Path integral Monte Carlo: Lattice QCD

Damiano Scevola, Federico Tonetto

October 2024

Path Integral Approach to Quantum Mechanics

Our goal is to compute quantum mechanical observables using the path integral approach. The central quantity we aim to compute is the correlation function:

$$\langle x(t_2)x(t_1)\rangle = G(t) = \frac{\int Dx(t)x(t_2)x(t_1)e^{-S[x]}}{\int Dx\, e^{-S[x]}},$$

which represents a weighted average over paths, where the weight is determined by the action $S[x]$. To do so, we discretize the path $t_j = t_0 + ja, \quad j = 0, ..., N$.

Numerical calculation of the propagator

Before calculating the correlation function, we see another application of the path integral formalism: the computation of the propagator. To do so, it is possible to shift the problem from quantum mechanics to numerical integration, indeed

$$\langle x| e^{-HT} |x\rangle \simeq A \int_0^{aN} dx_1 ... dx_{N-1} e^{-S_{lat}[x]},$$

hence, by computing the path integral on the left side, it is possible to compute the propagator given the ground state and the ground state energy. We have than it numerically by using the Vegas library. We observe that, in dimension greater than 1, this procedure is not efficient.

Harmonic oscillator

We have computed the propagator for the harmonic oscillator.

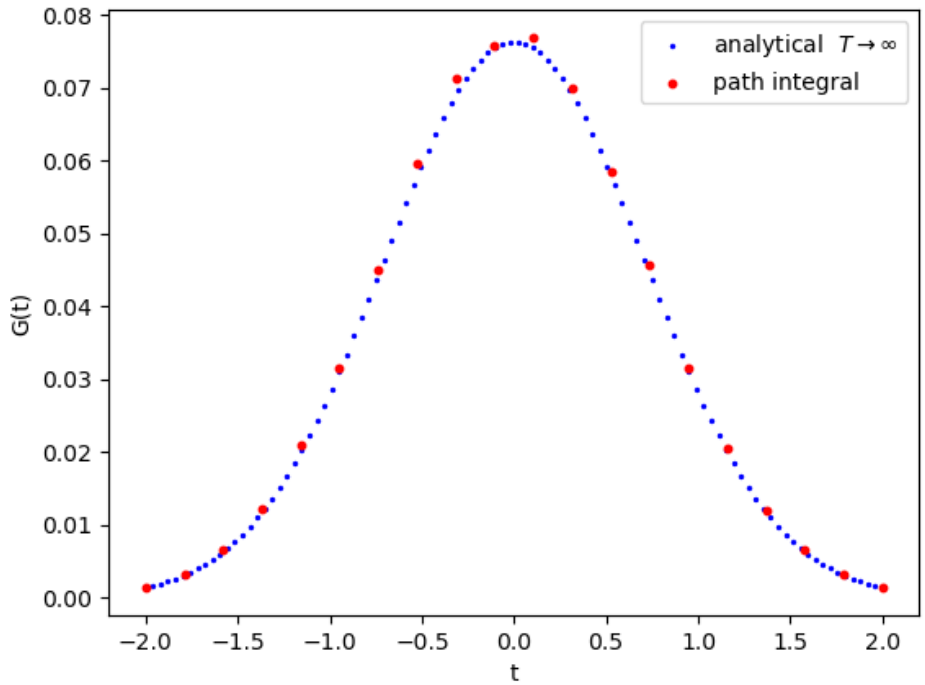


Figure: Propagators value of the harmonic oscillator versus time.

Anharmonic oscillator

We have also done it for the anharmonic oscillator $V(x) = \frac{x^4}{4}$

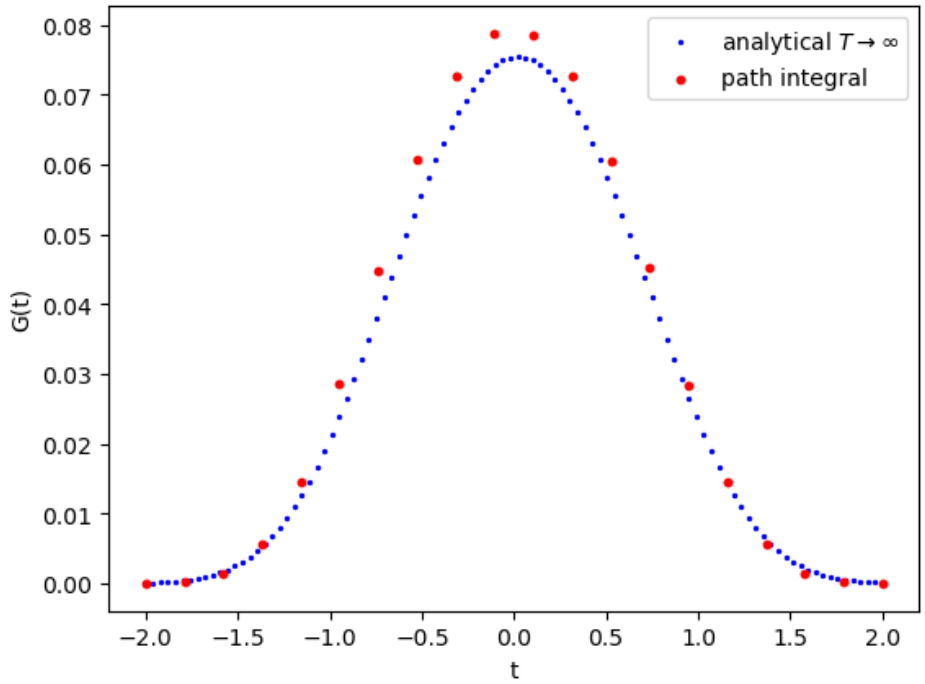


Figure: Propagators value of the anharmonic oscillator versus time. We observe that the values are shifted due to renormalization effects.

Monte Carlo Approach

A more convenient way to compute path integrals is to generate a large number of configurations at each time step:

$$x(n) = (x_1, \ldots, x_{N_{cf}}), \quad n = a, 2a, \ldots, Na, \text{ with } T = Na,$$

where the configurations follow the probability distribution determined by $e^{-S[x]}$. For this, we use the Metropolis algorithm.

Metropolis Algorithm

```
def update_path(path, S_per_timeslice, eps: np.float64):
    N = path.shape[0]
    for j in range(N):
        # Save the original value
        old_x = path[j]
        old_Sj = S_per_timeslice(j, path)
        # Propose a new value for path[j]
        path[j] += np.random.uniform(-eps, eps)
        # Calculate the change in action
        dS = S_per_timeslice(j, path) - old_Sj
        # Accept or reject the new value based on Metropolis
        if dS > 0 and np.exp(-dS) < np.random.uniform(0, 1):
            # Restore old value if not accepted
            path[j] = old_x
```

This algorithm ensures that paths with lower action are favored while allowing for random fluctuations.

Monte Carlo Process Overview

In general, we perform the following steps:

- Thermalize the lattice by updating all the paths for a given number of steps ($\approx 5 \times N_{cor}$).
- Update the path N_{cor} times, then compute a functional of interest. Repeat this process $N \times N_{cf}$ times.
- Store the results in a matrix.
- Finally, compute averages and errors from the matrix.

Improvements: Bootstrap Procedure

```
for i in range(N_copies):
    s = np.zeros((N_cf, N), dtype=np.float64)
    for row in range(N_cf):
        i = np.int32(np.random.uniform(0, N_cf))
        for n in range(N):
            s[row][n] = functional_samples[i][n]
    for n in range(N):
        avgs[i, n] = np.sum(s[:, n]) / N_cf
        stds[i, n] = np.sum((s[:, n] - avgs[i, n]) ** 2)
                     / (N_cf * np.sqrt(N_cf))
return avgs, stds
```

Binning Method

The binning procedure is useful when N_{cf} is large and N_{cor} is low. Instead of storing all N_{cf} values, we store only a subset. To achieve this, we define a bin size, bin_{size}, and store only the average of the points within each bin.

Applications

We have computed the difference between the first excited state and the ground state using the formula:

$$\Delta E_n = \log \left| \frac{G_n}{G_{n+1}} \right|$$

for both the harmonic oscillator and the anharmonic oscillator (where $V = \frac{x^4}{4}$). Additionally, we computed the cubic correlator:

$$\langle x^3(t_1) x^3(t_2) \rangle$$

for the harmonic oscillator.

Results for the Harmonic Oscillator

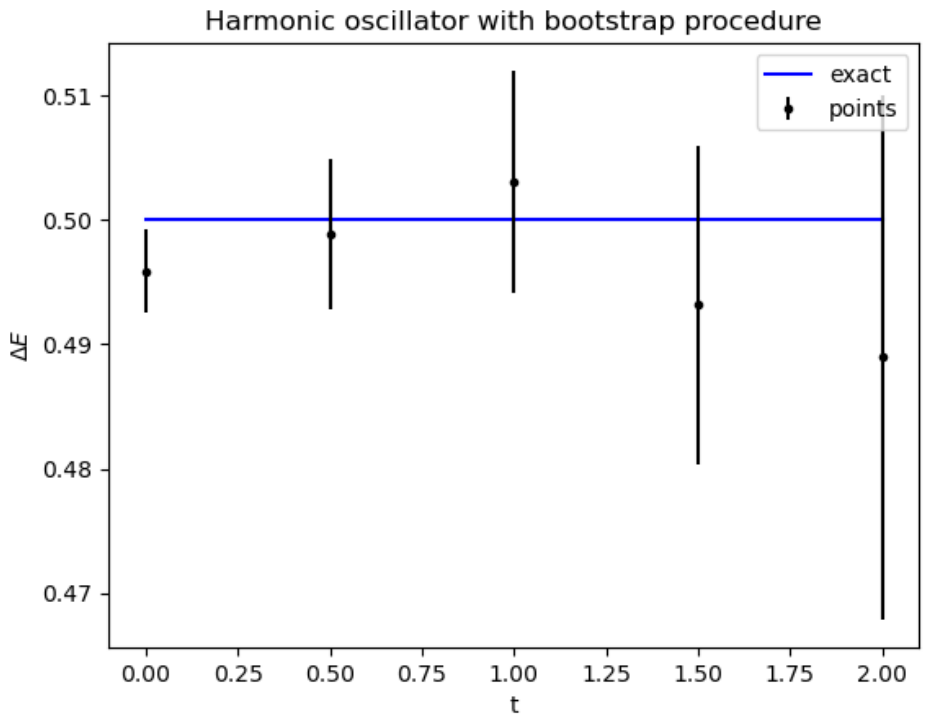


Figure: Monte Carlo values $\Delta E(t) = \log\left(\frac{G_n}{G_{n+1}}\right)$ plotted versus t for the harmonic oscillator. The exact result, $\Delta E = \frac{1}{2}$, is shown with a line.

Results for the Anharmonic Oscillator

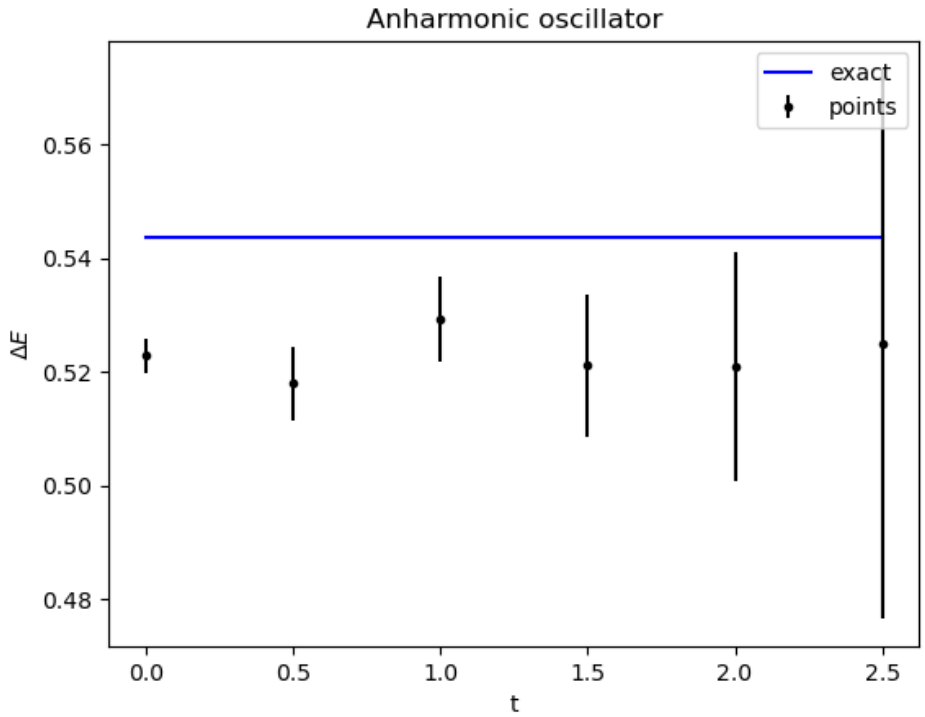


Figure: Monte Carlo values $\Delta E(t) = \log\left(\frac{G_n}{G_{n+1}}\right)$ for the anharmonic oscillator. The exact result, ΔE, is shown with a line and is computed by solving the Schrödinger equation. Renormalization causes a shift.

Cubic correlator

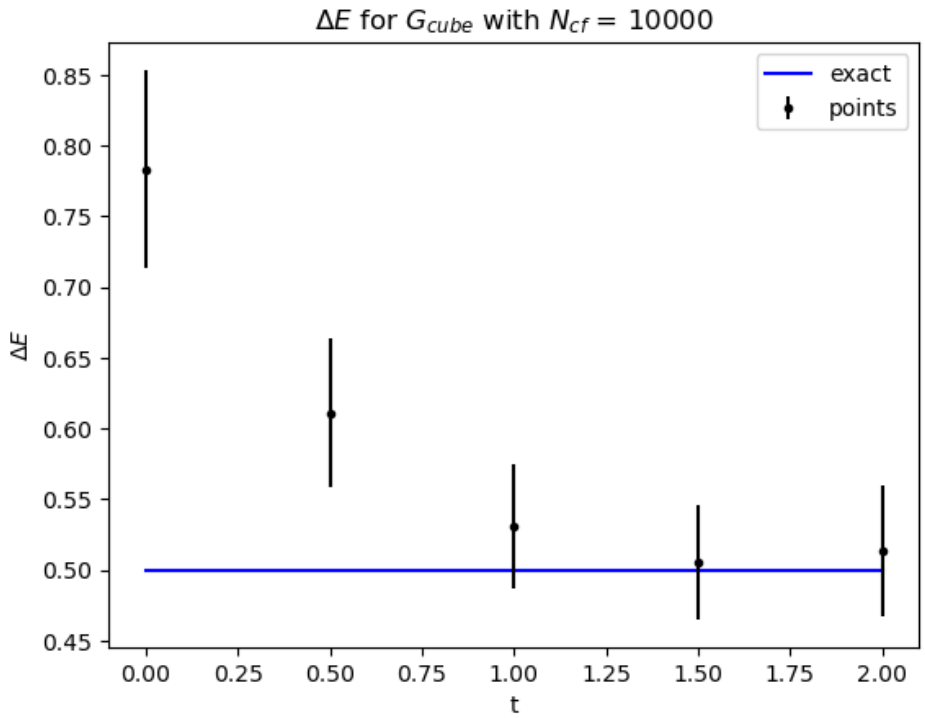


Figure: Monte Carlo values of $\Delta E(t) = \log\left(\frac{G_n}{G_{n+1}}\right)$ for the harmonic oscillator. The exact value is reached from above when the source and sink are the same.

A Comment on Thermalization and Binning

If N_{cor} is too small, the values become statistically correlated, leading to unreliable error estimates. This issue can be mitigated by increasing the bin size.

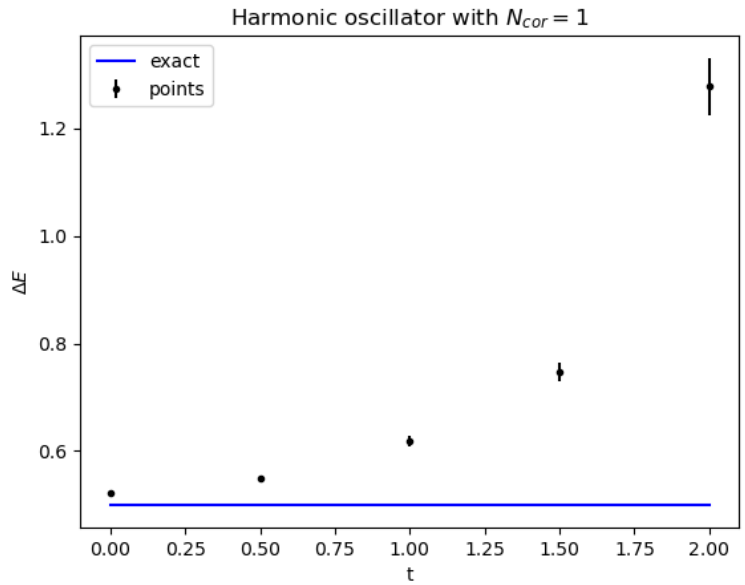


Figure: Monte Carlo values $\Delta E(t)$ for the harmonic oscillator with $N_{cor} = 1$. The error bars are unreliable.

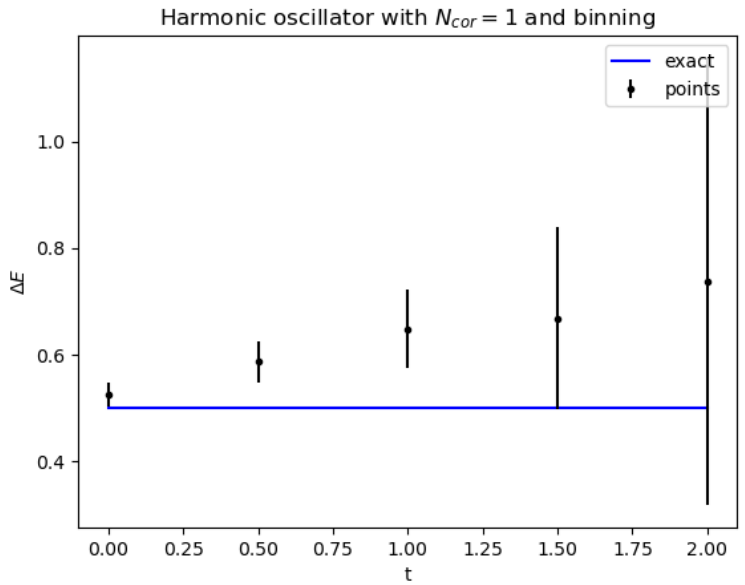


Figure: Monte Carlo values $\Delta E(t)$ with $N_{cor} = 1$ and $bin_{size} = 20$. The error bars looks like before.

Derivative in Lattice Approximation

On the lattice, derivatives are replaced by differences. We use the following approximation:

$$\frac{\partial^2 \phi}{\partial x^2} = \Delta_x^{(2)} \phi - \frac{a^2}{12} \left(\Delta_x^{(2)} \right)^2 \phi + O(a^4),$$

where

$$\Delta_x^{(2)} = \frac{\phi(x+a) - 2\phi(x) + \phi(x-a)}{a^2}.$$

Harmonic oscillator with improved action

To make use of the previous formula, it is possible to integrate by parts inside the action and, after the discretization, obtain the following action

$$S_{imp}[x] = \sum_{j=0}^{N-1} a \left[-\frac{1}{2} x_j \left(\Delta^{(2)} - a^2 \frac{(\Delta^{(2)})^2}{12} - 1 \right) x_j \right].$$

We compute again ΔE but now we update the path with the new action.

Numerical Ghosts

By solving the equation of motion for the harmonic oscillator, it is possible to find two different frequency solutions. The first corresponds to an improvement over the non-improved action, while the second is a so-called "numerical ghost". This ghost introduces an additional state that shifts the values of ΔE, causing the energies to rise from below.

Change of Coordinates

To eliminate the numerical ghosts, we can use a transformation that removes the extra states. This is done through an infinitesimal coordinate transformation, which effectively shifts the potential. After applying this transformation, the action becomes:

$$S_{\text{imp}}[x] = \sum_{j=0}^{N-1} a \left[-\frac{1}{2} x_j \Delta^{(2)} x_j + V_{\text{imp}}(x_j) \right],$$

where the improved potential is given by:

$$V_{\text{imp}}(x_j) = \frac{1}{2} x_j^2 \left(1 + \frac{a^2}{12} \right) + O(a^4).$$

Results for the Harmonic Oscillator

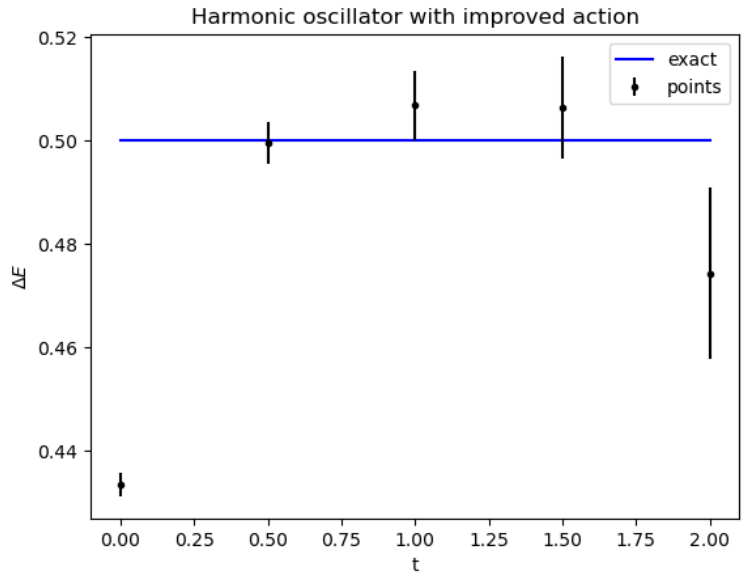


Figure: Monte Carlo values of $\Delta E(t) = \log\left(\frac{G_n}{G_{n+1}}\right)$ plotted versus t for the harmonic oscillator using the improved action. A numerical ghost with negative norm is observed.

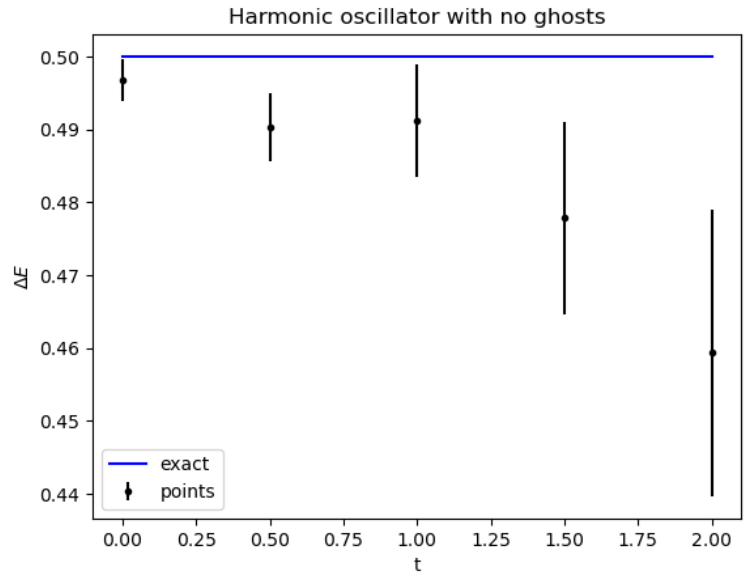


Figure: Monte Carlo values $\Delta E(t) = \log\left(\frac{G_n}{G_{n+1}}\right)$ plotted versus t for the harmonic oscillator using the improved potential. There are not states with negative norm.

Ghosts in the Anharmonic Oscillator

We also considered the case of the anharmonic oscillator, where the potential is given by:

$$V(x) = \frac{1}{2}x^2(1 + 2x^2).$$

Even in this case, a change of coordinates can be applied to remove the ghosts. With this transformation, the improved potential becomes:

$$V_{\text{imp}}(x) = \frac{1}{2}x^2(1 + 2x^2) + \frac{a^2}{24}\left(x + 4x^3\right)^2 - \frac{ax^2}{2} + \frac{a^3x^4}{8}.$$

Results for the Anharmonic Oscillator

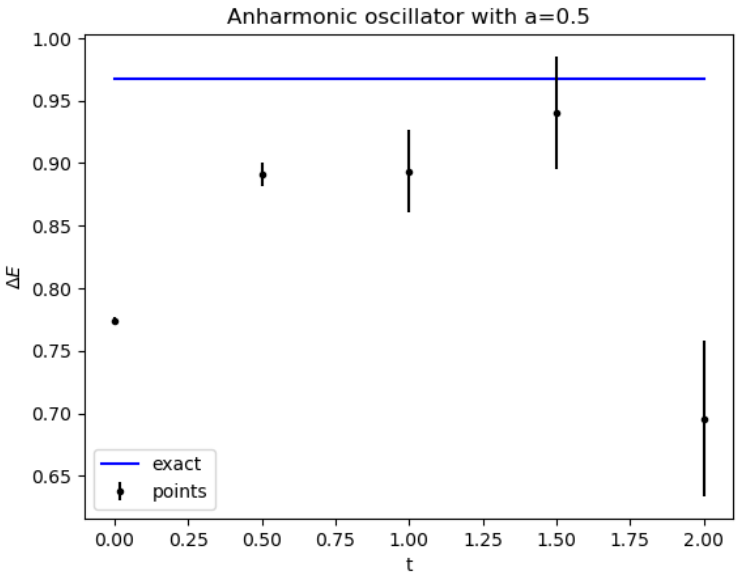


Figure: Monte Carlo values of $\Delta E(t) = \log\left(\frac{G_n}{G_{n+1}}\right)$ plotted versus t for the anharmonic oscillator using the improved action. A ghost with negative norm is observed.

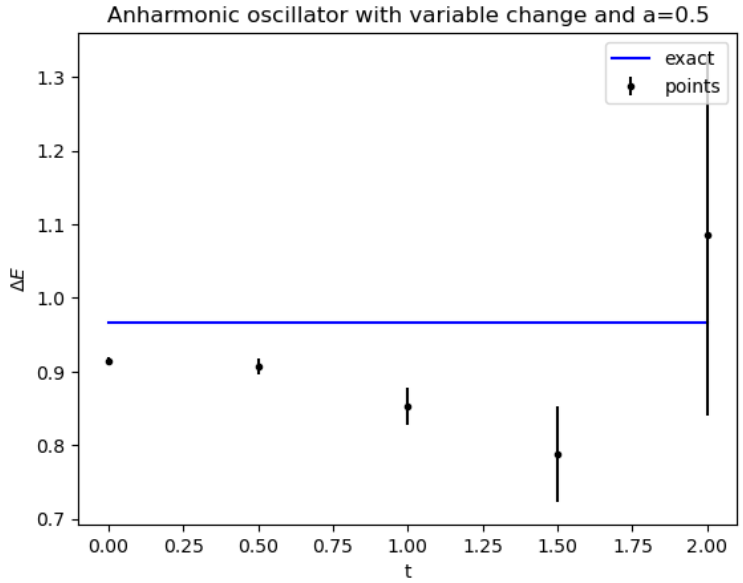


Figure: Monte Carlo values of $\Delta E(t) = \log\left(\frac{G_n}{G_{n+1}}\right)$ plotted versus t for the anharmonic oscillator using the improved potential. There are not states with negative norm.

Renormalization Effects

As is typical with non-free potentials, there are renormalization effects to consider. In the case of the anharmonic oscillator, the change of coordinates does not account for these effects, which can introduce errors of order $\mathcal{O}(a)$.

Calculation of the renormalization effects

To compute the renormalization effects we have calculated the exact ΔE by solving the Schroedinger equation. Then we have computed the difference between the exact ΔE and the average of the first 5 points of the path integral. We have done this both for $a = \frac{1}{2}$ and $a = \frac{1}{4}$. We have obtained

$$a = 0.5 \Rightarrow \delta = 0.11$$
$$a = 0.25 \Rightarrow \delta = 0.04.$$

Calculation of the correction factor

By adjusting the coefficient of x^2 in the potential it is possible to reduce this effect. Let z be that coefficient. We know that there exists a value of z that eliminates the discrepancy. Hence we have created a function that, given an interval, computes the smallest sub interval in which the theorem of the zeros still holds. We have parallelized the process by dividing the interval in a given numbers of slices.

Result

We have obtained the following value

$$z = 1.02 \pm 0.31.$$

With this value ΔE becomes

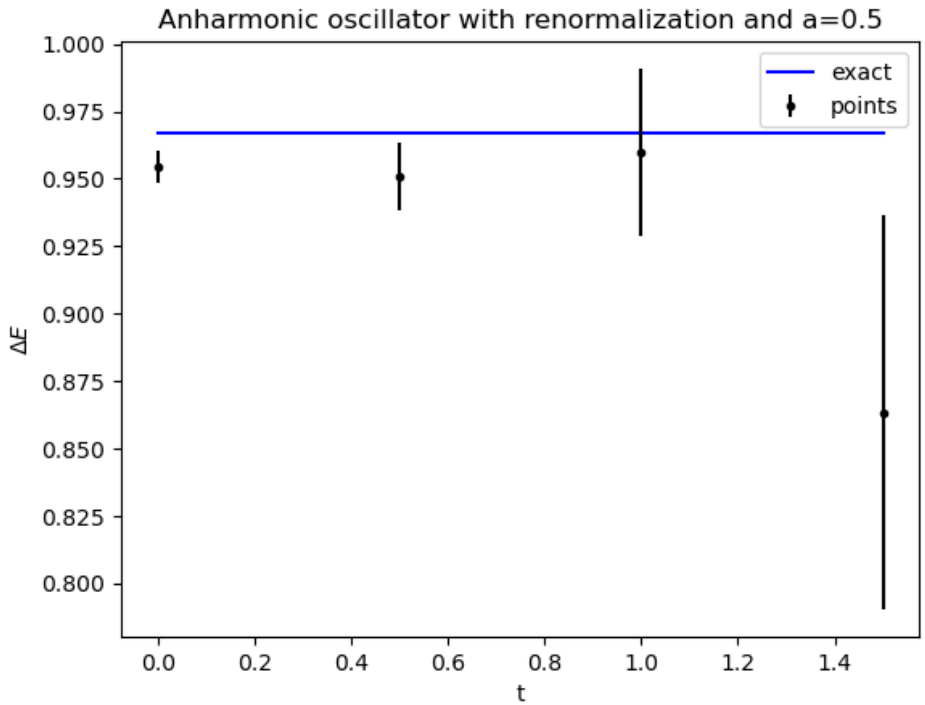


Figure: Monte Carlo values of $\Delta E(t) = \log\left(\frac{G_n}{G_{n+1}}\right)$ plotted versus t for the anharmonic oscillator using the correct value of z.

Lattice QCD

Field discretization

Discretize spacetime: N points per spacetime direction

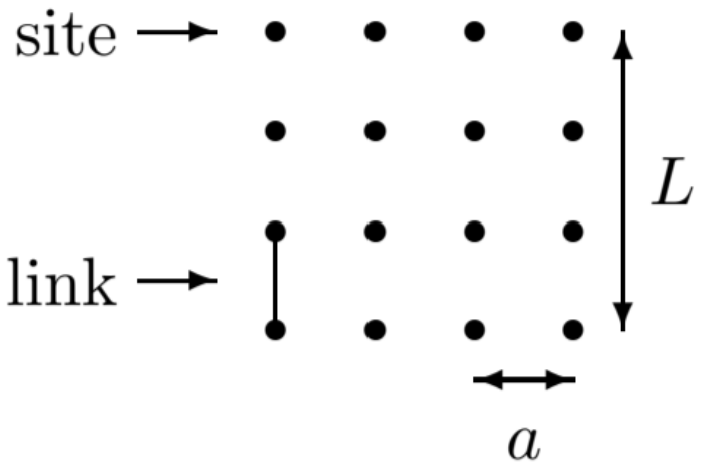


Figure: Lattice discretization for $d = 2$, $N = 4$

Field configuration encoding

- Each link is an $SU(3)$ matrix
- There is a link for each spacetime direction, for each spacetime point ($d \cdot N^d$ links in total).
- Use a complex `numpy` array with shape $(N^d, d, 3, 3)$.
- Nodes indexed by integers $i \in \{0, \ldots, N^d - 1\}$
- Spacetime coordinates: ordered array (length d) of digits given by the base-N conversion of the index i.
- Periodic boundary conditions

Paths and loops

Wilson loops:

$$W(\mathcal{C}) = \frac{1}{3} Re\ tr(\mathcal{C})$$

$\mathcal{C}$: product of links

```
compute_path(links, [1,0], [+1,+2,+1,+2,-1,-1,-2,-2])
```

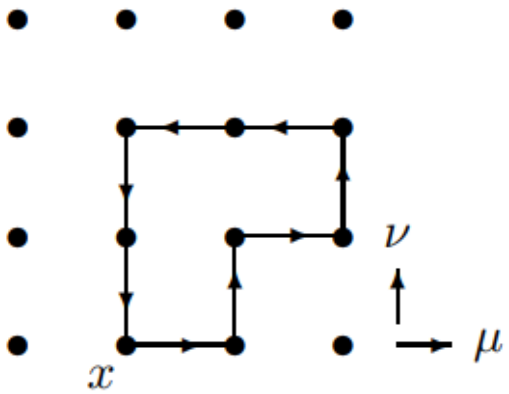


Figure: Example of a Wilson loop in a 4 x 4 2d lattice. $\mu \leftrightarrow 1$, $\nu \leftrightarrow 2$

Plaquettes and Wilson action

Wilson action:

$$S_{Wil} = \beta \sum_{x,\mu>\nu} (1 - P_{\mu\nu}(x))$$

where $P_{\mu\nu}(x)$ is the plaquette operator.

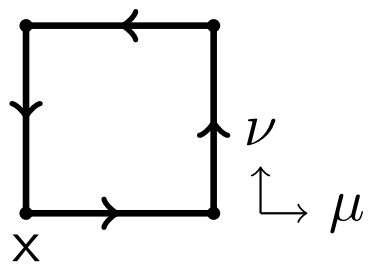


Figure: Plaquette operator $P_{\mu\nu}(x)$

- This is the most local, gauge invariant action that one can build.
- Poincarè symmetry is discretized.
- This action is the correct one in the limit $a \to 0$
- Discretization error of order $\mathcal{O}(a^2)$

Metropolis update for Wilson action

- Update $U_\mu(x)$ `hits=10` times
- Updating once means $U_\mu(x) \to MU_\mu(x)$ where $M \in SU(3)$ is a random SU(3) matrix s.t. $|M - \mathbb{I}_3| \sim \epsilon \ll 1$
- Check metropolis acceptance condition each hit
- $\Delta S = S[MU_\mu(x) - U_\mu(x)]$
- Factorize $U_\mu(x)$ and compute $\Gamma_\mu(x)$ before `hits` to save time.

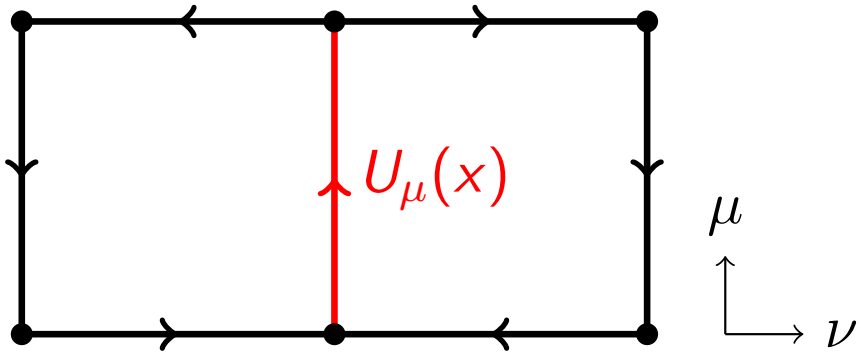


Figure: Plaquettes contributing to the action in each spacetime direction.

Rectangles and improved action

Improved action gives $\mathcal{O}(a^4)$ error (instead of $\mathcal{O}(a^2)$).

$$S = -\beta \sum_{x,\mu>\nu} \left[\frac{5}{3} \frac{P_{\mu\nu}(x)}{u_0^4} - \frac{R_{\mu\nu}(x) + R_{\nu\mu}(x)}{12 u_0^6} \right]$$

$R_{\mu\nu}(x)$ is the rectangle operator
$u_0 \in \mathbb{R}$ accounts for tadpole improvement

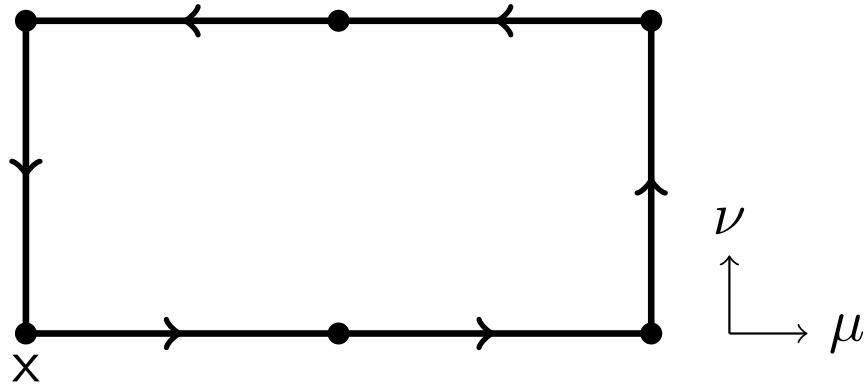


Figure: Rectangle operator $R_{\mu\nu}(x)$

Metropolis update for improved action

Six rectangle contributions $\forall \nu$

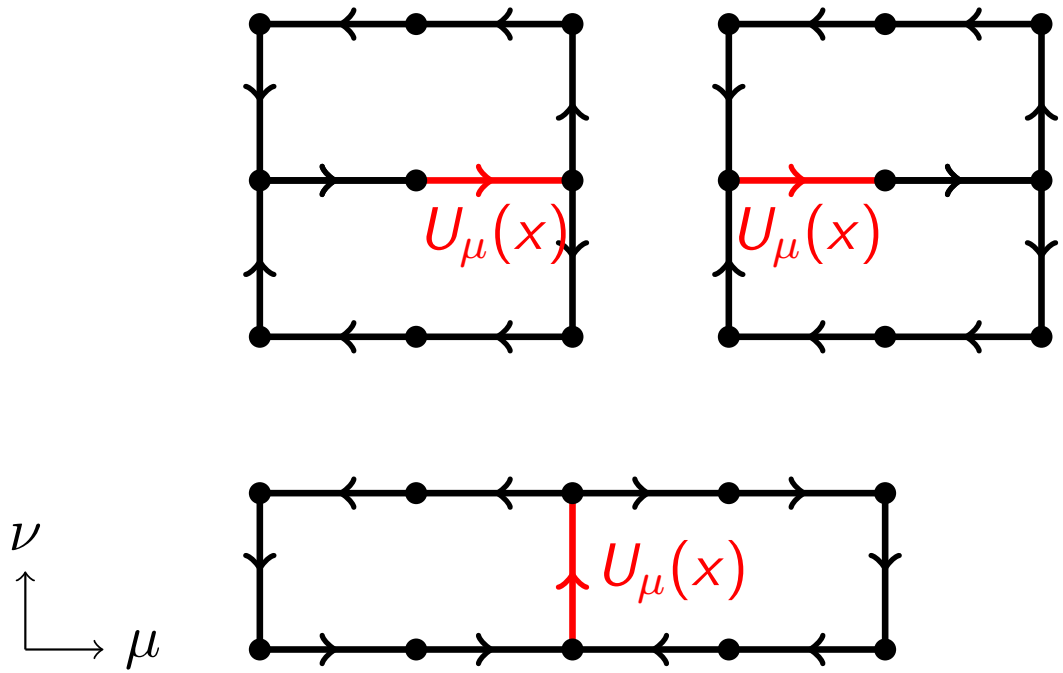


Figure: Rectangles contributing to the action in each spacetime direction where the link $U_\mu(x)$ appears.

Rotational invariance of the lattice

- Lattice is symmetric under permutations of spacetime directions (i.e. rotations)
- We can draw more samples for each lattice update.
- `compute_wilson_loop_average_fixed_frame(links, loops, rotated_axes)` exploits only translational invariance
- `rotated_axes` is a permutation of the first d integers
- `generate_wilson_samples` calls `compute_wilson_loop_average_fixed_frame` $d!$ times (or $(d-1)!$ times, see smearing) for each loop, for each sample.

Initialization and thermalization

- All lattice links are initialized to $\mathbb{I}_3$
- Perform `N_cor * thermalization_its` iterations of a full lattice update to thermalize
- After this, start drawing samples

Computing Wilson loop averages

- Prepare a list of loops in step notation (no symmetry-equivalent loops shall be included)
- For each rotational configuration, for each loop in the list, we call `compute_wilson_loop_average_fixed_frame`
- Average results
- Repeat `N_cf` times
- Apply bootstrap procedure for error estimation

Example of $a \times a$ and $a \times 2a$ Wilson loops

With $N = 8$ and $a = 0.25$ *fm* ($\beta = 5.5$ for Wilson action, $\beta = 1.719$ and $u_0 = 0.797$ for improved action), we computed the following two loops:

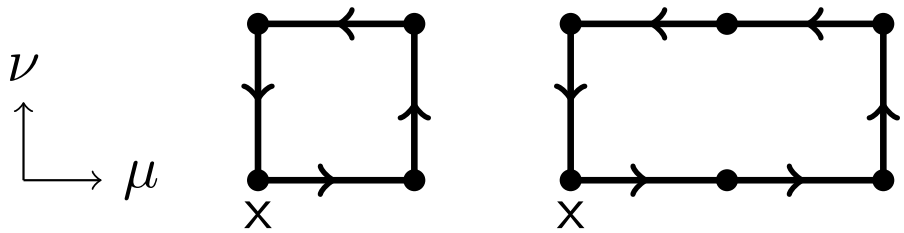


Figure: Wilson loops $a \times a$ and $2a \times a$

Loop	Wilson	Expected	Improved	Expected
$a \times a$	0.498 ± 0.001	0.50	0.5413 ± 0.0005	0.54
$2a \times a$	0.261 ± 0.001	0.26	0.2832 ± 0.0006	0.28

Static quark-antiquark potential

Potential between a quark q and the corresponding antiquark $\bar{q}$ in the static $m_q \to \infty$ limit given by:

$$W(r,t) \xrightarrow{t\to\infty} const\ e^{-V(r)t}$$

$W(r,t)$ is a Wilson loop like:

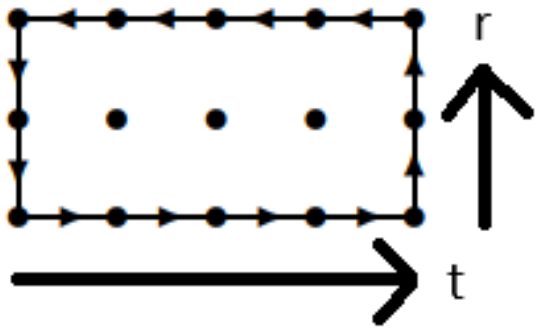


Figure: Wilson loop for computing the potential.

If we compute $W(r,t)$ and $W(r,t+a)$ with large t, we have:

$$a \cdot V(r) = \log \left| \frac{W(r,t)}{W(r,t+a)} \right|$$

Smearing

Numerical trick to hasten convergence: link smearing after first thermalization.

- Copy the current field configuration
- For each purely spacial link, apply smearing operator $(1 + \epsilon a^2 \Delta^{(2)})$,
- $\Delta^{(2)}$ is the discretized gauge covariant derivative (use old links)
- Repeat n times

Rotational symmetry broken between space and time.

Loops contributing to the first points of the potential

Spacial separations of the quark-antiquark pair up to $r = 3$ (in units of a):

(0,0,1)	(0,1,1)	(1,1,1)
(0,0,2)	(0,1,2)	(1,1,2)
(0,2,2)	(1,2,2)	(0,0,3)

These tuples are generated in non-decreasing order once, and then they are rotated in all the 6 rotational configurations of the spacial directions. We will have $V(r)$ points for r equals (in units of a):

$$1, \sqrt{2}, \sqrt{3}, 2, \sqrt{5}, \sqrt{6}, \sqrt{8}, 3$$

There will be two points for $r = 3$.

Non-planar Wilson loops

Given a spacial separation (x, y, z) between the two quarks loops are constructed as follows:

- start from the first quark in $(0, 0, 0)$
- do steps only in spacial directions to reach (x, y, z)
- do t (or $t + a$) steps in the temporal direction
- do steps only in spacial directions to reach $(0, 0, 0)$
- do t (or $t + a$) steps in the negative temporal direction

The spacial part of the loop can be arbitrairly complicated. However, the leading order contribution is given only by least-steps loops.
If there is more than one non-zero coordinate, loops are non-planar.

Static quark-antiquark potential: results and fit

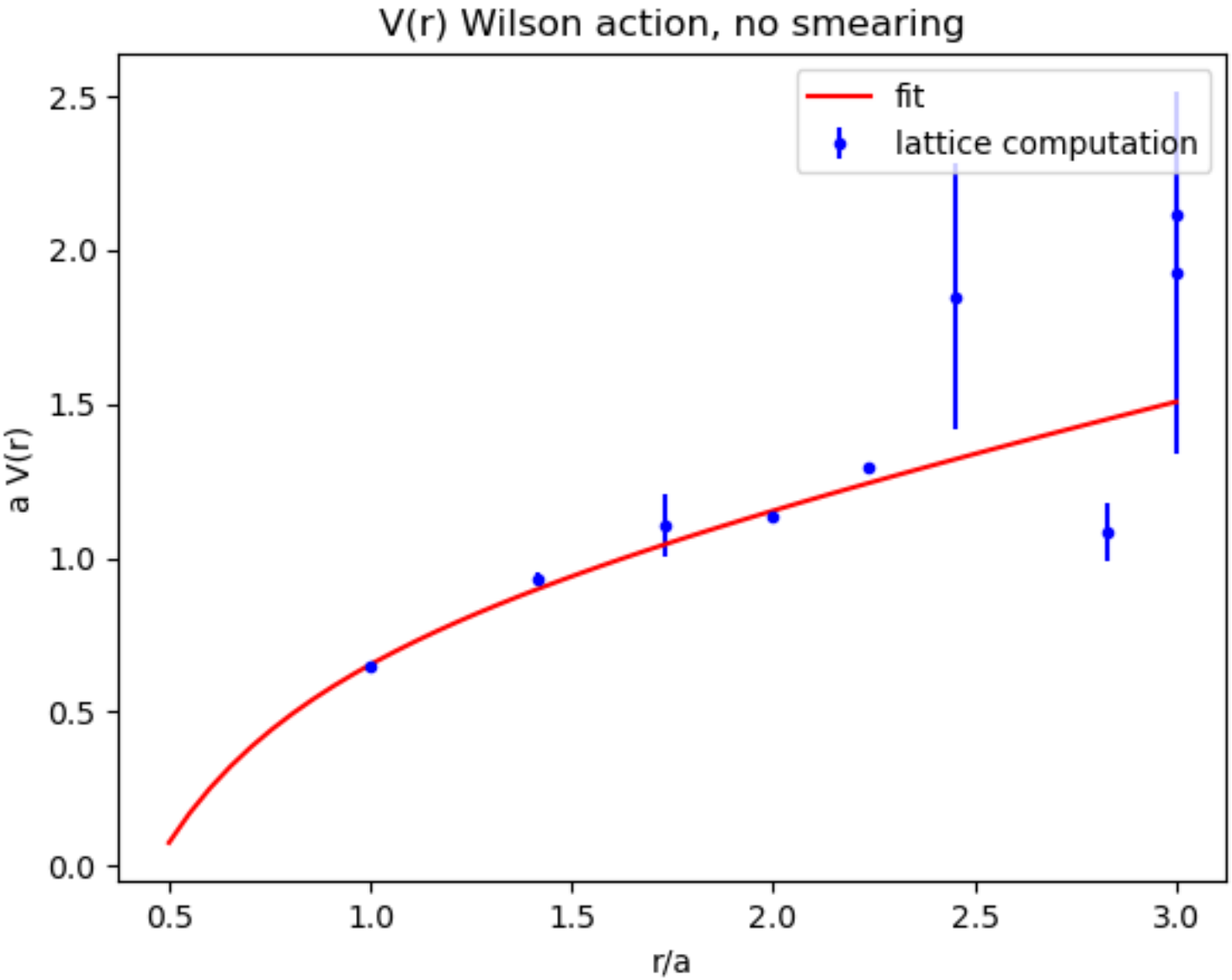
The parameters of our simulation:

- $N = 8$ lattice points in each spacetime direction
- $a = 0.25\ fm$
- $\beta = 5.5$ for Wilson action
- $\beta = 1.719$ and $u_0 = 0.797$ for improved action
- $\epsilon_{smear} = 1/12$
- $n_{smear} = 4$
- $N_{cor} = 50$
- $N_{cf} = 5$
- $N_{cor} = 50$
- $t = 2$ loop temporal width

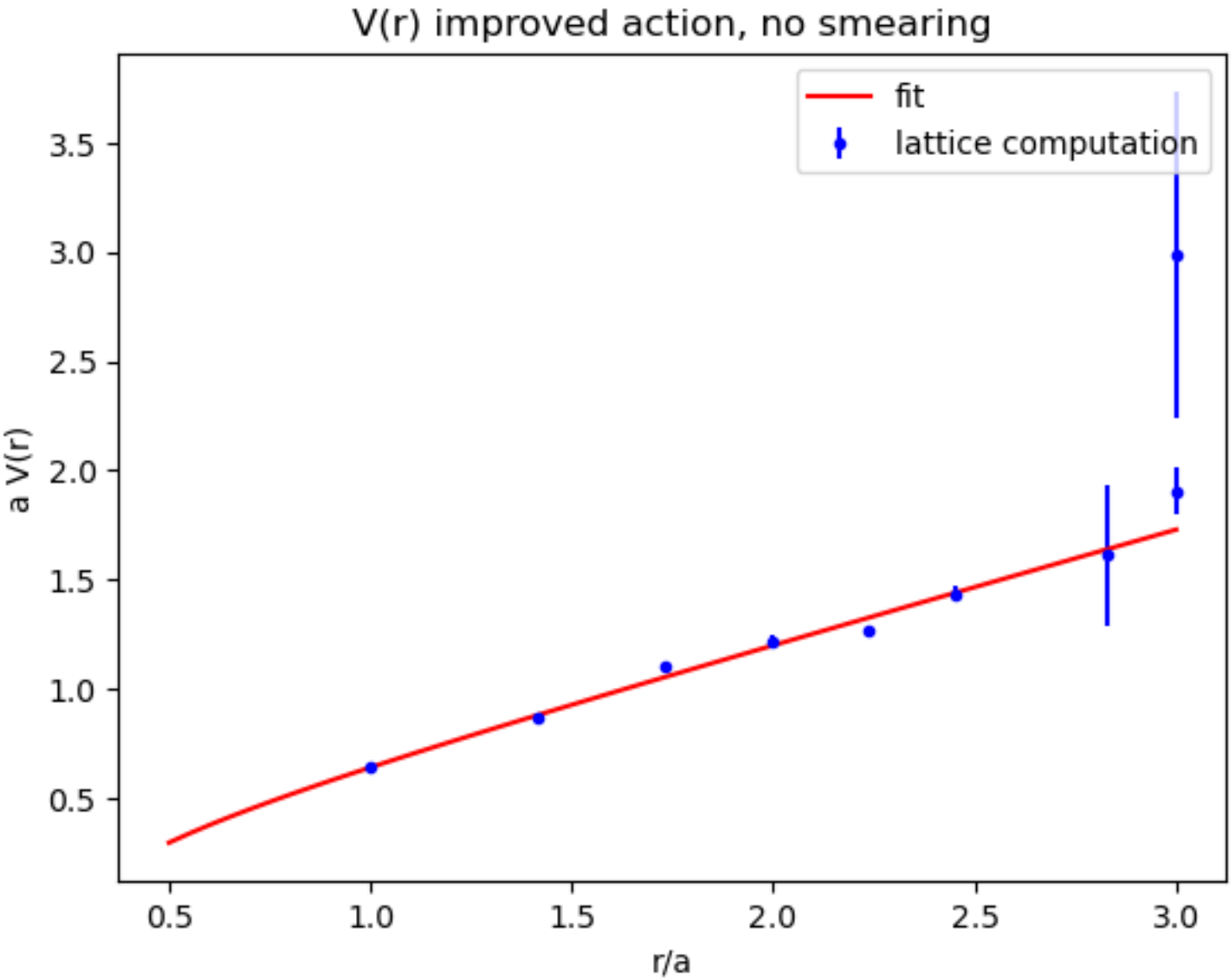
Fit function:

$$V(r) = ar - \frac{b}{r} + c$$

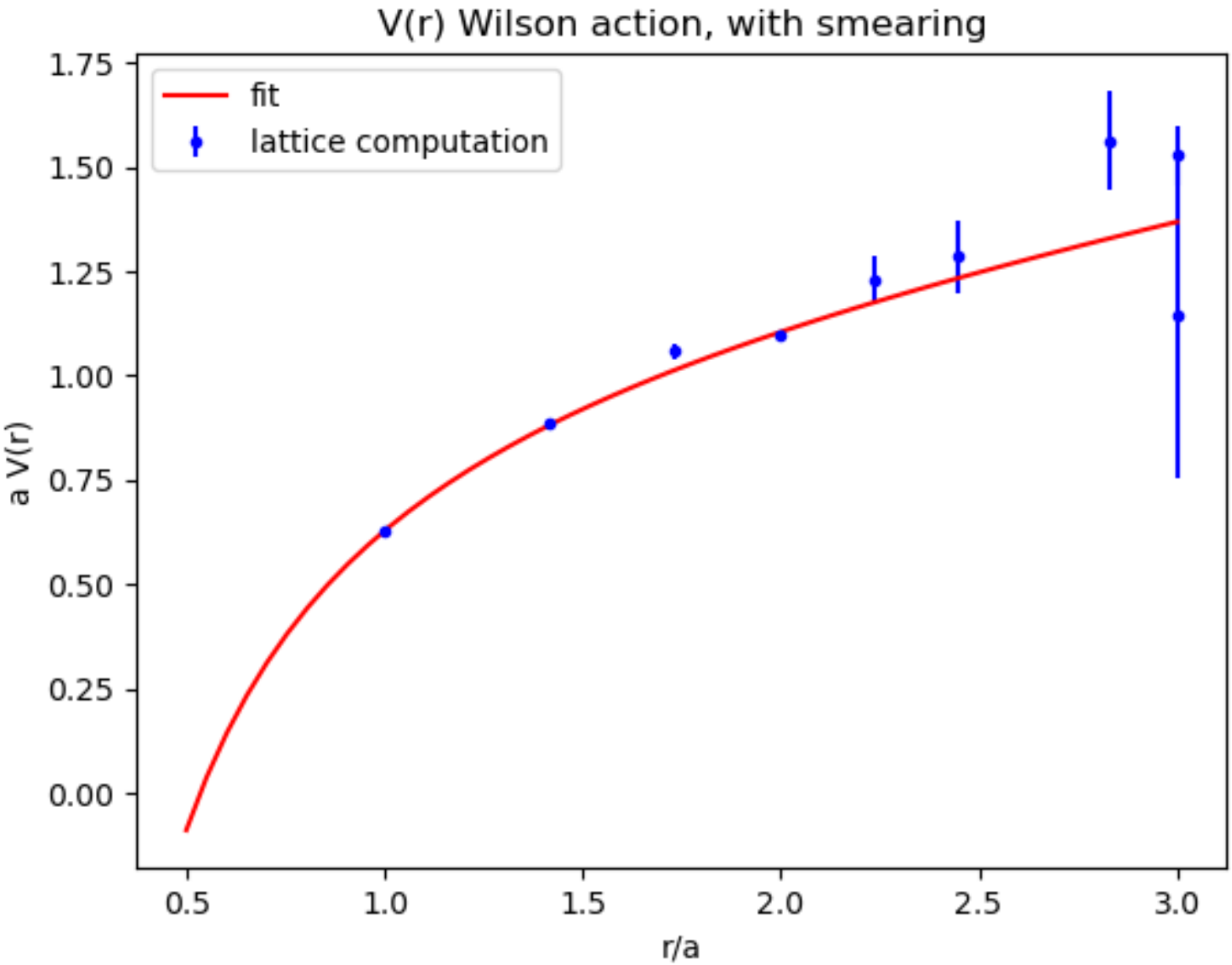
Potential: Wilson action, no smearing



Potential: Wilson action, with smearing



Potential: improved action, no smearing



Potential: improved action, with smearing