed in the introduction, the multilevel Monte Carlo algorithm computes the quantity of interest Q_0 on the coarsest level and adds corrections to this by computing the *difference* Y_ℓ of the observable on subsequent levels $\ell = 1, 2, \ldots, L-1$ according to the telescoping sum in Eq. (3). Since those differences Y_ℓ have a smaller variance, this allows shifting the cost to the coarser levels where samples can be generated cheaply. The original MLMC algorithm described in [9] assumes that it is possible to draw independent identically distributed (i.i.d.) samples from a distribution on each level. For the Markov chain Monte Carlo setting considered here this is not possible since subsequent samples in the chain are correlated and, as discussed in [17], this introduces an additional bias. This bias can be reduced by constructing sequences $\boldsymbol{z}_\ell^{(0)}, \boldsymbol{z}_\ell^{(1)}, \boldsymbol{z}_\ell^{(2)}, \ldots$ of samples for each level $\ell = 0, \ldots, L-1$ with Alg. 3 and sampling those sequences with sufficiently large sub-sampling rates t_ℓ. The typical rule in statistics is to use twice the integrated autocorrelation time $\tau_{\mathrm{int},\ell}$ to achieve (sufficient) independence. In our numerical experiments, we set $t_\ell = \lceil 2\tau_{\mathrm{int},\ell} \rceil$ and observe that the additional bias due to computing the coarse level samples which are only approximately independent is comparable to the discretisation error.

The multilevel Monte Carlo algorithm which we use in this work is presented in Alg. 4 and visualised in Fig. 2. It is similar to the multilevel algorithm in [17], but with the recursive independent sampler in [17, Alg. 3] replaced by the (suitably sub-sampled) hierarchical delayed-acceptance sampler in our Alg. 3 above. Multilevel Monte Carlo computes

$$\widehat{Q}_{L,\{N_\ell^{\mathrm{eff}}\}}^{\mathrm{MLMC}} = \sum_{\ell=0}^{L-1} \widehat{Y}_{\ell,N_\ell^{\mathrm{eff}}} \quad \text{with} \quad \widehat{Y}_{\ell,N_\ell^{\mathrm{eff}}} = \frac{1}{N_\ell^{\mathrm{eff}}}\sum_{j=1}^{N_\ell^{\mathrm{eff}}} Y_\ell^{(j)}, \tag{15}$$

which as unbiased estimator for the expectation $\mathbb{E}[Q]$ in Eq. (3). On each level ℓ, the number of samples is chosen to be

$$N_\ell^{\mathrm{eff}} = \max\left\{1\,,\ \epsilon_{\mathrm{stat}}^{-2}\left(\sum_{\ell=0}^{L-1}\sqrt{V_\ell \mathcal{C}_\ell^{\mathrm{eff}}}\right)\sqrt{\frac{V_\ell}{\mathcal{C}_\ell^{\mathrm{eff}}}}\right\}, \tag{16}$$

where $\mathcal{C}_\ell^{\mathrm{eff}}$ is the effective cost of generating an independent sample (taking into account autocorrelations) and $V_\ell = \mathrm{Var}[Y_\ell]$ is the variance of the quantity Y_ℓ on level ℓ, which converges to zero as $\ell \to \infty$.

Algorithm 4 Multilevel Monte Carlo.
Input: Number of levels L, number of samples per level N_ℓ^{eff} and sub-sampling rates t_ℓ for $\ell = 0, \ldots, L-1$
Output: MLMC estimate for QoI.

```
for level $\ell = 0, \ldots, L-1$ do
    for $j = 1, \ldots, N_\ell^{\mathrm{eff}}$ do
        if $\ell = 0$ then
            Create a new sample $\boldsymbol{x}_0^{(t+t_0)}$ from $\boldsymbol{x}_0^{(t)}$ with a standard Metropolis-Hastings method.
            Compute $Y_0^{(j)} = Q_0(\boldsymbol{x}_0^{(t+t_0)})$
        else
            Create a new sample $\boldsymbol{x}_\ell^{(t+1)}$ from $\boldsymbol{x}_\ell^{(t)}$ with Alg. 2 and $q_{\ell-1}(\cdot|\boldsymbol{x}_\ell^{(t)}) = \pi_{\ell-1}$ ;
            In practice, use $t_{\ell-1}$ steps of Alg. 3 to compute an approximately independent sample $\boldsymbol{z}_{\ell-1}^{(t+t_{\ell-1})}$ on level $\ell-1$.
            Compute $Y_\ell^{(j)} = Q_\ell(\boldsymbol{x}_\ell^{(t+1)}) - Q_{\ell-1}(\boldsymbol{z}_{\ell-1}^{(t+t_{\ell-1})})$.
        end if
    end for
end for
Compute the MLMC estimator defined in Eq. (15).
```

We now discuss how the cost of Alg. 4 increases as the tolerance on the total error is tightened, assuming for simplicity i.i.d. samples on all levels. Let the exact value of the observable in the continuum limit be Q^{exact}. The total mean square error of the multilevel Monte Carlo estimator defined in Eq. (15) can be expanded

$$\begin{aligned}&\mathbb{E}\left[\left(\widehat{Q}_{L,\{N_\ell^{\mathrm{eff}}\}}^{\mathrm{MLMC}} - Q^{\mathrm{exact}}\right)^2\right]\\ &= \mathrm{Var}\left[\widehat{Q}_{L,\{N_\ell^{\mathrm{eff}}\}}^{\mathrm{MLMC}}\right] + \left(\mathbb{E}\left[\widehat{Q}_{L,\{N_\ell^{\mathrm{eff}}\}}^{\mathrm{MLMC}}\right] - Q^{\mathrm{exact}}\right)^2,\end{aligned} \tag{17}$$

where the first term in the final line of Eq. (17) is the squared statistical error, whereas the second term is the

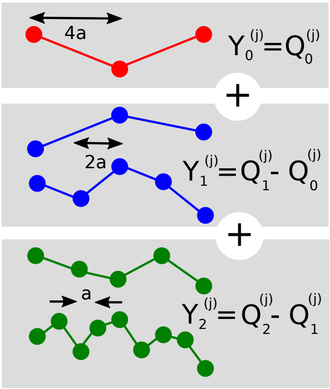

Figure 2: Schematic visualisation of Multilevel Monte Carlo, as described in Alg. 4, for $L = 3$ levels.

squared discretisation error. An easy calculation shows that choosing N_ℓ^{eff} as in Eq. (16) guarantees that

$$\text{Var}\left[\widehat{Q}^{\text{MLMC}}_{L,\{N_\ell^{\text{eff}}\}}\right] = \sum_{\ell=0}^{L-1} \frac{V_\ell}{N_\ell^{\text{eff}}} \le \epsilon_{\text{stat}}^2.$$

To analyse the complexity we assume that

(i) the discretisation error is of order $\mathcal{O}(a_\ell^\alpha)$,

(ii) V_ℓ converges with order $\mathcal{O}(a_\ell^\beta)$ for some $\beta > 0$,

(iii) the integrated autocorrelation times of Y_ℓ, and thus also the sub-sampling rates t_ℓ, can be bounded by a constant independent of ℓ such that the cost $\mathcal{C}_\ell^{\text{eff}}$ of generating an independent sample does not grow faster than the number of unknowns d_ℓ for all ℓ.

As shown in more detail in Appendix B, it is then possible to choose the number of levels L such that the discretisation error in Eq. (17) does not exceed ϵ_{disc}. As a consequence, the cost $\mathcal{C}_{\text{MLMC}}(\epsilon_{\text{disc}}, \epsilon_{\text{stat}})$ of computing the MLMC estimator in Eq. (15) with a statistical error less than ϵ_{stat} and a discretisation error less than ϵ_{disc} has the following computational complexity:

$$\mathcal{C}_{\text{MLMC}} = \begin{cases} \mathcal{O}\left(\epsilon_{\text{stat}}^{-2} + \epsilon_{\text{disc}}^{-1/\alpha}\right) & \text{for } \beta > 1, \\ \mathcal{O}\left(\epsilon_{\text{stat}}^{-2}|\log \epsilon_{\text{disc}}|^2 + \epsilon_{\text{disc}}^{-1/\alpha}\right) & \text{for } \beta = 1, \\ \mathcal{O}\left(\epsilon_{\text{stat}}^{-2}\epsilon_{\text{disc}}^{-\frac{1-\beta}{\alpha}} + \epsilon_{\text{disc}}^{-1/\alpha}\right) & \text{for } \beta < 1. \end{cases} \tag{18}$$

For the choice $\epsilon_{\text{disc}} = \epsilon_{\text{stat}} = \epsilon/\sqrt{2}$, the total mean square error in Eq. (17) does not exceed ϵ^2 and Eq. (18) becomes

$$\mathcal{C}_{\text{MLMC}}(\epsilon) = \begin{cases} \mathcal{O}\left(\epsilon^{-2}\right) & \text{for } \beta > 1 \\ \mathcal{O}\left(\epsilon^{-2}|\log \epsilon|^2\right) & \text{for } \beta = 1 \\ \mathcal{O}\left(\epsilon^{-2-\frac{1-\beta}{\alpha}}\right) & \text{for } \beta < 1 \end{cases}, \tag{19}$$

which is a special case of the well-known estimate in [9].

However, the samples created by Alg. 4 on each of the levels ℓ are generated with a Markov chain and thus only asymptotically distributed according to π_ℓ. As discussed in [17], the complexity analysis can be modified to address this issue, leading to an additional factor $|\log \epsilon_{\text{disc}}|$ in Eqs. (18) and (19). This seems to be not visible in the numerical results below or in [17] (at least pre-asymptotically).

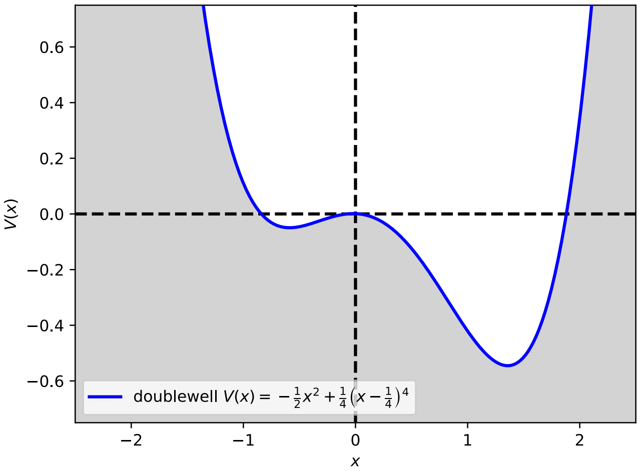

Figure 3: Double well potential used for numerical experiments

2.4 Memory requirements

Although the one dimensional quantum mechanical problems considered here do not require significant storage, the memory requirements of the algorithms introduced in this paper need to be considered in addition to their runtimes. This is particularly important for simulations of higher dimensional quantum field theories on modern many-core architectures where the memory per compute core is limited.

As discussed in detail in Appendix C, on a given level the hierarchical sampler in Alg. 3 requires less memory than a standard Metropolis Hastings method with a HMC proposal distribution. The memory footprint of the multilevel Monte Carlo method in Alg. 4 is less than three times that of a HMC based Metropolis Hastings algorithm.

3 Quantum mechanical model systems

To demonstrate the performance of the methods discussed in the previous section we consider two non-trivial quantum mechanical problems.

3.1 Non-symmetric double-well potential

The first system describes a particle with mass m_0 moving subject to a non-symmetric double-well potential $V(x) = \frac{m_0\mu^2}{2}x^2 + \frac{\lambda}{4}(x-\eta)^4$. Fig. 3 shows this potential for the choice of parameters that were used in our numerical experiments, namely $m_0 = 1$, $\mu^2 = -1$, $\lambda = 1$, $\eta = \frac{1}{4}$. In the Euclidean time formulation of the path integral the corresponding Lagrangian is

$$\mathcal{L}(x(t)) = \frac{m_0}{2}\left(\frac{dx}{dt}\right)^2 + \frac{m_0\mu^2}{2}x^2 + \frac{\lambda}{4}(x-\eta)^4 \tag{20}$$

where $x(t) \in \mathbb{R}$. For a given path $\boldsymbol{x} = (x_0, x_1, \dots, x_{d-1}) \in \mathbb{R}^d$ the discretised lattice action is

$$\begin{aligned} S(\boldsymbol{x}) = a\sum_{j=0}^{d-1}\Bigg\{&\frac{m_0}{2}\left(\frac{x_j - x_{j-1}}{a}\right)^2 \\ &+ \frac{m_0\mu^2}{2}x_i^2 + \frac{\lambda}{4}(x_i-\eta)^4\Bigg\}. \end{aligned} \tag{21}$$

The observable we consider is the average squared displacement

$$Q(\boldsymbol{x}) = \frac{1}{d}\sum_{j=0}^{d-1} x_j^2. \tag{22}$$

Note that since points on the lattice are correlated with a correlation length which is constant in physical units, the variance of this observable does not go to zero in the continuum limit. In other words, the sampling error is not automatically reduced on finer lattices.

3.1.1 Coarse level action

Coarse grained versions S_ℓ of the action in Eq. (21) are obtained by re-discretising the Lagrangian in Eq. (20) on the lattice $\mathcal{T}_\ell$ with $d_\ell = 2^{\ell-L+1}d$ points and lattice spacing $a_\ell = 2^{L-1-\ell}a$ on level ℓ to obtain

$$\begin{aligned} S_\ell(\boldsymbol{x}) = a_\ell\sum_{j=0}^{d_\ell-1}\Bigg\{&\frac{m_0}{2}\left(\frac{x_j - x_{j-1}}{a_\ell}\right)^2 \\ &+ \frac{m_0\mu^2}{2}x_j^2 + \frac{\lambda}{4}(x_j-\eta)^4\Bigg\}. \end{aligned}$$

To construct the action $\tilde{S}_\ell$ defined in Eq. (14), observe that

$$\begin{aligned} \pi_\ell(\boldsymbol{x}) = \pi_\ell^{\text{even}}&(x_0, x_2, \dots, x_{d_\ell-2}) \times \\ &\times \prod_{j=0}^{d_{\ell-1}-1}\pi_\ell^{\text{odd}}(x_{2j+1}|x_{2j}, x_{2j+2}) \end{aligned} \tag{23}$$

where π_ℓ^{even} is the marginal distribution of the even points

$$\pi_\ell^{\text{even}}(x_0, x_2, \dots, x_{d_\ell-2}) = \int_{\mathcal{D}}\cdots\int_{\mathcal{D}}\pi_\ell(\boldsymbol{x})dx_1\,dx_3\dots dx_{d_\ell-1}$$

and

$$\pi_\ell^{\text{odd}}(x_{2j+1}|x_{2j}, x_{2j+2}) = \mathcal{Z}_{\ell,j}^{-1}\exp\left[-W_\ell(x_{2j+1}|x_{2j}, x_{2j+2})\right]. \tag{24}$$

Here W_ℓ is defined for arbitrary values x_-, x_+ as

$$\begin{aligned} W_\ell(x|x_-, x_+) = &\frac{m_0}{a_\ell}\left(x^2 - (x_- + x_+)x\right) \\ &+ \; a_\ell\left(\frac{m_0\mu^2}{2}x^2 + \frac{\lambda}{4}(x-\eta)^4\right). \end{aligned}$$

and $\mathcal{Z}_{\ell,j} = \mathcal{Z}_{\ell,j}(x_{2j}, x_{2j+2})$ is a normalisation constant which depends on x_{2j}, x_{2j+2}. The distribution in Eq. (24) can be approximated by a Gaussian by writing

$$W_\ell(x|x_-, x_+) \approx G_\ell(x|x_-, x_+)$$

with

$$G_\ell(x|x_-, x_+) = \frac{m_0}{a_\ell}\sigma_\ell(x_-, x_+)\left(x - \zeta_\ell(x_-, x_+)\right)^2,$$

where $\zeta_\ell = \zeta_\ell(x_-, x_+)$ is the minimum of $W_\ell(x|x_-, x_+)$ and satisfies the non-linear equation

$$\left(1 + \frac{1}{2}a_\ell^2\mu^2\right)\zeta_\ell + a_\ell^2\frac{\lambda}{2m_0}(\zeta_\ell - \eta)^3 = \frac{x_- + x_+}{2}. \tag{25}$$

In the code ζ_ℓ is found by a small number of fixed point iterations of Eq. (25), using $\overline{x} = (x_- + x_+)/2$ as a starting guess. Further, $2m_0\sigma_\ell/a_\ell$ is the curvature of the function W_ℓ, evaluated at the point $\overline{x} \approx \zeta_\ell(x_-, x_+)$, i.e.

$$\begin{aligned} \frac{2m_0}{a_\ell}\sigma_\ell(x_-, x_+) &= \frac{\partial^2 W_\ell}{\partial x^2}(\overline{x}|x_-, x_+) \\ &= \frac{2m_0}{a_\ell}\left(1 + \frac{1}{2}a_\ell^2\left(\mu^2 + \frac{3\lambda}{m_0}(\overline{x} - \eta)^2\right)\right) \\ &\approx \frac{\partial^2 W_\ell}{\partial x^2}(\zeta_\ell(x_-, x_+)|x_-, x_+). \end{aligned}$$

Now write $\boldsymbol{x} = [\tilde{\boldsymbol{x}}, \boldsymbol{x}']$ as in Eq. (13). Given $\zeta_\ell(x'_j, x'_{j+1})$ and $\sigma_\ell(x'_j, x'_{j+1})$ for all $j = 0, 1, \dots, d_{\ell-1}-1$, we can then construct

$$\tilde{S}_\ell([\tilde{\boldsymbol{x}}, \boldsymbol{x}']) = \sum_{j=0}^{d_{\ell-1}-1} G_\ell(\tilde{x}_j|x'_j, x'_{j+1}).$$

The resulting probability density $\tilde{\pi}_\ell(\cdot|\boldsymbol{x}')$ defined in Eq. (14) is a multivariate normal distribution with diagonal covariance matrix, which can be easily sampled; the normalisation constant in Eq. (14) is

$$\tilde{\mathcal{Z}}_\ell(\boldsymbol{x}') = \sqrt{\left(\frac{4\pi m_0}{a_\ell}\right)^{d_{\ell-1}}\prod_{j=0}^{d_{\ell-1}-1}\sigma_\ell(x'_j, x'_{j+1})}.$$

3.2 Topological oscillator

The second model system is the topological oscillator, described for example in [26]. This is an interesting problem since it has a topological quantum number which can only take on integer values. The Lagrangian is

$$\mathcal{L}(x, t) = \frac{I_0}{2}\left(\frac{dx}{dt}\right)^2 \tag{26}$$

where now crucially $x \in [-\pi, \pi)$, i.e. the particle is confined to a finite interval. The Lagrangian in Eq. (26) can be obtained from the action of a free particle with mass m_0 confined to a circle with radius R,

$$\mathcal{L}(y, z, t) = \frac{m_0}{2}\left(\left(\frac{dy}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2\right),$$

with $(y, z) \in \mathbb{R}^2$, $y^2 + z^2 = R^2$ by setting $y(t) = R\cos(x(t))$, $z(t) = R\sin(x(t))$ and $I_0 = R^2 m_0$. The form of the discretised action chosen here is

$$S(\boldsymbol{x}) = \frac{I_0}{a}\sum_{j=0}^{d-1}\left(1 - \cos(x_j - x_{j-1})\right).$$

As above we used periodic boundary conditions $x_d = x_0$. Note that

$$\frac{1 - \cos(x_j - x_{j-1})}{a^2} = \frac{1}{2}\left(\frac{dx}{dt}\right)^2 + \mathcal{O}\left(a^2\right).$$

For a given path $x(t)$ the topological charge $q(x)$ of the system describes the number of complete revolutions during the time period T. Mathematically it is defined as

$$q(x(t)) = \frac{1}{2\pi}\int_0^T \frac{dx(t)}{dt}\,dt \in \mathbb{Z}.$$

For the discretised system this becomes

$$q(\boldsymbol{x}) = \frac{1}{2\pi}\sum_{j=0}^{d-1}\Big\{(x_j - x_{j-1}) \mod [-\pi, \pi)\Big\} \in \mathbb{Z}.$$

Following the notation in [26], for any $x \in \mathbb{R}$ the quantity $z = x \mod [-\pi, \pi)$ is defined as $z = x + 2\pi k$ with $k \in \mathbb{Z}$ such that $-\pi \leq z < \pi$. The observable we consider is the topological susceptibility

$$Q(\boldsymbol{x}) = \chi_t(\boldsymbol{x}) = \frac{q^2(\boldsymbol{x})}{T}. \tag{27}$$

Defining $\xi := T/I_0$ and $z := a/I_0$, a tedious but straightforward calculation shows that the expectation value of χ_t for finite a, T is given by

$$\mathbb{E}[\chi_t] = \frac{1}{4\pi^2 I_0}\left(1 - \xi\widehat{\Sigma}_2(\xi) + \left[\frac{1}{2} - \xi\widehat{\Sigma}_2(\xi) + \frac{1}{4}\xi^2\left(\widehat{\Sigma}_4(\xi) - \widehat{\Sigma}_2(\xi)^2\right)\right]z\right) + \mathcal{O}(z^2) \xrightarrow{a\to 0} \frac{1 - \xi\widehat{\Sigma}_2(\xi)}{4\pi^2 I_0} \xrightarrow{T\to\infty} \frac{1}{4\pi^2 I_0}, \tag{28}$$

where for any $p \in \mathbb{N}$, $\xi > 0$ the function $\widehat{\Sigma}_p$ is defined as

$$\Sigma_p(\xi) := \sum_{m\in\mathbb{Z}} m^p \exp\left[-\frac{1}{2}\xi m^2\right], \quad \widehat{\Sigma}_p(\xi) := \frac{\Sigma_p(\xi)}{\Sigma_0(\xi)}. \tag{29}$$

Eq. (28) allows the calculation of the constant Δ_0 in the Taylor expansion $\mathbb{E}[\chi_t] = \mathbb{E}[\chi_t(a = 0)] + \Delta_0 a + \mathcal{O}(a^2)$ of the topological susceptibility. In other words, we can work out the bias for a given lattice spacing. This will also allow us to balance the discretisation- and statistical-errors in the MLMC estimator if we choose $\epsilon_{\text{disc}} = \epsilon_{\text{stat}}$. In the continuum limit $(a \to 0)$ the variance of χ_t can be shown to be

$$\operatorname{Var}[\chi_t] = \mathbb{E}\left[(\chi_t - \mathbb{E}[\chi_t])^2\right] = \frac{R(4\pi^2/\xi)}{8\pi^4 I_0^2} \xrightarrow{T\to\infty} \frac{1}{8\pi^4 I_0^2} \tag{30}$$

with the function R defined by

$$R(\zeta) := \frac{1}{2}\zeta^2\left(\widehat{\Sigma}_4(\zeta) - \widehat{\Sigma}_2(\zeta)^2\right).$$

3.2.1 Coarse level action

For the topological oscillator the coarse level action is

$$S_\ell(\boldsymbol{x}) = \frac{I_0^{(\ell)}}{a_\ell}\sum_{j=0}^{d_\ell - 1}\left(1 - \cos(x_j - x_{j-1})\right),$$

where the moment of inertia $I_0^{(\ell)}$ is level dependent. In the simplest case one could simply set $I_0^{(\ell)} = I_0$ for all $\ell = 0, 1, \dots, L-1$. However, as will be shown below, performance can be improved significantly by using a perturbative matching procedure to construct $I_0^{(\ell)}$ on the coarser levels. To obtain $\tilde{S}_\ell$, rewrite π_ℓ as in Eqs. (23) and (24), where now

$$W_\ell(x|x_-, x_+) = \overline{W}_\ell(x|x_-, x_+) + 2 - \frac{1}{2}\sigma_\ell(x_-, x_+)$$

with

$$\begin{aligned}
\overline{W}_\ell(x|x_-, x_+) &= \frac{I_0^{(\ell)}}{a_\ell}\sigma_\ell(x_-, x_+)\sin^2\left(\frac{x - \zeta_\ell(x_-, x_+)}{2}\right) \\
\sigma_\ell(x_-, x_+) &= 4\left|\cos\left(\frac{x_+ - x_-}{2}\right)\right|, \\
\tan\zeta_\ell(x_-, x_+) &= \frac{\sin(x_+) + \sin(x_-)}{\cos(x_+) + \cos(x_-)}.
\end{aligned}$$

Again write $\boldsymbol{x} = [\tilde{\boldsymbol{x}}, \boldsymbol{x}']$ as in Eq. (13), and given $\zeta_\ell(x'_j, x'_{j+1})$ and $\sigma_\ell(x'_j, x'_{j+1})$ for all $j = 0, 1, \dots, d_{\ell-1} - 1$ construct

$$\tilde{S}_\ell([\tilde{\boldsymbol{x}}, \boldsymbol{x}']) = \sum_{j=0}^{d_{\ell-1}-1}\overline{W}_\ell(\tilde{x}_j|x'_j, x'_{j+1}).$$

The normalisation constant in Eq. (14) is

$$\begin{aligned}
\mathcal{Z}_\ell(\boldsymbol{x}') = (2\pi)^{d_{\ell-1}}\exp&\left[-\frac{I_0^{(\ell)}}{2a_\ell}\sum_{j=0}^{d_{\ell-1}-1}\sigma_\ell(x'_j, x'_{j+1})\right] \\
&\times\prod_{j=0}^{d_{\ell-1}-1}B_0\left(\frac{I_0^{(\ell)}}{2a_\ell}\sigma_\ell(x'_j, x'_{j+1})\right)
\end{aligned}$$

where B_0 is the zero-order modified Bessel function of the first kind. The resulting probability density $\tilde{\pi}_\ell$ is the product of one-dimensional densities of the form

$$\begin{aligned}
p_{\sigma,\delta x}(x) &= \mathcal{Z}_\sigma^{-1}\exp\left[-2\sigma\sin^2\left(\frac{x - \delta x}{2}\right)\right] \quad \text{with} \\
\mathcal{Z}_\sigma &= 2\pi e^{-\sigma}B_0(\sigma),
\end{aligned} \tag{31}$$

which can be easily sampled for arbitrary values of σ and δx. In our code we find that rejection sampling with a suitable Gaussian envelope (as described in Appendix D) gives good results.

3.2.2 Coarse level matching

Ideally, the coarse level actions should be obtained by recursively integrating out the modes that can be represented on a given lattice, but not on the next coarser one. In other words, $S_{\ell-1}$ is an effective action obtained from S_ℓ. While for an arbitrary action this can not be done exactly, an approximate effective action can be constructed by a perturbative renormalisation group transformation or through (approximate) matching. Here we follow the latter procedure for the topological oscillator to adjust the moment of inertia $I_0^{(\ell)}$ on the coarser levels, starting from the physical value $I_0 = I_0^{(L-1)}$ on the finest lattice. Let $\chi_t(a, I_0, T)$ be the topological susceptibility calculate for a given I_0, T and lattice spacing a, and recall that we can compute $\chi_t(a, I_0, T)$ up to corrections of $\mathcal{O}((a/I_0)^2)$. We now require that

$$\chi_t(a_{\ell-1}, I_0^{(\ell-1)}, T) = \chi_t(a_\ell, I_0^{(\ell)}, T) + \mathcal{O}((a_\ell/I_0^{(\ell)})^2)$$

for all $\ell = 1, \dots, L-1$. Using Eq. (28) this gives

$$I_0^{(\ell-1)} = \left(1 + \frac{a_\ell}{I_0^{(\ell)}} \cdot \delta_I\left(T/I_0^{(\ell)}\right)\right) I_0^{(\ell)} + \mathcal{O}((a_\ell/I_0^{(\ell)})^2).$$

with

$$\delta_I(\xi) = \frac{1}{2} \cdot \frac{1 - 2\xi\widehat{\Sigma}_2(\xi) + \frac{1}{2}\xi^2\left(\widehat{\Sigma}_4(\xi) - \widehat{\Sigma}_2(\xi)^2\right)}{1 - 2\xi\widehat{\Sigma}_2(\xi) + \xi^2\left(\widehat{\Sigma}_4(\xi) - \widehat{\Sigma}_2(\xi)^2\right)}$$

and $\widehat{\Sigma}_p$ as defined in Eq. (29). As the following numerical results show, computing $I_\ell^{(0)}$ with this approximate coarse level matching procedure significantly improves performance both for the hierarchical sampler in Alg. 3 and the MLMC method in Alg. 4.

4 Results

We now quantify the performance gains of the numerical algorithms described above. All results were generated with a C++ code developed by the authors which is freely available at

https://bitbucket.org/em459/mlmcpathintegral/

The reported runtimes were obtained by running a sequential version of the code (which was compiled with version 18.5.274 of the Intel C compiler) on a single core of an Intel E5-2650 v2 (2.60 GHz) CPU.

For all numerical results we set $T = 4$; as remarked above we do not consider finite-volume errors here, i.e. we assume that the exact value is the expectation value of the observable in the limit $a \to 0$ at a given T. As can be seen from Eq. (28), finite-volume errors are exponentially suppressed for the topological oscillator[1]. For the double-well potential the mass is set to $m_0 = 1.0$ whereas the moment of inertia for the topological oscillator is $I_0 = 0.25$.

4.1 Autocorrelations

To quantify the significant reduction of autocorrelations which is achieved by hierarchical sampling, we measure the integrated autocorrelation time τ_{int} for the single level Metropolis-Hastings algorithm (Alg. 1) if either a simple HMC algorithm or the hierarchical delayed acceptance sampler in Alg. 3 is used. We refer to the first method as “StMC” from now on, whereas the latter is denoted as “HSMC”. In the latter case the number of levels is chosen such that the coarsest level is fixed and always has $d_0 = 16$ points for the double-well potential and $d_0 = 32$ for the topological oscillator (corresponding to lattice spacings of $a_0 = 0.25$ and $a_0 = 0.125$ respectively). A HMC sampler is used to generate proposals on the coarsest level. In all cases (i.e. either on the fine level for the StMC method or on the coarsest level for HSMC) 100 HMC steps are carried out and the size of the HMC timestep is tuned such that the acceptance probability of the HMC sampler is close to 80%. We implemented a simple HMC method based on a symplectic leapfrog integrator. The integrated autocorrelation time defined in Eq. (11) is estimated by measuring the QoI for $N = 10^5$ samples and computing

$$\widehat{\tau}_{\text{int}} = 1 + 2\sum_{s=1}^{W} \frac{\widehat{\rho}(s)}{\widehat{\rho}(0)} \approx \tau_{\text{int}} \qquad \text{with}$$

$$\begin{aligned}\widehat{\rho}(s) &= \frac{1}{N-s}\sum_{j=1}^{N-s} Q^{(j)}Q^{(j+s)} \\ &\approx \mathbb{E}[Q(\boldsymbol{x}^{(t_{\text{meas}})})Q(\boldsymbol{x}^{(t_{\text{meas}}+s)})],\end{aligned}$$

where t_{meas} is defined as in Eq. (11). As described in [36] the size of the window W is chosen such that systematic and statistical errors on $\widehat{\tau}_{\text{int}}$ are balanced. Fig. 4 shows the integrated autocorrelation time of the quantity of interest defined in Eq. (22) for the double well potential. As can be seen from this plot, τ_{int} increases in proportion to a^{-z} with $z \approx 2.29$ for small lattice spacings, whereas it is completely flat for the hierarchical sampler.

For the topological oscillator the observable is the topological susceptibility defined in Eq. (27). Here two different setups are considered for the hierarchical sampler: in the first setup the value of $I_0^{(\ell)}$ on the coarse levels is adjusted with the perturbative matching procedure described in Section 3.2.2. For comparison we also consider the case where $I_0^{(\ell)} = I_0 = 0.25$ is kept fixed on all levels, we refer to this as the “not renormalised” setup in the plots. As Fig. 5 shows, for the topological susceptibility the integrated autocorrelation time increases very rapidly with approximately $\tau_{\text{int}} \propto a^{-z}$, $z = 8.77$ for small lattice spacings if a standard HMC sampler is used. In fact, the measured $\widehat{\tau}_{\text{int}}$ is larger than 1000 for lattice spacings smaller than 0.03, and the single level method becomes

[1] To see this, note that for $T \gg I_0$ the leading order term in the sum $\hat{\Sigma}_2(T/I_0)$ defined in Eq. (29) is $2e^{-T/(2I_0)}$.

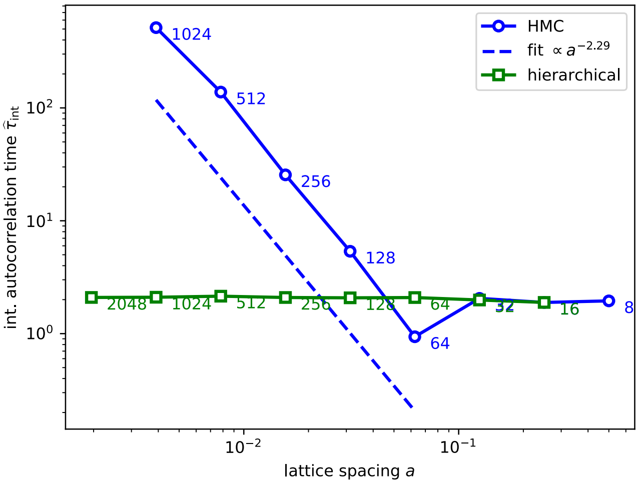

Figure 4: Integrated autocorrelation time for double well potential. Results are shown both for a standard HMC and the hierarchical sampler.

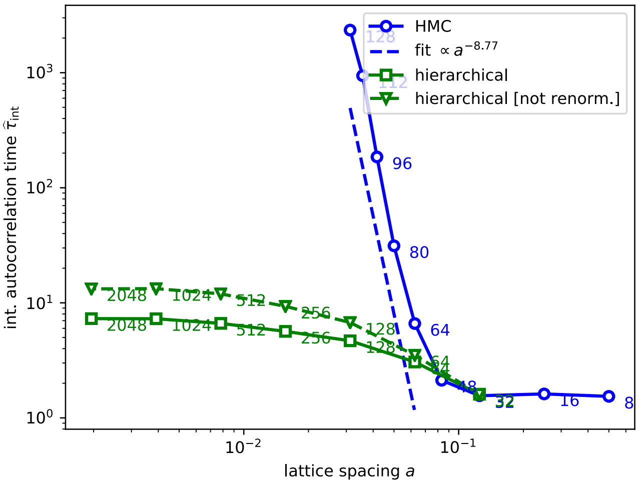

Figure 5: Integrated autocorrelation time for topological oscillator. Results are shown both for a standard HMC and the hierarchical sampler.

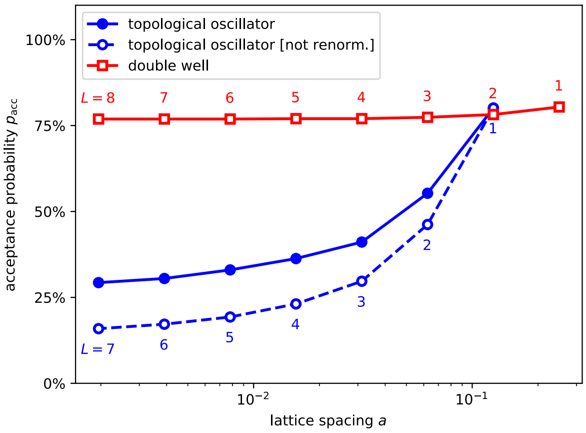

Figure 6: Acceptance probability $p_{\text{acc}} = \mathbb{P}(\boldsymbol{x}_{L-1}^{(t+1)} \neq \boldsymbol{x}_{L-1}^{(t)})$ of standard Monte Carlo with hierarchical sampling (HSMC). Results are shown both for the double well potential and the topological oscillator action.

practically unusable if a is reduced further. This is consistent with the results shown in [26, Fig. 1] and can be attributed to freezing of the integer-valued topological charge q: for small lattice spacings tunnelling between sectors with different values of q becomes increasingly unlikely. If the hierarchical sampler is used, this problem is dramatically reduced: τ_{int} is around 10 and grows only weakly for small lattice spacings. Perturbative matching reduces τ_{int} by a factor of approximately two.

The slow growth of the integrated autocorrelation for the hierarchical sampler is related to the acceptance probability of Alg. 3. Recall that a proposal on the finest level is only accepted (i.e. $\boldsymbol{x}_{L-1}^{(t+1)} \neq \boldsymbol{x}_{L-1}^{(t)}$) if all coarse level proposals have been accepted. In other words, the overall acceptance probability $p_{\text{acc}} = \mathbb{P}(\boldsymbol{x}_{L-1}^{(t+1)} \neq \boldsymbol{x}_{L-1}^{(t)})$ is the probability of accepting the proposal generated with HMC on the coarsest level (this probability is tuned to around 80%), times the probabilities of accepting the proposals generated with the two-level step in Alg. 2 on all levels $\ell = L-1, L-2, \ldots, 1$.

Fig. 6 shows this overall acceptance probability p_{acc} as the number of levels L increases for both the double-well potential and the topological oscillator. For the double-well potential the overall acceptance rate does not drop below 75% which implies that the acceptance probability of an individual two-level Metropolis-Hastings step approaches 100% on the finer levels. For the topological oscillator a similar behaviour can be observed, although the curve flattens slower and the total acceptance rate approaches a smaller value for small lattice spacings. As expected, the acceptance probability is higher for the renormalised action. This is not surprising since in the two-level Metropolis-Hastings step the coarse level proposal is a better approximation of the even modes on the next-finer level. Although this explains the smaller absolute value of the autocorrelation time in Fig. 5, measurements of the runtime (see Tab. 3 below) show that

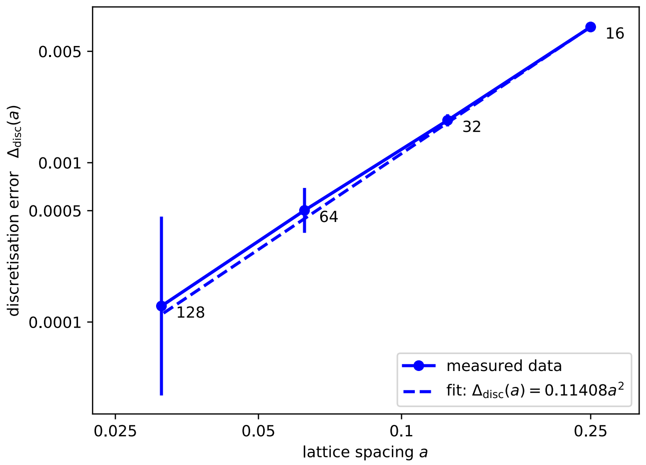

Figure 7: Discretisation error $\Delta_{\mathrm{disc}}(a)$ as a function of the lattice spacing a for the double well potential. The fit takes the form $\Delta_0 a^2$. Statistical errors are shown as vertical bars, and the data points are labelled with the number of dimensions d for each lattice spacing.

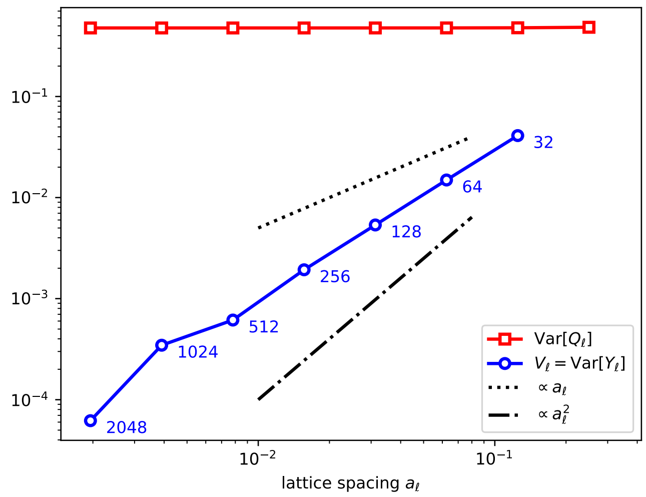

Figure 8: Variance of difference estimators Y_ℓ and the quantity of interest Q_ℓ for the double well potential. The lattice spacing on level ℓ is $a_\ell = 2^{L-1-\ell}a$. The data points are labelled with the number of dimensions d_ℓ for each lattice spacing.

using the renormalised action for the HSMC method has a smaller impact on the overall runtime since the average cost per sample grows as the acceptance probability increases. This can be seen immediately from Alg. 2: if the proposal is already rejected on one of the coarser levels, it is no longer necessary to carry out the more expensive two-level Metropolis Hastings steps on the finer levels.

4.2 Discretisation error and variance decay

To quantify the discretisation error $\Delta_{\mathrm{disc}}(a)$ as a function of the lattice spacing, we derive an asymptotic bound on $\Delta_{\mathrm{disc}}(a)$. For this assume that

$$\Delta_{\mathrm{disc}}(a) = \mathbb{E}[Q(a)] - \mathbb{E}[Q(a=0)] = \Delta_0 a^\alpha + \mathcal{O}(a^{\alpha+1}).$$

For the double-well potential the parameters Δ_0 and α are obtained by calculating $\widehat{Q}(a)$ with $N = 4 \cdot 10^8$ samples (using the hierarchical method in Alg. 3) for a range of lattice spacings $a = 1/32, 1/16, 1/8, 1/4$. As shown in Fig. 7, the measured data is consistent with $\alpha = 2$. The coefficient Δ_0 is estimated by approximating $\mathbb{E}[Q(a=0)]$ by $\widehat{Q}(a_{\mathrm{fine}})$ with $a_{\mathrm{fine}} = 1/512$ and fitting a function of the form $\log\Delta_0 + 2\log a$ to $\log(\widehat{Q}(a) - \widehat{Q}(a_{\mathrm{fine}}))$ to obtain $\Delta_0 = 0.11408$. Based on this result we use the relationship $\epsilon_{\mathrm{disc}} = \Delta_0 a^2$ to relate the lattice spacing to the tolerance on the discretisation error in the following. For the topological oscillator the asymptotic form of the discretisation error can be deduced from Eq. (28), which implies that at leading order the error is linear in the lattice spacing ($\alpha = 1$). For our choice of numerical values we find that $\Delta_0 = 0.21567$.

For the performance of the multilevel Monte Carlo method the behaviour of the variance V_ℓ of the difference of the quantity of interest between subsequent levels is important. Recall in particular that the computational complexity of the MLMC algorithm given in Eq. (18) depends on value of β which bounds $V_\ell/V_{\ell-1} \leq 2^{-\beta}$. Fig. 8 shows V_ℓ for the double well potential as well as the variance of the quantity of interest itself. As can be seen from this plot, β is larger than 1 but smaller than 2, and hence (since $\alpha = 2$, as discussed above) we expect the computational complexity of MLMC to be $\mathcal{O}(\epsilon_{\mathrm{stat}}^{-2} + \epsilon_{\mathrm{disc}}^{-1/2})$. This assumes that the sub-sampling rates and integrated autocorrelation times can be bounded, which appears plausible given the results shown in Figs. 4 and 5. The variance decay for the topological oscillator is shown in Fig. 9, both for the perturbatively renormalised action and the un-renormalised action. Renormalising the action reduces the absolute value of V_ℓ. In both cases it is safe to assume that $\beta \geq 1$, and hence we expect the computational complexity to be no worse than $\mathcal{O}(\epsilon_{\mathrm{stat}}^{-2}|\log\epsilon_{\mathrm{disc}}|^2 + \epsilon_{\mathrm{disc}}^{-1})$, provided the sub-sampling rates and integrated autocorrelation times can be bounded as $a \to 0$.

4.3 Total runtime

Finally, we compare the total runtime for three different setups:

StMC The standard single level Monte Carlo method in Alg. 1 with a HMC sampler.

HSMC The standard Monte Carlo method in Alg. 1 with the hierarchical delayed-acceptance sampler written down in Alg. 3.

MLMC The multilevel Monte Carlo method in Alg. 4

The configuration of StMC and HSMC is described in Section 4.1. For the multilevel method the coarsest level has $d_0 = 16$ points for the double-well potential and

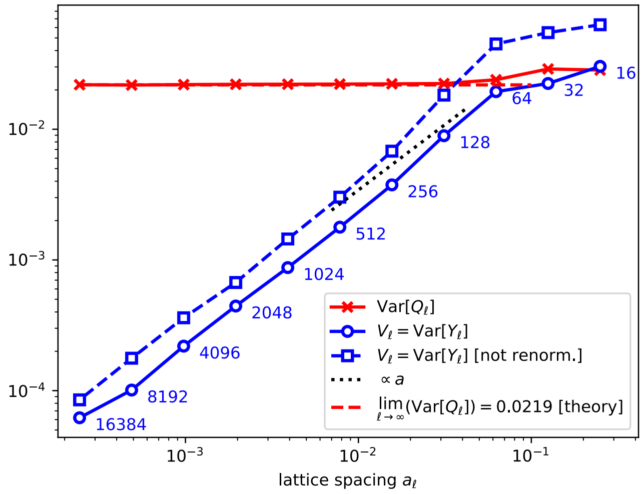

Figure 9: Variance of difference estimators Y_ℓ and the quantity of interest Q_ℓ for the topological oscillator. The lattice spacing on level ℓ is $a_\ell = 2^{L-1-\ell}a$. The continuum limit as given in Eq. (30) is shown as a red dashed line. The data points are labelled with the number of dimensions d_ℓ for each lattice spacing.

$d_0 = 32$ points for the topological oscillator. The subsampling rates t_ℓ in Alg. 4 are set to $\lceil 2\widehat{\tau}_{\text{int},\ell}\rceil$ where $\widehat{\tau}_{\text{int},\ell}$ is the estimated integrated autocorrelation time of the quantity of interest on level ℓ obtained with the hierarchical sampler. We confirmed that this choice of subsampling rate is sufficient to generate approximately independent samples and that any additional bias in the final MLMC estimator due to imperfect sub-sampling is comparable to the discretisation error. In all cases we generated and discarded a sufficiently large number of samples before computing estimators to ensure that the Markov chains are equilibrated on all levels. The runtimes reported here do not include the time spent in this burn-in phase of the simulation. The tolerance on the statistical error is set to a fixed value of $\epsilon_{\text{stat}} = 10^{-4}$ for the double well potential and $\epsilon_{\text{stat}} = 10^{-2}$ for the topological oscillator, where the difference in size accounts for the fact that the discretisation error decreases much more rapidly for the double well problem. Figs. 10 and 11 show the total runtime for those values of ϵ_{stat} and different lattice spacings a, corresponding to different values of ϵ_{disc}: as discussed in Section 4.2, for both considered problems we bound the discretisation error by $\epsilon_{\text{disc}} = \Delta_0 a^\alpha$ with $\alpha = 2$, $\Delta_0 = 0.11408$ for the double-well potential and $\alpha = 1$, $\Delta_0 = 0.21567$ for the topological oscillator. The times reported for HSMC and MLMC in Figs. 10 and 11 were obtained with the renormalised coarse level action. As can be seen from those figures, the runtime grows rapidly with $\epsilon_{\text{disc}}^{-(1+z)/\alpha}$ for the StMC method, which is proportional to a high power a^{-1-z} of the inverse lattice spacing since the discretisation error is first order in all cases. Here a factor a^{-1} arises since the cost of generating a path is $\mathcal{O}(a^{-1})$ and the remaining power a^{-z} can be explained by the growth in τ_{int} discussed in Section 4.1. As the results in Figs. 10 and 11 show, by taming au-

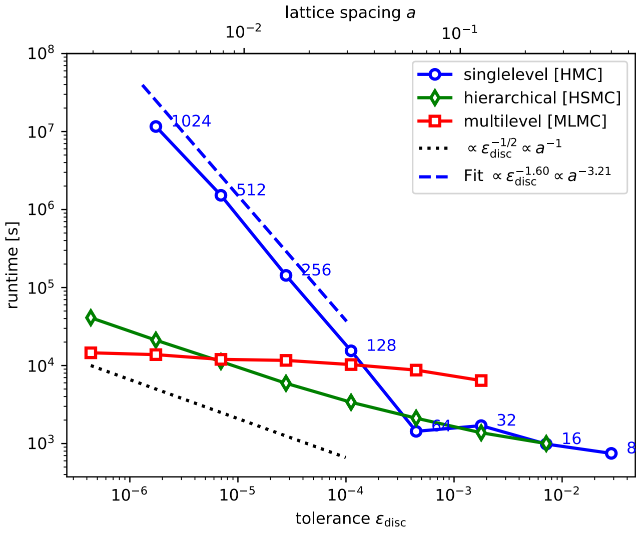

Figure 10: Runtime of different Monte Carlo sampling algorithms for the double well potential with a fixed tolerance $\epsilon_{\text{stat}} = 10^{-4}$ on the statistical error. Results are shown in seconds and as a function of the tolerance ϵ_{disc}. The data points are labelled with the number of dimensions d for each lattice spacing.

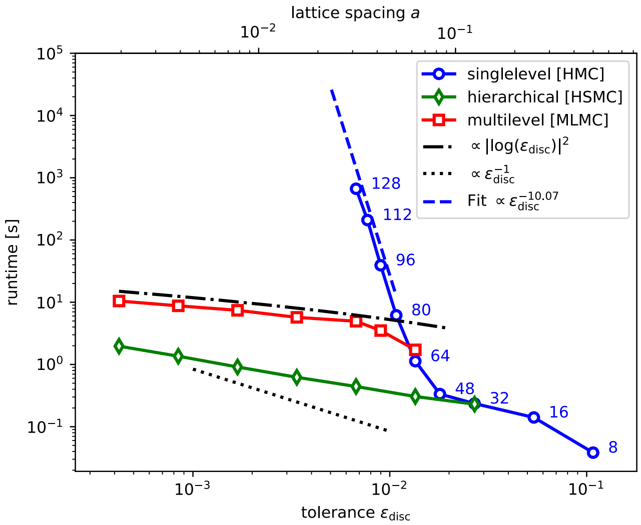

Figure 11: Runtime of different Monte Carlo sampling algorithms for the topological oscillator with a fixed tolerance $\epsilon_{\text{stat}} = 10^{-2}$ on the statistical error. Results are shown in seconds and as a function of the tolerance ϵ_{disc}. The data points are labelled with the number of dimensions d for each lattice spacing.

tocorrelations the HSMC method reduces the growth of computational cost. In fact, for the lattice spacings considered here the cost grows slower than predicted by the theoretical $\mathcal{O}(\epsilon_{\text{disc}}^{-1})$ complexity bound for the topological oscillator. This is a pre-asymptotic effect and the reason for it is twofold: firstly, the (fixed) cost of the expensive coarse level HMC sampler still contributes significantly to the overall cost of the hierarchical sampler which is not yet dominated by the evaluation of the action on the finer levels. Secondly, as can be seen from the initial drop of the total acceptance probability in Fig. 6, the probability of accepting a proposed sample in the two-level Metropolis-Hastings step on a given level is smaller than 1 on the coarser levels, before is approaches 1 on the finer levels. As a consequence, in a significant proportion of cases generating a hierarchical sample does not require the evaluation of the fine-level action since the proposal is already rejected on a coarser level.

MLMC reduces the asymptotic rate of growth further, and for the double-well potential MLMC is significantly faster than HSMC for the smallest tolerance ϵ_{disc} considered here. Tab. 1 summarises the speedup of MLMC over StMC for both problems. The relative gain of MLMC over HSMC is shown in Tab. 2. Although the gap between the runtime of the two methods also reduces for the topological oscillator, for the tolerances considered here HSMC is still faster than MLMC.

While here we kept the tolerance ϵ_{stat} fixed, in Appendix E we also show the runtime as a function of the tolerance ϵ on the total root mean square error, i.e. for $\epsilon_{\text{disc}} = \epsilon_{\text{stat}} = \epsilon/\sqrt{2}$.

4.3.1 Breakdown of MLMC cost

For the multilevel method it is instructive to break down the total computational cost into the time spent on the individual levels of the lattice hierarchy. To estimate the fraction of the runtime spent on level ℓ we computed

$$\frac{N_\ell^{\text{eff}}\mathcal{C}_\ell^{\text{eff}}}{\sum_{\ell=0}^{L-1} N_\ell^{\text{eff}}\mathcal{C}_\ell^{\text{eff}}},$$

which is plotted in Fig. 12. As can be seen from this plot, for the double well potential more than half of the time is spent on the coarsest level of the lattice hierarchy. This can be explained by the fact that, as Fig. 8 shows, the variance of difference estimators decreases by a factor between 2 and 4 between subsequent levels. The cost is more evenly distributed between levels for the topological oscillator problem since in this case the variance decays with a near-linear rate (see Fig. 9).

4.3.2 Gains from coarse level matching

Finally, we quantify the gains from coarse-level matching for the topological oscillator. For this the HSMC and MLMC runs were repeated without coarse level matching, i.e. with $I_0^{(\ell)} = I_0 = 0.25$ for all $\ell = 0, 1, \ldots, L-1$. Tab. 3 shows that this results in a relatively modest reduction of the runtime for the HSMC sampler. As already discussed at the end of Section 4.1, this can be explained by the fact that renormalising the coarse level action leads to a reduction of the integrated autocorrelation time, but this effect is largely compensated by the increased cost per sample. As the corresponding speedups for MLMC in Tab. 4 show, the gain is significantly larger for the multilevel method, where coarse level matching more than halves the runtime. This is because for MLMC matching the actions has the additional effect of reducing the absolute value of the variance of the difference estimators, as can be seen from Fig. 9.

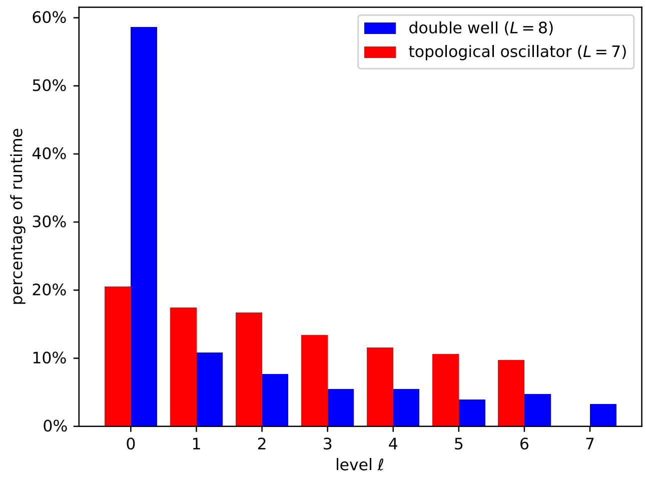

Figure 12: Estimated breakdown of cost per level for MLMC. Results are shown for the finest lattice spacing with $d = 2048$ for the double well potential and $d = 1024$ for the topological oscillator.

4.3.3 Multilevel accelerated cluster algorithm

For the topological oscillator the cluster algorithm [27] can be used to generate Monte Carlo updates with extremely small autocorrelations for arbitrarily small lattice spacing. We implemented a variant of Alg. 4 in which the new samples $\boldsymbol{x}_0^{(t+t_0)}$ (line 4) and the coarse level samples $\boldsymbol{z}_{\ell-1}^{(t+t_{\ell-1})}$ (line 7) are generated with the (single-update) cluster algorithm instead of the hierarchical sampler in Alg. 3. Again the sub-sampling rates are set to $t_\ell = \lceil 2\widehat{\tau}_{\text{int},\ell} \rceil$ and we find numerically that $\widehat{\tau}_{\text{int},\ell} \approx 3$ for all levels ℓ considered here. The number of unknowns on the coarsest level was fixed to $d_0 = 512$, while increasing to fine-level problem size from $d = 1024$ to $d = 16384$. The performance of this MLMC algorithm is compared to the standard, single-level cluster algorithm, again updating a single cluster in each Metropolis Hastings step. As the numerical results in Fig. 13 show, MLMC is around 40% faster than the standalone cluster algorithm for the smallest lattice spacing considered here. The speedup of the MLMC accelerated cluster algorithm over the single-level cluster algorithm for all lattice spacings is shown in Tab. 5. More importantly, the numerical experiments show that the runtime of MLMC increases roughly as $\mathcal{O}(|\log(\epsilon_{\text{disc}})|^2)$ and thereby grows significantly slower than the runtime of the cluster algorithm, which shows the expected $\mathcal{O}(\epsilon_{\text{disc}}^{-1})$ growth. Again we also show the

	d	a	ϵ_{disc}	t_{StMC}	t_{MLMC}	speedup
doublewell	32	0.1250	$1.78 \cdot 10^{-3}$	0.17	0.64	0.3×
	64	0.0625	$4.46 \cdot 10^{-4}$	0.14	0.87	0.2×
	128	0.0312	$1.11 \cdot 10^{-4}$	1.54	1.03	1.5×
	256	0.0156	$2.79 \cdot 10^{-5}$	14.32	1.16	12.3×
	512	0.0078	$6.96 \cdot 10^{-6}$	152.21	1.20	126.9×
	1024	0.0039	$1.74 \cdot 10^{-6}$	1160.02	1.38	840.9×
topological oscillator	64	0.0625	$1.35 \cdot 10^{-2}$	1.13	1.72	0.7×
	96	0.0417	$8.99 \cdot 10^{-3}$	39.26	3.51	11.2×
	128	0.0312	$6.74 \cdot 10^{-3}$	665.32	4.95	134.4×

Table 1: Comparison of runtime for standard, single-level Monte Carlo (StMC) and multilevel Monte Carlo (MLMC). All times for the double well potential were obtained with $\epsilon_{\text{stat}} = 10^{-4}$ and are given in units of 10^4 seconds. For the topological oscillator a value of $\epsilon_{\text{stat}} = 10^{-2}$ was used and times are given in seconds.

	d	a	ϵ_{disc}	t_{HSMC}	t_{MLMC}	speedup
doublewell	32	0.1250	$1.78 \cdot 10^{-3}$	0.14	0.64	0.2×
	64	0.0625	$4.46 \cdot 10^{-4}$	0.21	0.87	0.2×
	128	0.0312	$1.11 \cdot 10^{-4}$	0.34	1.03	0.3×
	256	0.0156	$2.79 \cdot 10^{-5}$	0.59	1.16	0.5×
	512	0.0078	$6.96 \cdot 10^{-6}$	1.13	1.20	0.9×
	1024	0.0039	$1.74 \cdot 10^{-6}$	2.12	1.38	1.5×
	2048	0.0020	$4.35 \cdot 10^{-7}$	4.10	1.46	2.8×
topological oscillator	64	0.0625	$1.35 \cdot 10^{-2}$	0.31	1.72	0.2×
	128	0.0312	$6.74 \cdot 10^{-3}$	0.44	4.95	0.1×
	256	0.0156	$3.37 \cdot 10^{-3}$	0.62	5.72	0.1×
	512	0.0078	$1.68 \cdot 10^{-3}$	0.91	7.42	0.1×
	1024	0.0039	$8.42 \cdot 10^{-4}$	1.36	8.76	0.2×
	2048	0.0020	$4.21 \cdot 10^{-4}$	1.97	10.47	0.2×

Table 2: Comparison of runtime for Monte Carlo with hierarchical sampling (HSMC) and multilevel Monte Carlo (MLMC). All times for the double well potential were obtained with $\epsilon_{\text{stat}} = 10^{-4}$ and are given in units of 10^4 seconds. For the topological oscillator a value of $\epsilon_{\text{stat}} = 10^{-2}$ was used and times are given in seconds.

d	a	ϵ_{disc}	$t^{(0)}_{\text{HSMC}}$	t_{HSMC}	speedup
64	0.0625	$1.35 \cdot 10^{-2}$	0.36	0.31	1.2×
128	0.0312	$6.74 \cdot 10^{-3}$	0.64	0.44	1.4×
256	0.0156	$3.37 \cdot 10^{-3}$	0.96	0.62	1.5×
512	0.0078	$1.68 \cdot 10^{-3}$	1.44	0.91	1.6×
1024	0.0039	$8.42 \cdot 10^{-4}$	1.90	1.36	1.4×
2048	0.0020	$4.21 \cdot 10^{-4}$	2.57	1.97	1.3×

Table 3: Comparison of Monte Carlo with hierarchical sampling (HSMC) without and with coarse level mass matching, denoted by $t^{(0)}_{\text{HSMC}}$ and t_{HSMC} respectively. All times were obtained with $\epsilon_{\text{stat}} = 10^{-2}$ and are given in seconds.

d	a	ϵ_{disc}	$t^{(0)}_{\text{MLMC}}$	t_{MLMC}	speedup
64	0.0625	$1.35 \cdot 10^{-2}$	4.07	1.72	2.4×
128	0.0312	$6.74 \cdot 10^{-3}$	10.68	4.95	2.2×
256	0.0156	$3.37 \cdot 10^{-3}$	16.47	5.72	2.9×
512	0.0078	$1.68 \cdot 10^{-3}$	19.27	7.42	2.6×
1024	0.0039	$8.42 \cdot 10^{-4}$	21.92	8.76	2.5×
2048	0.0020	$4.21 \cdot 10^{-4}$	25.64	10.47	2.5×

Table 4: Comparison of Multilevel Monte Carlo (MLMC) runtime without and with coarse level mass matching, denoted by $t^{(0)}_{\text{MLMC}}$ and t_{MLMC} respectively. All times were obtained with $\epsilon_{\text{stat}} = 10^{-2}$ and are given in seconds.

corresponding results for varying $\epsilon_{\text{stat}} = \epsilon_{\text{disc}} = \epsilon/\sqrt{2}$ in Fig. 17 in Appendix E.

It should be stressed at this point, that while the cluster algorithm proved to be highly efficient for the topological oscillator, its applicability is highly problem dependent and can for example not be directly used for the double well potential problem considered in this work or many other problems in quantum field theory.

5 Conclusion

In this paper we have described a hierarchical sampling algorithm and applied it for simulations in quantum mechanics. We demonstrated that this can overcome the rapid growth of autocorrelations as the continuum limit is approached. In particular, we considered the anharmonic oscillator with a non-symmetric double well potential and the quantum mechanical topological oscillator model described in [26]. Empirically we find that for both cases the integrated autocorrelation time does not show any signifcant increase towards the continuum limit when the lattice spacing approaches zero. This result is particularly significant for the susceptibility of a topological oscillator, which suffers from freezing of the topological charge if a single level method with a standard HMC sampler is used.

d	a	ϵ_{disc}	t_{StMC}	t_{MLMC}	speedup
1024	0.003906	$8.42 \cdot 10^{-4}$	2.44	13.13	0.2×
2048	0.001953	$4.21 \cdot 10^{-4}$	4.86	18.50	0.3×
4096	0.000977	$2.11 \cdot 10^{-4}$	9.38	19.06	0.5×
8192	0.000488	$1.05 \cdot 10^{-4}$	20.60	23.81	0.9×
16384	0.000244	$5.27 \cdot 10^{-5}$	39.51	27.38	1.4×

Table 5: Comparison of singlelevel (StMC) and multilevel (MLMC) cluster algorithm for the topological oscillator. All times were obtained with $\epsilon_{\text{stat}} = 10^{-3}$ and are given in seconds.

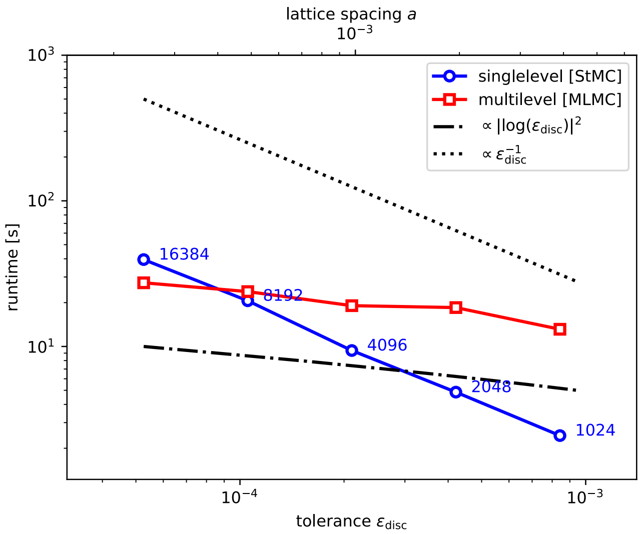

Figure 13: Runtime of singlelevel (StMC) and multilevel (MLMC) cluster algorithm for the topological oscillator with a tolerance $\epsilon_{\text{stat}} = 10^{-3}$ on the statistical error. Results are shown in seconds and as a function of the tolerance ϵ_{disc}. The data points are labelled with the number of dimensions d for each lattice spacing.

Combining this new hierarchical sampling technique with a multilevel Monte Carlo acceleration results in a dramatic reduction of the computational complexity and a significant reduction of the overall runtime. For the finest considered lattice spacings the additional speedup from MLMC (compared to hierarchical sampling for a non-symmetric double well potential or the cluster algorithm for a topological oscillator) is around 1.4× to 2.8×. We find that the accurate construction of coarse level theories with an approximate matching procedure is important to achieve optimal performance.

In this paper we have concentrated on reducing the time spent in the sampling phase of the Markov Chain Monte Carlo simulation and did not include burn-in times in the reported runtimes. However, also burn-in can be accelerated with hierarchical sampling since the reduction in autocorrelation time allows chains to equilibrate much faster.

While here we have demonstrated the methods for quantum mechanical systems, the same techniques can be used in lattice field theory simulations. In fact, as explained in the introduction, we expect the speedup to be more significant in this case since the relative cost of computations on coarser levels is further suppressed. A crucial step will be to construct suitable coarse grained theories which could be achieved analytically in perturbation theory or by adopting the framework of Symanzik's effective theory, where the improvement coefficients can be computed perturbatively. Since many physically interesting theories, and in particular lattice QCD, is asymptotically free, this is expected to work increasingly well as the continuum limit is approached. Of, course, finding non-perturbative methods to construct the course grained theory would be even better.

Recently, multilevel Monte Carlo has received significant attention in other areas, which led to further innovations. While here the method is described in the most natural setup, where coarse levels are constructed by increasing the lattice spacing, coarsening in other categories is also possible and potentially leads to further performance gainsby using the multiindex Monte Carlo [37] technique. For example the complexity of the theory could be reduced on coarser levels or the physical volume of the lattices could be increased with ℓ, thus aiming to approach the continuum limit $a \to 0$ and large volume limit $T \to \infty$ simultaneously. In lattice QCD one might increase the dynamical quark masses on the coarser levels, which simplifies the computation of the fermion determinant.

In summary, the success of the benchmark computations presented in this paper suggests that applying MLMC techniques to higher dimensional theories, e.g. to gauge theories, is indeed a promising approach which we plan to follow in the future.

Acknowledgements

This research made use of the Balena High Performance Computing (HPC) Service at the University of Bath. We would like to thank Stefan Schäfer (DESY Zeuthen) for useful discussions and comments on this paper. We are grateful to Christopher Anders and Pan Kessel (BIFOLD & TU Berlin) for pointing out a bug in our code which resulted in incorrect results in the original version of this paper.

A Extension to higher dimensions

To illustrate how the methods in this paper, and in particular the two-level Metropolis-Hastings step in Alg. 2, can be extended to higher dimensions, consider a discretised two-dimensional theory for which the degrees of freedom are located at the vertices of a uniform lattice with spacing a. If Ω is the state space and $S : \Omega \to \mathbb{R}$ is the lattice action, the probability density π^* which is sampled with a Monte Carlo method is defined by

$$\pi^*(\mathbf{\Phi}) = \mathcal{Z}^{-1}e^{-S(\mathbf{\Phi})} \qquad \text{for all } \mathbf{\Phi} \in \Omega.$$

Starting from the finest lattice $\mathcal{T}$, construct a hierarchy of L lattices $\mathcal{T} = \mathcal{T}_{L-1}, \mathcal{T}_{L-2}, \dots, \mathcal{T}_0$ by doubling the lattice spacing simultaneously in both dimensions; this is shown for two subsequent levels $\mathcal{T}_\ell$, $\mathcal{T}_{\ell-1}$ of the hierarchy in Fig. 14 (left). On level ℓ the lattice spacing is written as $a_\ell = 2^{\ell-L+1}a$ and we assume that there is an action $S_\ell : \Omega_\ell \to \mathbb{R}$ with associated probability density π_ℓ where

$$\pi_\ell(\mathbf{\Phi}) = \mathcal{Z}_\ell^{-1}e^{-S_\ell(\mathbf{\Phi})} \qquad \text{for all } \mathbf{\Phi} \in \Omega_\ell. \tag{32}$$

Again, the coarse level theories are naturally obtained as (approximate) effective theories of the original fine-level theory on level $L-1$ where $S_{L-1} = S$ and $\pi_{L-1} = \pi^*$.

While the coarse level actions are constructed by starting from the original lattice, the hierarchical sampler in Alg. 2 constructs new fine level samples by generating a proposal on the coarsest lattice and successively adding fine-level modes. On each level ℓ this requires a mechanism for filling in the values of unknowns in the fine-level space Ω_ℓ for a given state $\mathbf{\Phi}' \in \Omega_{\ell-1}$ in the coarse level space $\Omega_{\ell-1}$. We use two iterations of the construction described for the Ising model in [31] to achieve this. As illustrated in Fig. 1(a) there, the key idea is to use a rotated lattice with a lattice spacing that is reduced by a factor $\sqrt{2}$. The values at the additional sites that are generated in each rotation are drawn from a distribution which depends only on the values at the already existing sites[2].

To explain this process in more detail, observe that on each level ℓ the state space Ω_ℓ can be written as the direct sum of three spaces $\Omega_\ell^{(2)}$, $\Omega_\ell^{(1)}$ and $\Omega_{\ell-1}$ with $\Omega_\ell := \Omega_\ell^{(2)} \oplus \Omega_\ell^{(1)} \oplus \Omega_{\ell-1}$, which should be compared to the decomposition in Eq. (12). To see this and to define $\Omega_\ell^{(1)}$, $\Omega_\ell^{(2)}$, separate the unknowns $\mathbf{\Phi} \in \Omega_\ell$ into three different classes, depending on which topological entity of a coarse grid cell on level $\ell - 1$ they are associated with, namely

1. coarse-level unknowns associated with coarse level vertices, collected in a vector $\mathbf{\Phi}' \in \Omega_{\ell-1}$ and shown as empty black circles in Fig. 14,

2. fine-level unknowns associated with the interior of coarse level cells, collected in $\tilde{\mathbf{\Phi}}^{(1)} \in \Omega_\ell^{(1)}$ and shown as empty blue squares and

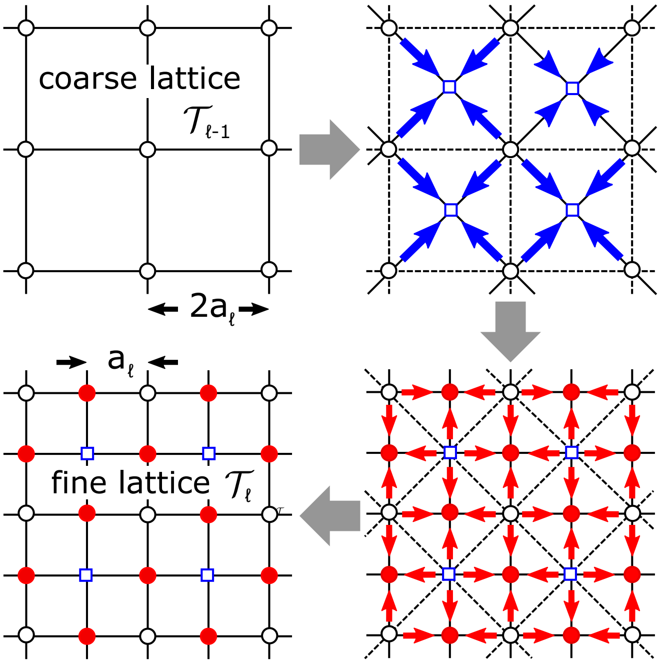
coarse lattice
$\mathcal{T}_{\ell\text{-}1}$
2a$_\ell$
a$_\ell$
fine lattice $\mathcal{T}_\ell$

Figure 14: Fill-in of fine-level unknowns on a hierarchical two-dimensional lattice as required in lines 5 and 6 of Alg. 5. Starting from the coarse level unknowns on a rotated lattice (upper left), first the unknowns in $\Omega_\ell^{(1)}$ associated with the empty blue squares are filled in using $\tilde{\pi}_\ell^{(1)}$ (upper right). Next, the distribution $\tilde{\pi}_\ell^{(2)}$ is used to fill in the unknowns in $\Omega_\ell^{(2)}$ associated with the solid red circles (lower right) to finally obtain the state on the fine lattice (lower left).

3. fine-level unknowns associated with edges of coarse level cells, collected in $\tilde{\mathbf{\Phi}}^{(2)} \in \Omega_\ell^{(2)}$ and shown as solid red circles in the figure.

Given $\mathbf{\Phi}' \in \Omega_{\ell-1}$, $\tilde{\mathbf{\Phi}}^{(1)} \in \Omega_\ell^{(1)}$ and $\tilde{\mathbf{\Phi}}^{(2)} \in \Omega_\ell^{(2)}$ write $\overline{\mathbf{\Phi}} = [\tilde{\mathbf{\Phi}}^{(1)}, \mathbf{\Phi}'] \in \overline{\Omega}_\ell := \Omega_\ell^{(1)} \oplus \Omega_{\ell-1}$ and

$$\mathbf{\Phi} = [\tilde{\mathbf{\Phi}}^{(2)}, \overline{\mathbf{\Phi}}] = [\tilde{\mathbf{\Phi}}^{(2)}, [\tilde{\mathbf{\Phi}}^{(1)}, \mathbf{\Phi}']] \in \Omega_\ell, \tag{33}$$

which should be compared to Eq. (13) in the main text. Assume that there is a conditional probability density $\tilde{\pi}^{(1)}(\cdot|\mathbf{\Phi}')$ on the state space $\Omega_\ell^{(1)}$, given the values of the coarse level unknowns $\mathbf{\Phi}' \in \Omega_{\ell-1}$. Since the empty black circles and empty blue squares in the top right of Fig. 14 define a rotated lattice with spacing $\sqrt{2}a_\ell$, the density $\tilde{\pi}^{(1)}$ can be constructed by writing down an action $\tilde{S}^{(1)} : \overline{\Omega}_\ell \to \mathbb{R}$ on this rotated lattice, namely

$$\tilde{\pi}^{(1)}(\tilde{\mathbf{\Phi}}^{(1)}|\mathbf{\Phi}') = \left(\tilde{\mathcal{Z}}_\ell^{(1)}(\mathbf{\Phi}')\right)^{-1} \exp\left[-\tilde{S}_\ell^{(1)}([\tilde{\mathbf{\Phi}}^{(1)}, \mathbf{\Phi}'])\right]. \tag{34}$$

This action could for example be obtained by a renormalisation group transformation on S_ℓ, followed by some approximations that guarantee that it is possible to effectively generate states in $\Omega_\ell^{(1)}$ for a given $\mathbf{\Phi}'$. Similarly define a conditional probability density $\tilde{\pi}^{(2)}(\cdot|\overline{\mathbf{\Phi}})$ on $\Omega_\ell^{(2)}$, given the values of the unknowns $\overline{\mathbf{\Phi}} \in \overline{\Omega}_\ell$. Here

[2] Although sampling is particularly simple if the action only contains nearest-neighbour interactions (as for the Ising model in [31]) so that the value at each new site can be drawn independently, this is not a necessary condition.

$S^{(2)} : \Omega_\ell \to \mathbb{R}$ can be expressed as an approximation of S_ℓ with

$$\begin{aligned}\tilde{\pi}^{(2)}(\tilde{\mathbf{\Phi}}^{(2)}|\overline{\mathbf{\Phi}}) &= \left(\tilde{\mathcal{Z}}_\ell^{(2)}(\overline{\mathbf{\Phi}})\right)^{-1} \exp\left[-\tilde{S}_\ell^{(2)}([\tilde{\mathbf{\Phi}}^{(2)}, \overline{\mathbf{\Phi}}])\right] \\ &= \left(\tilde{\mathcal{Z}}_\ell^{(2)}([\tilde{\mathbf{\Phi}}^{(1)}, \mathbf{\Phi}'])\right)^{-1} \\ &\quad \times\ \exp\left[-\tilde{S}_\ell^{(2)}([\tilde{\mathbf{\Phi}}^{(2)}, [\tilde{\mathbf{\Phi}}^{(1)}, \mathbf{\Phi}']])\right].\end{aligned} \tag{35}$$

The exact choice of $\tilde{S}^{(1)}$ and $\tilde{S}^{(2)}$ influences the acceptance rate, but does not have any impact on the fine-level discretisation error. Alg. 3 for the Metropolis-Hastings step $\mathbf{\Phi}_\ell^{(t)} \to \mathbf{\Phi}_\ell^{(t)}$ can now be rewritten for a two-dimensional theory as shown in the following Alg. 5.

Algorithm 5 Two-level Metropolis Hastings step for two-dimensional theories.
Input: level ℓ, current sample $\mathbf{\Phi}_\ell^{(t)} \sim \pi_\ell$, proposal distribution $q_{\ell-1}$
Output: new sample $\mathbf{\Phi}_\ell^{(t+1)} \sim \pi_\ell$

```
Let $\mathbf{\Phi}_\ell^{(t)} = [\tilde{\mathbf{\Phi}}_\ell^{(2,t)}, [\tilde{\mathbf{\Phi}}_\ell^{(1,t)}, \mathbf{\Phi}_{\ell-1}^{(t)}]]$ with $\mathbf{\Phi}_{\ell-1}^{(t)} \in \Omega_{\ell-1}$, $\mathbf{\Phi}_\ell^{(1,t)} \in \Omega_\ell^{(1)}$, $\mathbf{\Phi}_\ell^{(2,t)} \in \Omega_\ell^{(2)}$ as in Eq. (33) and pick $\mathbf{\Psi}_{\ell-1}$ from $q_{\ell-1}(\cdot|\mathbf{\Phi}_{\ell-1}^{(t)})$.
if $\mathbf{\Phi}_{\ell-1}^{(t+1)} = \mathbf{\Phi}_{\ell-1}^{(t)}$ (the coarse level proposal was rejected) then
    Set $\mathbf{\Phi}_\ell^{(t+1)} \mapsfrom \mathbf{\Phi}_\ell^{(t)}$
else
    Pick $\tilde{\mathbf{\Psi}}_\ell^{(1)}$ from $\tilde{\pi}_\ell^{(1)}(\cdot|\mathbf{\Psi}_{\ell-1})$
    Pick $\tilde{\mathbf{\Psi}}_\ell^{(2)}$ from $\tilde{\pi}_\ell^{(2)}(\cdot|[\tilde{\mathbf{\Psi}}^{(1)}, \mathbf{\Psi}_{\ell-1}])$
    Let $\mathbf{\Psi}_\ell = [\tilde{\mathbf{\Psi}}_\ell^{(2)}, [\tilde{\mathbf{\Psi}}_\ell^{(1)}, \mathbf{\Psi}_{\ell-1}]]$ and compute

        $\exp[-\Delta S_\ell] = \rho_\ell \cdot \rho_\ell^{(2)} \cdot \rho_\ell^{(1)} \cdot \rho'_{\ell-1}$   with
        $\rho_\ell := \frac{\pi_\ell(\mathbf{\Psi}_\ell)}{\pi_\ell(\mathbf{\Phi}_\ell^{(t)})}$
        $\rho_\ell^{(2)} := \frac{\tilde{\pi}_\ell^{(2)}(\tilde{\mathbf{\Phi}}_\ell^{(2,t)}|[\tilde{\mathbf{\Phi}}_\ell^{(1,t)}, \mathbf{\Phi}_{\ell-1}^{(t)}])}{\tilde{\pi}_\ell^{(2)}(\tilde{\mathbf{\Psi}}_\ell^{(2)}|[\tilde{\mathbf{\Psi}}_\ell^{(1)}, \mathbf{\Psi}_{\ell-1})]}$
        $\rho_\ell^{(1)} := \frac{\tilde{\pi}_\ell^{(1)}(\tilde{\mathbf{\Phi}}_\ell^{(1,t)}|\mathbf{\Phi}_{\ell-1}^{(t)})}{\tilde{\pi}_\ell^{(1)}(\tilde{\mathbf{\Psi}}_\ell^{(1)}|\mathbf{\Psi}_{\ell-1})}$
        $\rho'_{\ell-1} := \frac{\pi_{\ell-1}(\mathbf{\Phi}_{\ell-1}^{(t)})}{\pi_{\ell-1}(\mathbf{\Psi}_{\ell-1})}$

    Accept the proposal $\mathbf{\Psi}_\ell$ and set $\mathbf{\Phi}_\ell^{(t+1)} \mapsfrom \mathbf{\Psi}_\ell$ with probability $\min\{1, \exp[-\Delta S_\ell]\}$; set $\mathbf{\Phi}_\ell^{(t+1)} \mapsfrom \mathbf{\Phi}_\ell^{(t)}$ if the proposal is rejected.
end if
```

An explicit expression for ΔS_ℓ is readily obtained from Eqs. (32), (34) and (35). The key difference between Alg. 2 and Alg. 5 is that the fine level states from $\Omega_\ell^{(1)}$ and $\Omega_\ell^{(2)}$ are filled in in two steps and that the triple product of ratios in Alg. 2 has been replaced by the product of the four ratios ρ_ℓ, $\rho_\ell^{(2)}$, $\rho_\ell^{(1)}$ and $\rho'_{\ell-1}$ in Alg. 5. A similar construction is possible in higher dimensions where it is necessary to successively fill in the fine level unknowns which are not in the coarse level state space.

B Multilevel Monte Carlo cost analysis

We make the reasonable assumption that the cost $\mathcal{C}_{\text{coarse}}$ of generating a sample $\boldsymbol{x}_0^{(t+1)}$ with the standard Metropolis sampler on the coarsest level is proportional to the number of unknowns d_0, and does not increase as the number of levels increases (while keeping a_0 fixed). More specifically, we assume that this cost $\mathcal{C}_{\text{coarse}}$ can be bounded by

$$\mathcal{C}_{\text{coarse}} \le A_0 d_0 = 2^{-L+1} A_0 d$$

for some constant A_0. Furthermore, given the coarse level sample $\boldsymbol{y}_{\ell-1}$, the cost of executing Alg. 2 is proportional to d_ℓ and can be bounded by

$$\mathcal{C}_\ell^{2-\text{level}} \le B_0 d_\ell = 2^{\ell-L+1} B_0 d$$

for some other constant B_0. A straightforward calculation shows that the cost of obtaining a new sample $\boldsymbol{x}_\ell^{(t+1)}$ with Alg. 3 can be bounded by

$$\mathcal{C}_\ell \le (A_0 + B_0) d_\ell = 2^{\ell-L+1}(A_0 + B_0) d,$$

i.e. does not grow more than linearly with the number d_ℓ of unknowns on level ℓ. Taking into account the sub-sampling rates t_ℓ, the cost of obtaining an independent measurement of $Y_\ell^{(j)}$ on level ℓ in Alg. 4 is therefore

$$\mathcal{C}_\ell^{\text{eff}} = \begin{cases} \lceil \tau_{\text{int},\ell} \rceil \left(\mathcal{C}_\ell^{2-\text{level}} + t_{\ell-1}\mathcal{C}_{\ell-1}\right) & \text{for } \ell = 1, \dots, L-1 \\ \lceil \tau_{\text{int},0} \rceil t_0 \mathcal{C}_{\text{coarse}} & \text{for } \ell = 0 \end{cases} \tag{36}$$

where $\tau_{\text{int},\ell}$ is the integrated autocorrelation time on level ℓ. In our code we measured $\mathcal{C}_{\text{coarse}}$, $\mathcal{C}_\ell^{2-\text{level}}$ and $\mathcal{C}_\ell$ during the setup phase of each run, and then used Eq. (36) to compute $\mathcal{C}_\ell^{\text{eff}}$ required in Eq. (16). The integrated autocorrelation time were updated on-the-fly in the multilevel Monte Carlo algorithm, generating additional samples if this increased the N_ℓ^{eff} in Eq. (16).

To quantify the cost of the multilevel Monte Carlo algorithm in Alg. 4, further assume that the sub-sampling rates t_ℓ are bounded by some $t_{\max} \ge t_\ell$ for all $\ell = 1, \dots, L-2$. By definition, this is also an upper bound on the integrated autocorrelation times on all levels, i.e. $\tau_{\text{int},\ell} \le t_{\max}$ for $\ell = 0, 1, \dots, L-1$. Then, there exists a constant $\tilde{C}_0$ such that $\mathcal{C}_\ell^{\text{eff}} \le \tilde{C}_0 d_\ell = 2^\ell \tilde{C}_0 d_0$, in other words the cost for generating an independent measurement $Y_\ell^{(j)}$ on level ℓ does not grow at a faster rate than the number of unknowns $d_\ell = 2^\ell d_0 = 2^{\ell-L+1} d$ on this particular level. More generally, to make the following derivation applicable for field theories in $D > 1$ dimensions (where $D = 1$ corresponds to quantum mechanics), we assume that there is a $C_0 > 0$ such that

$$\mathcal{C}_\ell^{\text{eff}} \le 2^{D\ell} C_0 \quad \text{for all } \ell = 0, 1, \dots, L-1.$$

We now show how the cost of the MLMC algorithm depends on the tolerances ϵ_{disc} and ϵ_{stat} as $\epsilon_{\text{disc}}, \epsilon_{\text{stat}} \to 0$. Using the definition of N_ℓ^{eff} in Eq. (16) and the fact that

$\max\{A,B\} \le A+B$, the total cost of MLMC with a tolerance ϵ_{stat} on the statistical error and a given number of levels L can be bounded by

$$\mathcal{C}_{\text{MLMC}} = \sum_{\ell=0}^{L-1} N_\ell^{\text{eff}} \mathcal{C}_\ell^{\text{eff}} \le \epsilon_{\text{stat}}^{-2}\sigma(L)^2 + \tilde{\sigma}(L) \tag{37}$$

where

$$\sigma(L) := \sum_{\ell=0}^{L-1}\sqrt{V_\ell \mathcal{C}_\ell^{\text{eff}}}, \qquad \tilde{\sigma}(L) := \sum_{\ell=0}^{L-1}\mathcal{C}_\ell^{\text{eff}}.$$

Assuming that

$$V_\ell \le 2^{-\beta\ell}V_0,$$

a straightforward calculation shows that $\sigma(L)$ can be bounded as follows, depending on whether β is larger, equal or smaller than D:

$$\sigma(L) \le \kappa_0\sum_{\ell=0}^{L-1}2^{\frac{D-\beta}{2}\ell} \le \begin{cases} \kappa_+ & \text{for } \beta > D \\ \kappa_0 L & \text{for } \beta = D \\ \kappa_- 2^{\frac{1-\beta}{2}L} & \text{for } \beta < D. \end{cases} \tag{38}$$

with the constants

$$\kappa_0 = \sqrt{C_0V_0}, \qquad \kappa_+ = \frac{\kappa_0}{1-2^{\frac{D-\beta}{2}}} = -\kappa_-.$$

The sum $\tilde{\sigma}(L)$ is readily bounded by

$$\tilde{\sigma}(L) \le C_0\frac{2^{DL}}{2^D-1} \tag{39}$$

To obtain a bound on the number of levels L, we further assume that the discretisation is of order α, i.e. for a given lattice spacing a the discretisation error $\Delta_{\text{disc}}(a)$ can be bounded by

$$\Delta_{\text{disc}}(a) \le \tilde{\Delta}_0 a^\alpha$$

for some constants $\alpha \ge 1$, $\tilde{\Delta}_0$. If we set $a = 2^{-L_{\max}+1}a_0$ with

$$L_{\max} = 1 + \left\lceil \log_2\left(a_0\tilde{\Delta}_0^{1/\alpha}\right) - \frac{1}{\alpha}\log_2\epsilon_{\text{disc}}\right\rceil$$

the discretisation error will be smaller than ϵ_{disc}. Hence, it is not necessary to use more than $L_{\max}$ levels, and L in Eq. (38) can be bounded by

$$L \le 2 + \log_2\left(a_0\tilde{\Delta}_0^{1/\alpha}\right) - \frac{1}{\alpha}\log_2\epsilon_{\text{disc}}.$$

Using this bound in Eqs. (38) and (39) implies that the cost in Eq. (37) has the following computational complexity as a function of ϵ_{disc} and ϵ_{stat}:

$$\mathcal{C}_{\text{MLMC}} = \begin{cases} \mathcal{O}\left(\epsilon_{\text{stat}}^{-2} + \epsilon_{\text{disc}}^{-D/\alpha}\right) & \text{for } \beta > D \\ \mathcal{O}\left(\epsilon_{\text{stat}}^{-2}|\log\epsilon_{\text{disc}}|^2 + \epsilon_{\text{disc}}^{-D/\alpha}\right) & \text{for } \beta = D \\ \mathcal{O}\left(\epsilon_{\text{stat}}^{-2}\epsilon_{\text{disc}}^{-\frac{D-\beta}{\alpha}} + \epsilon_{\text{disc}}^{-D/\alpha}\right) & \text{for } \beta < D. \end{cases} \tag{40}$$

For the quantum mechanical problems considered in this paper we have that $D=1$, which leads to the computational complexity in Eq. (18); Eq. (4) in the introduction is a special case of this for $\alpha=\beta=1$. In fact, as explained in [17], $\alpha=\beta$ holds more generally for the Markov Chain variant of the multilevel Monte Carlo algorithm. Hence, for quantum field theories in higher dimensions with $D > \alpha = \beta$ the third case in Eq. (40) applies, which results in Eq. (5) in the introduction. Finally, setting $\epsilon_{\text{stat}} = \epsilon_{\text{disc}} = \epsilon/\sqrt{2}$ gives Eq. (6).

C Memory requirements

To put the memory requirements of the algorithms described in this paper into context consider a D dimensional quantum field theory and HMC sampling as an established reference method. In addition to the current state $\boldsymbol{x}^{(t)}$, both the proposal $\boldsymbol{y}$ and $\nu \ge 1$ temporary vectors have to be stored to implement the symplectic timestepping scheme in the enlarged phase space. For the simple leapfrog implementation used in this work $\nu = 1$ since only one additional momentum vector is required. If there are d lattice points in each direction this leads to a total storage requirement of $2+\nu$ state vectors of length d^D or $\mathcal{M}_{\text{HMC}} = (2+\nu)d^D$ double precision variables in D dimensions. Executing the two-level Metropolis Hastings step in Alg. 2 on level ℓ of the hierarchy requires storage for $\boldsymbol{x}_\ell^{(t)}$ and the proposal $\boldsymbol{y}_\ell$, which are both vectors of length d_ℓ^D. Depending on how the proposal on level $\ell-1$ is generated, this might require additional vectors of length $d_{\ell-1}^D$. For example, if the proposals $\boldsymbol{y}_{\ell-1}$ are drawn from $q_{\ell-1}(\cdot|\boldsymbol{x}_{\ell-1}^{(t)})$ with a single level Metropolis Hasting method and a HMC proposal distribution, one would require ν additional vectors. However, since unknowns on the finer levels are filled in recursively and existing entries of $\boldsymbol{x}_\ell^{(t)}$ are used to represent the current state on coarser levels, the hierarchical sampler in Alg. 3 only needs to store two vectors of length d_ℓ^D to represent $\boldsymbol{x}_\ell^{(t)}$ and the proposal $\boldsymbol{y}_\ell$ as well as ν vectors of length d_0^D to account for the Metropolis Hastings step on the coarsest level with $\ell = 0$. This leads to total storage requirements of $\mathcal{M}_{\text{HS}}(\ell) = 2d_\ell^D + \nu d_0^D$ for Alg. 3. In particular, on the finest level

$$\mathcal{M}_{\text{HS}}(L-1) = \left(2 + 2^{-(L-1)D}\nu\right)d^D < \mathcal{M}_{\text{HMC}}. \tag{41}$$

To obtain the memory requirements of the MLMC method in Alg. 4 note that on each level both the current state $\boldsymbol{x}_\ell^{(t)}$ and a proposal $\boldsymbol{y}_\ell$ have to be stored. In addition, the storage requirements of the hierarchical sampler in Alg. 3 have to be taken into account on all but the very finest level. Consequently for $L \ge 2$ levels the total amount of required memory is

$$\begin{aligned}\mathcal{M}_{\text{MLMC}} &= 2\sum_{\ell=0}^{L-1}d_\ell^D + \sum_{\ell=0}^{L-2}\mathcal{M}_{\text{HS}}(\ell) \\ &= \left(2 + 4\frac{1-2^{-(L-1)D}}{2^D-1} + \nu\frac{L-1}{2^{(L-1)D}}\right)d^D \\ &< (6+\nu)d^D < 3\mathcal{M}_{\text{HMC}}.\end{aligned} \tag{42}$$

As Eqs. (41) and (42) show, the memory footprint of the hierarchical sampler in Alg. 3 is actually smaller than that of HMC, whereas the MLMC method in Alg. 4 requires less than three times the amount of storage used by a standard HMC method for any dimension D. Limited storage usually restricts the size of systems that can be simulated for higher dimensions ($D \geq 3$). As the second line of Eq. (42) shows, for those higher dimensional problems the additional memory overhead of MLMC (compared to HMC) is actually less than 30%.

D Rejection sampling

To draw samples from the distribution $p_{\sigma,\delta x}$ defined in Eq. (31) we use rejection sampling with a Gaussian envelope, as described in the following algorithm:

Algorithm 6 Rejection sampling for distribution $p_{\sigma,\delta x}$ defined in Eq. (31)

```
loop
    Draw sample $x$ from Gaussian distribution $g_\sigma$
    with $g_\sigma(x) = \sqrt{\frac{2\sigma}{\pi^3}} \exp\left[-\frac{2\sigma}{\pi^2}x^2\right]$.
    if $-\pi \leq x \leq \pi$ then
        Draw uniformly distributed random $u \in [0,1)$.
        if $u \leq \exp\left[-2\sigma\left(\sin^2\left(\frac{x}{2}\right) - \frac{x^2}{\pi^2}\right)\right]$ then
            return $x + \delta x$
        end if
    end if
end loop
```

E Fixed tolerance on the total error

While for the results presented in the main text we fixed ϵ_{stat} and varied the tolerance on the discretisation error, in the following we also show the (estimated) runtime as a function of the tolerance ϵ on the total root mean square error. For this we set $\epsilon_{\text{stat}} = \epsilon_{\text{disc}} = \epsilon/\sqrt{2}$ as is common in the multilevel Monte Carlo literature. Figs. 15 and 16 show the runtime of the single-level HMC method, the hierarchical sampler and the multilevel method as a function of the tolerance ϵ on the total error; they should be compared to Figs. 10 and 11. Finally, Fig. 17 shows the runtime of the standard cluster-sampler and the multilevel-accelerated variant of the method; the corresponding plot in the main text is Fig. 13.

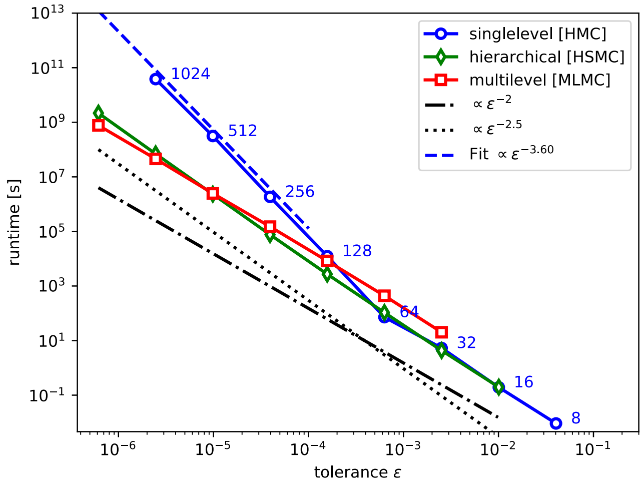

Figure 15: Estimated runtime of different Monte Carlo sampling algorithms for the double well potential. Results are shown in seconds and as a function of the tolerance ϵ on the total error. The data points are labelled with the number of dimensions d for each lattice spacing.

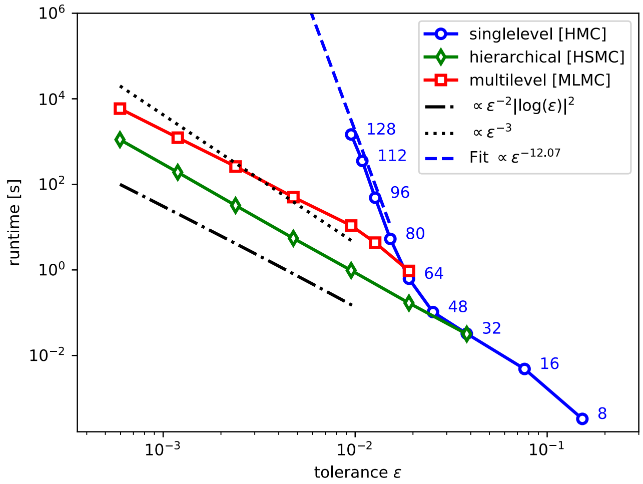

Figure 16: Runtime of different Monte Carlo sampling algorithms for the topological oscillator. Results are shown in seconds and as a function of the tolerance ϵ on the total error. The data points are labelled with the number of dimensions d for each lattice spacing.

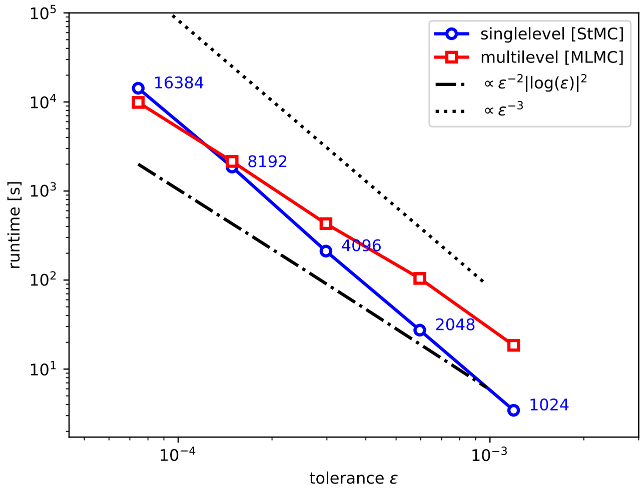

Figure 17: Runtime of singlelevel (StMC) and multilevel (MLMC) cluster algorithm for the topological oscillator. Results are shown in seconds and as a function of the tolerance ϵ on the total error. The data points are labelled with the number of dimensions d for each lattice spacing.

References

[1] Richard P Feynman, Albert R Hibbs, and Daniel F Styer. *Quantum mechanics and path integrals.* Courier Corporation, 2010.

[2] M Creutz and B Freedman. A statistical approach to quantum mechanics. *Annals of Physics*, 132(2):427–462, 1981.

[3] Heinz J Rothe. *Lattice Gauge Theories: An Introduction Third Edition*, volume 74. World Scientific Publishing Company, 2005.

[4] Detar Carleton et al. *Lattice methods for quantum chromodynamics.* World Scientific, 2006.

[5] Martin Lüscher. Computational strategies in lattice QCD. *Mod. Persp. in Lat. QCD*, pages 331–399, 2010.

[6] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *J Chem. Phys*, 21(6):1087–1092, 1953.

[7] W Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[8] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

[9] Michael B Giles. Multilevel monte carlo path simulation. *Oper. Res.*, 56(3):607–617, 2008.

[10] Stefan Schaefer, Rainer Sommer, Francesco Virotta, Alpha Collaboration, et al. Critical slowing down and error analysis in lattice QCD simulations. *Nucl. Phys. B*, 845(1):93–119, 2011.

[11] Edward Witten. Current algebra theorems for the U (1) "Goldstone boson". *Nucl. Phys. B*, 156(2):269–283, 1979.

[12] Gabriele Veneziano. U(1) without instantons. *Nucl. Phys. B*, 159(1-2):213–224, 1979.

[13] Martin Lüscher and Stefan Schäfer. Lattice qcd without topology barriers. *J. High Energy Phys.*, 2011(7):36, 2011.

[14] Martin Lüscher and Stefan Schäfer. Lattice QCD with open boundary conditions and twisted-mass reweighting. *Comput. Phys. Commun.*, 184(3):519–528, 2013.

[15] J Andrés Christen and Colin Fox. Markov chain Monte Carlo using an approximation. *Journal of Computational and Graphical statistics*, 14(4):795–810, 2005.

[16] Stefan Heinrich. Multilevel monte carlo methods. In *Int. Conf. on Large-Scale Sci. Comp.*, pages 58–67. Springer, 2001.

[17] Tim J Dodwell, Chris Ketelsen, Robert Scheichl, and Aretha L Teckentrup. A hierarchical multilevel Markov chain Monte Carlo algorithm with applications to uncertainty quantification in subsurface flow. *J. on Uncert. Quant.*, 3(1):1075–1108, 2015.

[18] TJ Dodwell, C Ketelsen, R Scheichl, and AL Teckentrup. Multilevel markov chain monte carlo. *Siam Review*, 61(3):509–545, 2019.

[19] K.A. Cliffe, M.B. Giles, R. Scheichl, and A.L. Teckentrup. Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Comp. and Vis. in Sci.*, 14(1):3, 2011.

[20] R Scheichl, AM Stuart, and AL Teckentrup. Quasi-Monte Carlo and multilevel Monte Carlo methods for computing posterior expectations in elliptic inverse problems. *SIAM/ASA J. on Uncert. Quant.*, 5(1):493–518, 2017.

[21] TJ Dodwell, S Kinston, R Butler, RT Haftka, Nam H Kim, and R Scheichl. Multilevel Monte Carlo Simulations of Composite Structures with Uncertain Manufacturing Defects. *arXiv preprint arXiv:1907.10271*, 2019.

[22] Kurt Symanzik. Continuum limit and improved action in lattice theories:(I). Principles and φ^4 theory. *Nuclear Physics B*, 226(1):187–204, 1983.

[23] Martin Lüscher and Peter Weisz. On-shell improved lattice gauge theories. *Communications in Mathematical Physics*, 97(1-2):59–77, 1985.

[24] Martin Lüscher, Stefan Sint, Rainer Sommer, Peter Weisz, and Ulli Wolff. Non-perturbative $\mathcal{O}(a)$ improvement of lattice QCD. *Nuclear Physics B*, 491(1-2):323–343, 1997.

[25] Karl Jansen and Rainer Sommer. $\mathcal{O}(a)$ improvement of lattice QCD with two flavors of Wilson quarks. *arXiv preprint hep-lat/9803017*, 1998.

[26] Andreas Ammon, Alan Genz, Tobias Hartung, Karl Jansen, Hernan Leövey, and Julia Volmer. On the efficient numerical solution of lattice systems with low-order couplings. *Computer Physics Communications*, 198:71–81, 2016.

[27] Ulli Wolff. Collective Monte Carlo updating for spin systems. *Physical Review Letters*, 62(4):361, 1989.

[28] Jonathan Goodman and Alan D Sokal. Multigrid Monte Carlo method for lattice field theories. *Physical Review Letters*, 56(10):1015, 1986.

[29] Wolfhard Janke and Tilman Sauer. Multicanonical multigrid monte carlo method. *Physical Review E*, 49(4):3475, 1994.

[30] KE Schmidt. Using renormalization-group ideas in Monte Carlo sampling. *Physical Review Letters*, 51(24):2175, 1983.

[31] M Faas and HJ Hilhorst. Hierarchical Monte Carlo simulation of the Ising model. *Physica A*, 135(2-3):571–590, 1986.

[32] Michael G Endres, Richard C Brower, William Detmold, Kostas Orginos, and Andrew V Pochinsky. Multiscale Monte Carlo equilibration: Pure Yang-Mills theory. *Physical Review D*, 92(11):114516, 2015.

[33] Robert H Swendsen and Jian-Sheng Wang. Nonuniversal critical dynamics in Monte Carlo simulations. *Physical Review Letters*, 58(2):86, 1987.

[34] Jian-Sheng Wang and Robert H Swendsen. Cluster monte carlo algorithms. *Physica A*, 167(3):565–579, 1990.

[35] S Chen, Alan M Ferrenberg, and DP Landau. Monte Carlo simulation of phase transitions in a two-dimensional random-bond Potts model. *Physical Review E*, 52(2):1377, 1995.

[36] Ulli Wolff, Alpha Collaboration, et al. Monte Carlo errors with less errors. *Computer Physics Communications*, 156(2):143–153, 2004.

[37] Abdul-Lateef Haji-Ali, Fabio Nobile, and Raúl Tempone. Multi-index Monte Carlo: when sparsity meets sampling. *Numerische Mathematik*, 132(4):767–806, 2016.