Instantons and MonteCarlo Methods in Quantum Mechanics

Alex Lombardi
Aldo Perri

Department of Physics and Astronomy University of Bologna

July 2023

Table of Contents

Project Objectives

This project is based on T. Schaefer's "*Instantons and Monte Carlo Methods in Quantum Mechanics*". The objective was to write in Python lenguage a series of codes that take advantage of Monte Carlo methods in order to understand the role of tunneling events in a quantum mechanical toy model (anharmonic potential). The main steps were:

- Exact diagonalization of the Hamiltonian problem
- Solution of the path integral formulation of the problem on a Euclidian lattice with Monte Carlo techniques
- Adiabatic switching technique to compute the free energy
- Extraction of instanton and anti-instanton content using cooling method
- Non-Gaussian corrections to the instanton density using adiabatic switching
- Analysis of a random instanton ensamble also with Gaussian fluctuations
- Consideration about the effect of interaction between instantons

Why Python?

The originale code written by Schaefer in 1996 to study this physical toy model was *fortran77*:

- excellent computational power
- codes are really user-unfriendly
- old-aged language.

For these reasons we decided to use *python* to build our project:

- high-level language, really user-friendly and easy to understand
- loss of computational power (much slower respect the fortran ones).

To avoid this last problem we applied *numba*, which consent us to

- maintain the python language simplicity
- improve the performance of our codes ($\sim$ fortran).

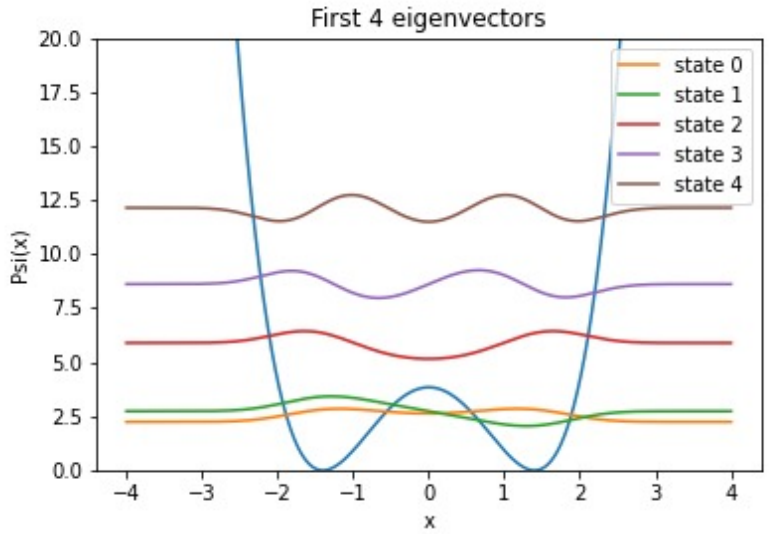
In this way, we were able to write a much readable and user friendly code, without any loss in the code performance.

Introduction

- **Instantons**: classical solution to equations of motion with a finite, non-zero action. They can be used to study the tunneling behavior.
- **Double weel potential**:

$$H = \frac{p^2}{2m} + \lambda(x^2 - \eta^2)^2, \qquad 2m = \lambda = 1.$$

Classically, there are two degenerate minima at $x = \pm\eta$. Quantum mechanically, the two states can mix:
$\psi_{0,1}(x) = \frac{1}{\sqrt{2}}(\psi_+(x) \pm \psi_-(x))$.



Exact Diagonalization

The Schroedinger eq.$(-\frac{\hbar^2}{2m}\frac{d^2}{dx^2} + V(x))\psi(x) = E\psi(x)$ is solved by the **"Finite Difference Matrix method"**. We start discretiziting the eq.:

- $-\frac{\hbar^2}{2m}\frac{\psi_{j+1}-2\psi_j+\psi_{j-1}}{\Delta x^2} + V_j\psi_j = E\psi_j + \mathcal{O}(\Delta x^2)$,
  where $j = 1, \ldots, N-2$, $\psi_j = \psi(x_j)$, $\quad \Delta x = x_{j+1} - x_j$
- We rearrange the equation:
  $-k\psi_{j+1} + 2(k + V_j)\psi_j - k\psi_{j-1} = E\psi_j + \mathcal{O}(\Delta x^2)$ with $k = \frac{\hbar^2}{2m\Delta x^2}$
- This creates a system of linear eq.

$$\begin{cases} \text{j=1} \quad \text{-k}\psi_2 + (2k + V_1)\psi_1 - k\alpha = E\psi_1 \\ . \\ . \\ . \\ j = N-2 \quad -k\beta + (2k + V_{N-2})\psi_{N-2} - k\psi_{N-3} = E\psi_{N-2} \end{cases}$$

Exact Diagonalization

- moving , $k\alpha, k\beta$ on RHS we get a matrix equation:

$$\begin{bmatrix} 2k+V_1 & -k & 0 & \dots & 0 \\ -k & 2k+V_2 & -k & \dots & 0 \\ 0 & -k & 2k+V_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots \\ 0 & \dots & 0 & -k & 2k+V_{N-2} \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \vdots \\ \psi_{N-2} \end{bmatrix}$$

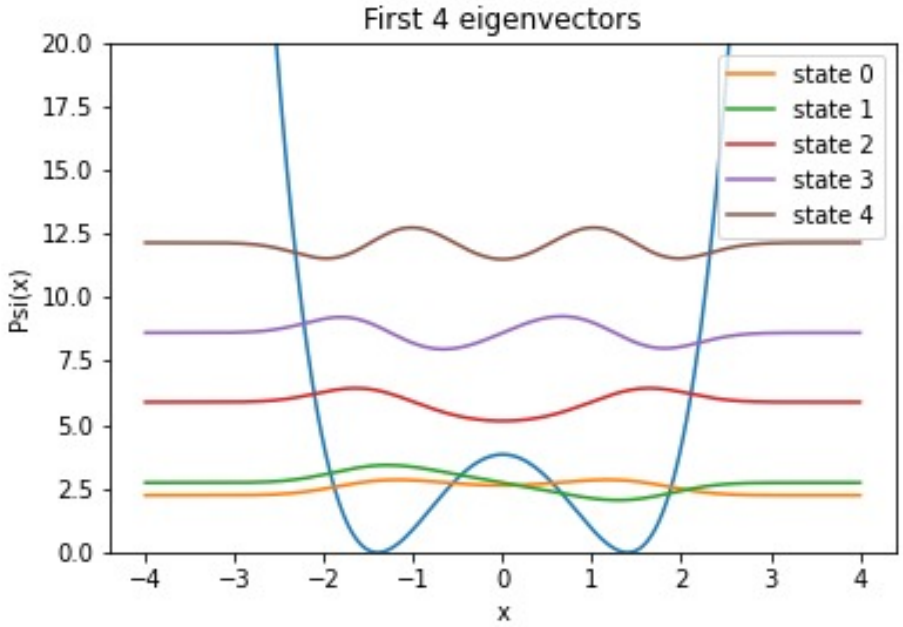
$$= E \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \vdots \\ \psi_{N-2} \end{bmatrix} + k \begin{bmatrix} \alpha \\ 0 \\ 0 \\ \vdots \\ \beta \end{bmatrix} \quad \text{or} \quad H\psi = E\psi + k\mathbf{b} + \mathcal{O}(\Delta x^2)$$

with b costant vector s.t. $b_1 = \psi_0 = \alpha, b_{N-2} = \psi_{N-1} = \beta$

- The solution is $\psi = k(H - EI)^{-1}\mathbf{b} + \mathcal{O}(\Delta x^2)$. $(H - EI)$ is a **symmetric tridiagonal matrix**, so the ψ solution can be found using tridiagonal linear solver (es. *np.linalg.eig()*)

Exact Diagonalization

H conserve parity, so we can decompose the matrix $H_{ij} = \langle i|H|j\rangle$ into even and odd components $H = H_{even} + H_{odd}$, such that their eigenvalues have positive and negative parity respectivly.
All this procedure is computed in *qmdiag.py*.

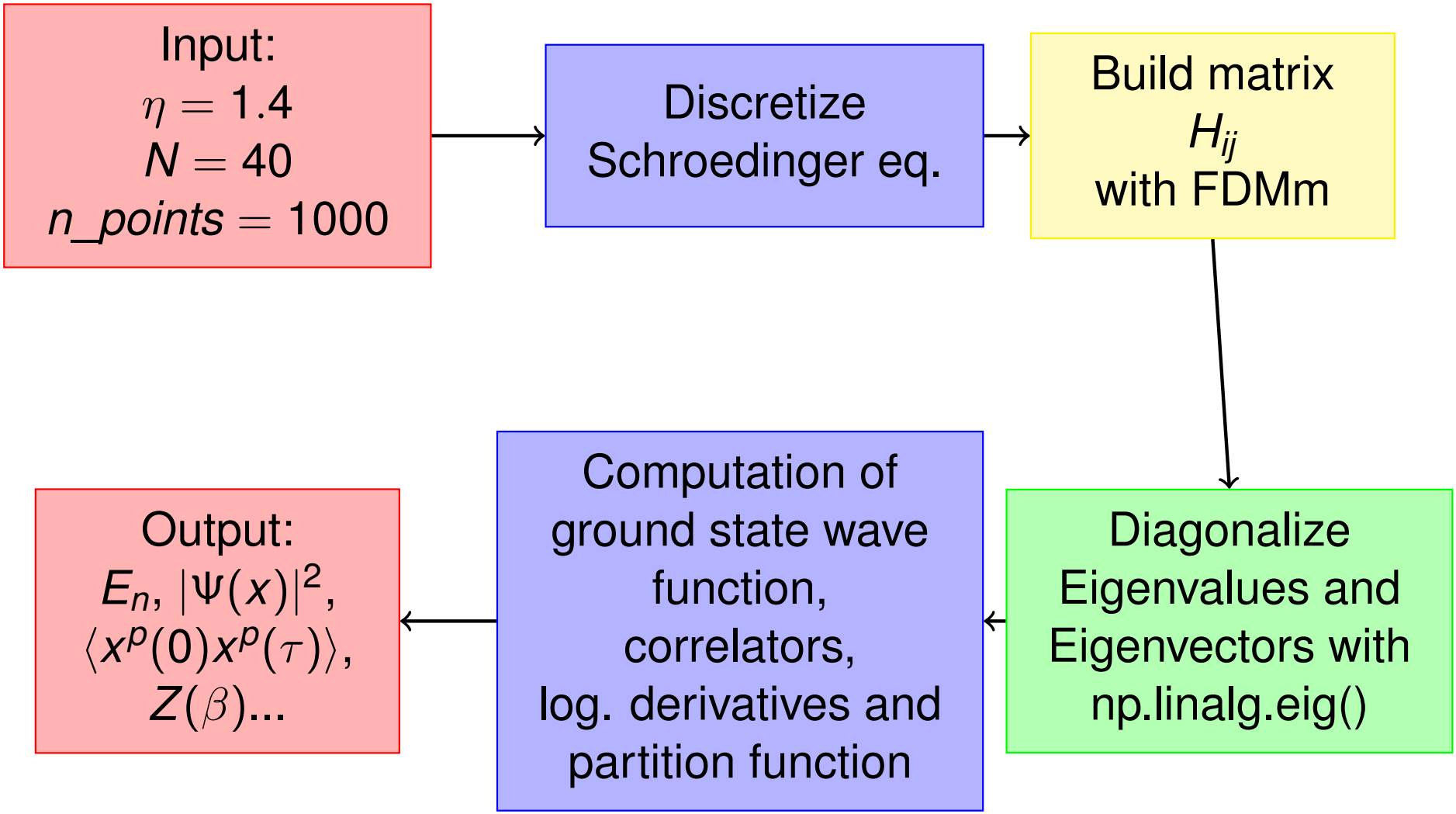


qmdiag.py

qmdiag.py computes the spectrum and the eigenfunctions of the anharmonic oscillator. The results are used in order to compute euclidean correlation functions.

- **Inputs**:
  - η: minimum of the anahrmonic potential $V = (x^2 - \eta^2)^2$
  - N: dimension of basis choose for diagonalizing H ($N \geq 40$)
  - *n_points*: number of points choosed for the Schroedinger's equation discretization (*n_points* $\sim$ 1000)
- Output:
  - E_n: Hamiltonian eigenvalues
  - $\psi(x)$: ground state wavefunction
  - $\langle x(0)x(\tau)\rangle$: Euclidian correlation function (also for x^2, x^3)
  - $dlog(\Pi)/d\tau$: log derivative of $\Pi = \langle x(0)x(\tau)\rangle$ (also for x^2, x^3)
  - $Z(\beta)$: partition function

qmdiag.py: algorithm



qmdiag.py: results

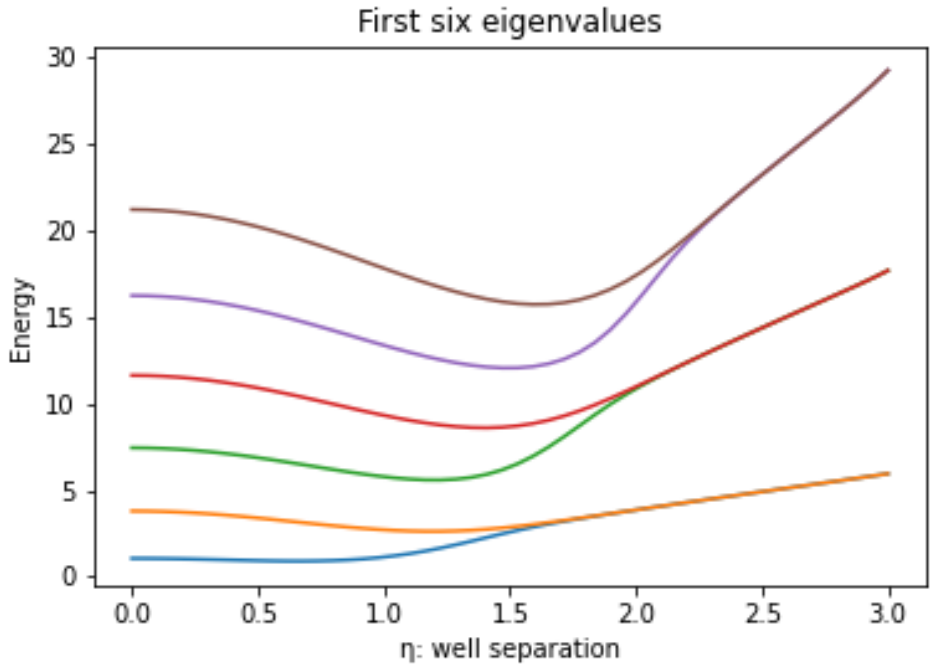


Figure: Spectrum of the double well potential as a function of the parameter η.

Positive and negative parity states become degenerate when $\eta \to \infty$.
In this limit the eigenfunctions are of the form:

$$\psi_{0,1}(x) = \frac{1}{\sqrt{2}}(\psi_+(x) \pm \psi_-(x)).$$

Feynman Path Integral formulation

The path integral for the anharmonic oscillator is given by

$$\langle x_1|e^{-iHt_1}|x_0\rangle = \int_{x_0}^{x_1} \mathcal{D}x e^{iS} \quad with \quad S = \int_0^{t_1} dt(\frac{1}{4}\dot{x}^4 - (x^2-\eta^2)^2)$$

With a Wick rotation to an Euclidian space-time we get the partition function:

$$Z(T) = \int \mathcal{D}x e^{-S_E} \quad with \quad S_E = \int_0^{\beta} d\tau(\frac{1}{4}\dot{x}^4 + (x^2-\eta^2)^2)$$

$\beta = T^{-1}$ is the inverse of the temperature, the partition function can be expressed in terms of the eigenvalues of the Hamiltonian

$$Z(T) = \Sigma_n(-E_n/T)$$

Numerical Simulations

Discretize the Euclidian action and the Euclidian time
$\tau_i = ia, \quad i = 1, \ldots, n$

$$S = \Sigma_{i=1}^{n} \{ \frac{1}{4a} (x_i - x_{i-1})^2 + a(x_i^2 - \eta^2)^2 \} \quad with \quad x_i = x(\tau_i)$$

Periodic boundary conditions $x_0 = x_n$.
This statistical system can be studied using standard Monte-Carlo sampling methods. In the following we will simply use the Metropolis algorithm.

Numerical Simulations: Metropolis Algorithm

- The Metropolis method generates an ensemble of configurations $x_i^{(k)}$ where $i = 1, \cdots, n$ are the lattice points and $k = 1, \cdots, N_{conf}$ the configurations

$$x_i^{(k+1)} = x_i^{(k)} + \delta x,$$

  acceptance rate: $P(x_i^{(k)} \to x_i^{(k+1)}) = min\{exp(-\Delta S), 1\}$
- Quantum mechanical averages: $\langle O \rangle = lim_{N_{conf} \to \infty} \frac{1}{N_{conf}} \Sigma_{k=1}^{N_{conf}} O^{(k)}$
- Fluctuations: $\Delta \langle O \rangle = \sqrt{\frac{\langle O^2 \rangle - \langle O \rangle^2}{N_{conf}}}$
- Euclidian Correlation function via spectral representation: $\Pi(\tau) = \langle O(0) O(\tau) \rangle = \Sigma_n |\langle 0|O(0)|n\rangle|^2 exp(-(E_n - E_0)\tau)$

Time Scale Behaviour

A typical path that appears in Monte Carlo simulations is governed by two time scales:

- **oscillation time** at short time scale: $\tau_{osc} \sim \omega^{-1} \sim (4\eta)^{-1}$
- **tunnelling time** for large time scale: $\tau_{tun} \sim e^{4\eta^3/3}$

In order to perform reliable simulation we make sure that:

$$a << \tau_{osc} \qquad \tau_{tun} << na$$

Then a typical choice for $\eta = 1.4$ is $nmc \sim 10^5$, $n = 800$ and $a = 0.05$.

qm.py

This programs computes correlation functions of the anharmonic oscillator using Monte Carlo simulations on a euclidean lattice.

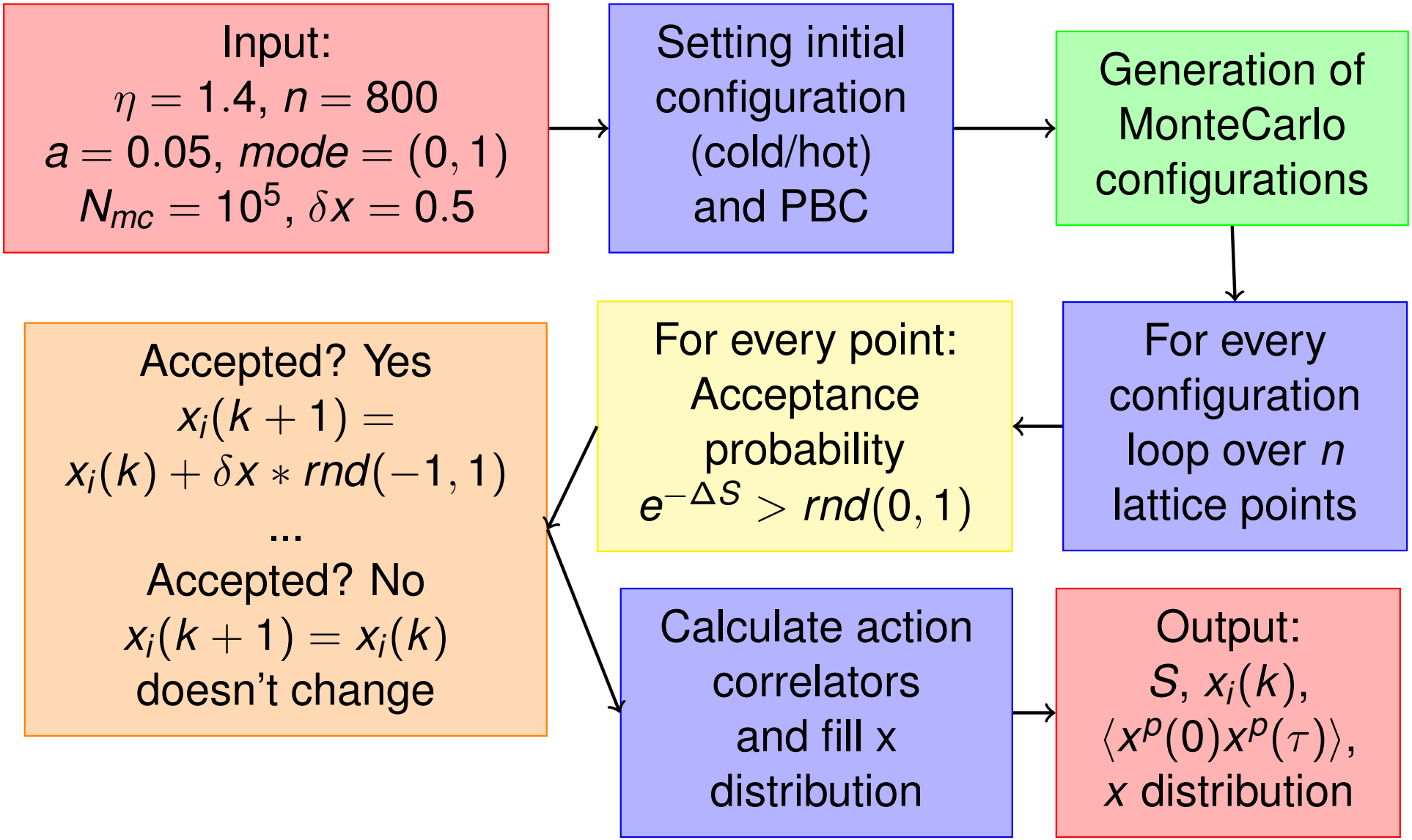
- **Inputs**:
  - n: number of points on Euclidian lattice ($n \sim 800$)
  - a: lattice spacing ($a \sim 0.05$)
  - mode: $0 \rightarrow$ "cold start" ($x_i = -\eta$), $1 \rightarrow$ "hot start" ($x_i = rnd()$)
  - n_{mc}: MonteCarlo configuration ($n_{mc} \sim 10^5$)
  - n_{eq}: number equilibration sweeps ($n_{eq} \sim 100$)
  - δx:width of Gaussian distribution used for MonteCarlo update ($\delta x \sim 0.5$)
  - n_p: number of points in correlation function($n_p = 20$)
  - n_{mea}: number of measurements of the correlation function in a given MonteCarlo configuration x_i ($n_{mea} \sim 5$)

qm.py

This programs computes correlation functions of the anharmonic oscillator using Monte Carlo simulations on a euclidean lattice.

- **Outputs**:
  - S_{tot}: average total action per configuration
  - T_{tot}, V_{tot}: average total kinetic and potential energy per configuration
  - $\langle x^n \rangle$: average for $n = 1, \cdots, 4$
  - $\Pi(\tau)$: euclidian correlation function $\langle O(0)O(\tau) \rangle$ with $O = x, x^2, x^3$

qm.py: algorithm



qm.py: results

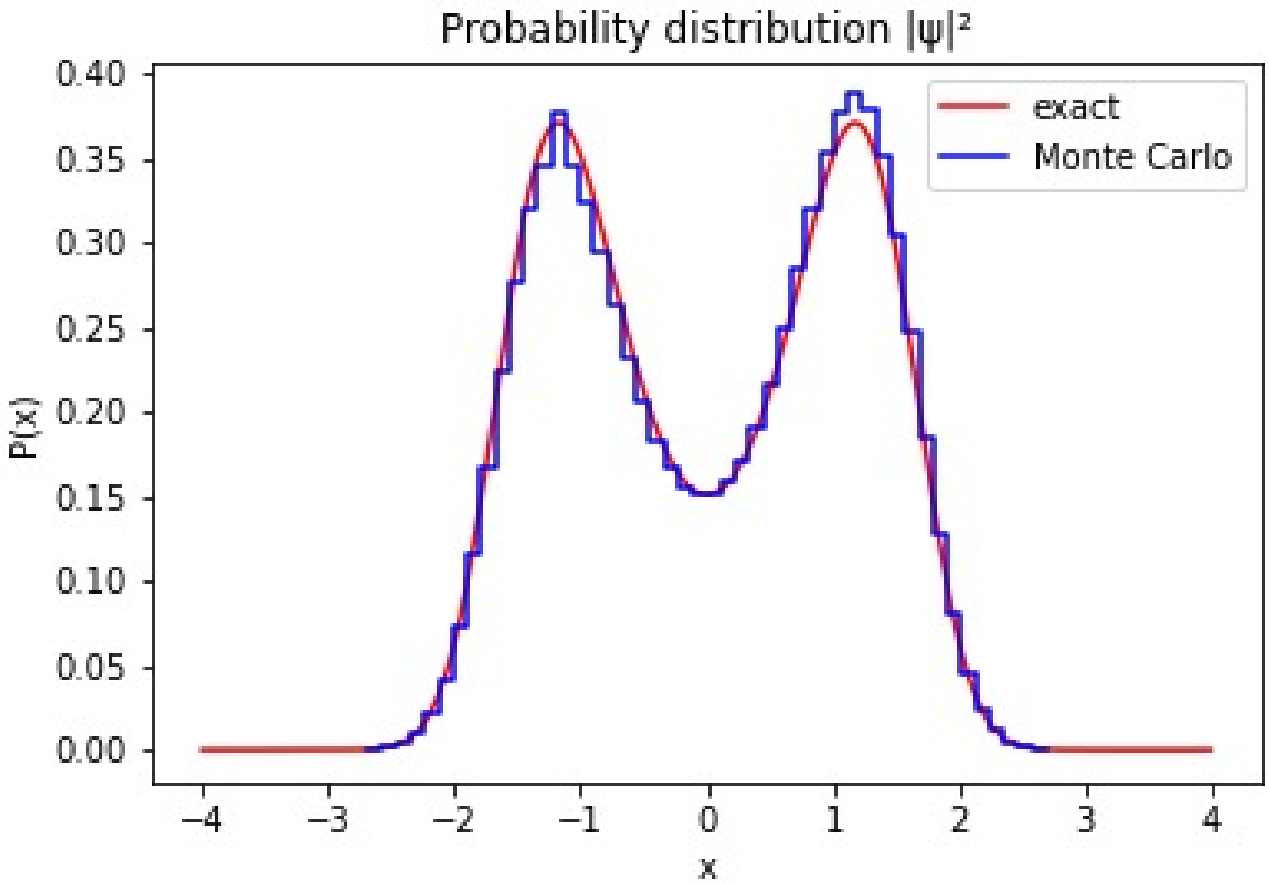


Figure: Probability distribution $|\psi(x)|^2$ in the double well potential for $\eta = 1.4$.

qm.py: results

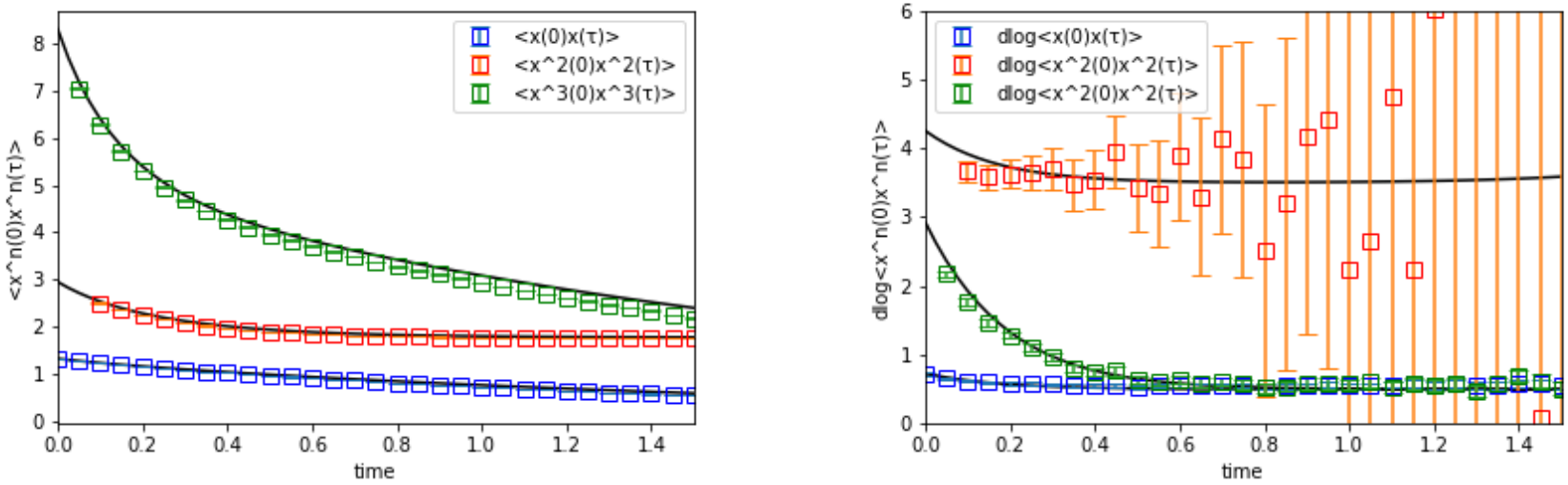


Figure: Correlation functions and their logarithmic derivative in the double well potential with $\eta = 1.4$ from Monte Carlo simulations

- For even powers of x the correlator has huge a constant term $|\langle 0|\, x^{2n}\, |0\rangle|^2$ which causes the big error bar in the graph.

Adiabatic Switching

The Monte Carlo method:

- very good in computing expectation values in quantum and statistichal mechanics
- doesn't provide directly the partition function or the free energy $\rightarrow$**adiabatic switching technique** .
  - Start from a reference system and compute its action S_0
  - Vary the configuration through MonteCarlo techniques, compute the new action S and $S_\alpha = S_0 + \alpha \Delta S$, where $\Delta S = S - S_0$ and α is the coupling costant ($0 < \alpha < 1$)
  - Integrate the relation $\partial logZ(\alpha)/(\partial\alpha) = -\langle \Delta S \rangle$ and find

$$logZ(\alpha = 1) = logZ(\alpha = 0) - \int_0^1 d\alpha' \, \langle \Delta S \rangle_{\alpha'}$$

Adiabatic Switching

In the present case we have the harmonic oscillator basis as reference, so we would find:

$$Z(0) = \Sigma_n exp(-\beta E_n^0) = \frac{exp(-\beta\omega_0/2)}{1 - exp(-\beta\omega_0)}$$

Note: the free energy of the anharmonic oscillator should be independent of ω_0.

- The integral over the coupling constant α can easily be calculated in Monte Carlo simulations by slowly changing α from 0 to 1 during the simulation
- In order to estimate systematic errors due to incomplete equilibration it is useful to repeat the calculation with α changing from 1 to 0 and study possible hysteresis effects

qm_switch.py

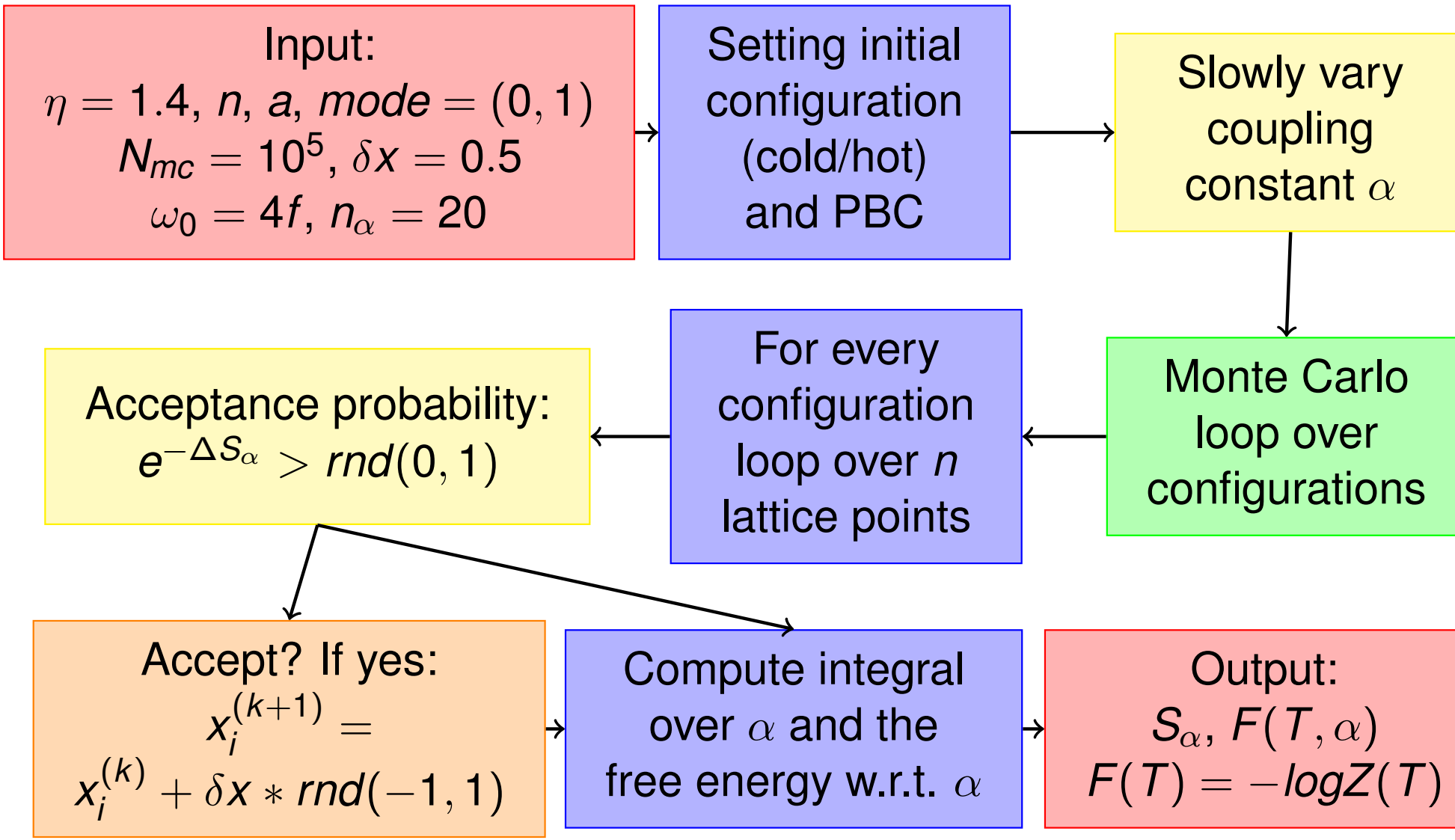
The program computes the free energy $F = -Tlog(Z)$ of the anharmonic oscillator using the method of adiabatic switching between the harmonic and the anharmonic oscillator.

- Action: $S_\alpha = S_0 + \alpha(S - S_0)$
- α vary from 0 *to* 1 for the adiabatic switch, and back from 1 *to* 0 for the hysteresis effects ($n_\alpha = 20$ adiabatic switch steps)
- hysteresis effects are used in order to estimate errors from incomplete equilibration

The uncertainties in the final result are estimated as
$\Delta F = \Delta F_{stat} + \Delta F_{eq} + \Delta F_{disc}$

- ΔF_{stat} statistical error
- ΔF_{eq} error due to incomplete equilibration (hysteresis)
- ΔF_{disc} error due to α discretization

qm_switch: algorithm



qm_switch: results

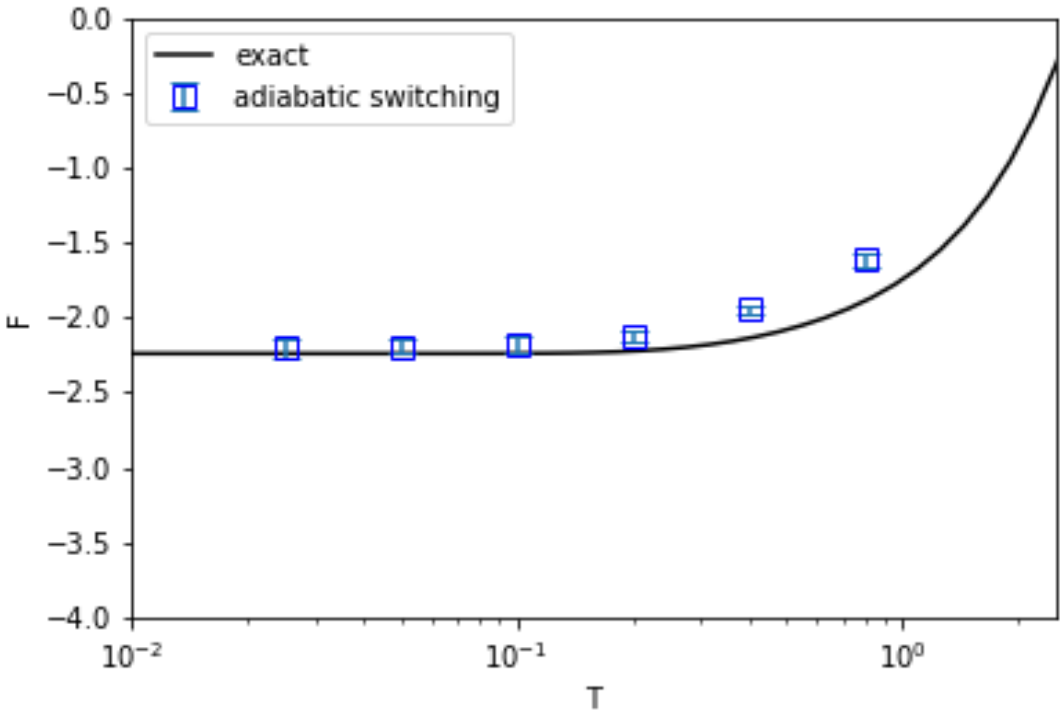


Figure: Free energy of the anharmonic oscillator as a function of the temperature T. The solid line is calculated using the Hamiltonian spectrum. This graph is obtained varying T in *qmswitch_loop.py*

$$F = -Tln(Z), \qquad T = 1/\beta, \qquad \beta = na$$

Instatons Solutions

- instanton solution can be found from the classical equation of motion

$$\frac{\partial S_E}{\partial x(\tau)} = 0 \rightarrow m\ddot{x} = V'(x)$$

with a solution which satisfies the boundaries $x(\tau \rightarrow \pm\infty) = \pm\eta$:

$$x_I(\tau) = \eta tanh[\frac{\omega}{2}(\tau - \tau_0)]$$

- anti-instanton solution: $x_A(\tau) = -x_I(\tau)$
- classical action for instantons: $S_0 = \frac{4\eta^3}{3}$
- tunneling rate: $n_{I+A} = N_{I+A}/\beta \sim exp(-S_0)$ exponentially small

In order to determine the pre-factor one has to study small fluctuations around the instanton solution. In a NLO semiclassical expansion we find:

$$n_{I+A} = 8\eta^{\frac{5}{2}}\sqrt{\frac{2}{\pi}}e^{-S_0-\frac{71}{72}\frac{1}{S_0}}$$

Cooling

The tunneling events can be studied in more detail after removing short distance fluctuations. **Cooling**:

- Metropolis algorithm in which we pick only the new configurations such that $\Delta S < 0$
- drive the system towards the nearest classical solution and allow to study the instanton content
- we can extract the instanton and anti-instanton locations from the zero crossings in the Euclidian path
- we can consider the cooled configurations as a superposition of tunnelling events $\rightarrow$ "**sum ansatz**":

$$x_{sum}(\tau) = \eta\{\Sigma_i(Q_i tanh[\frac{\omega}{2}(\tau - \tau_i)])\} \qquad Q_i = \pm 1$$

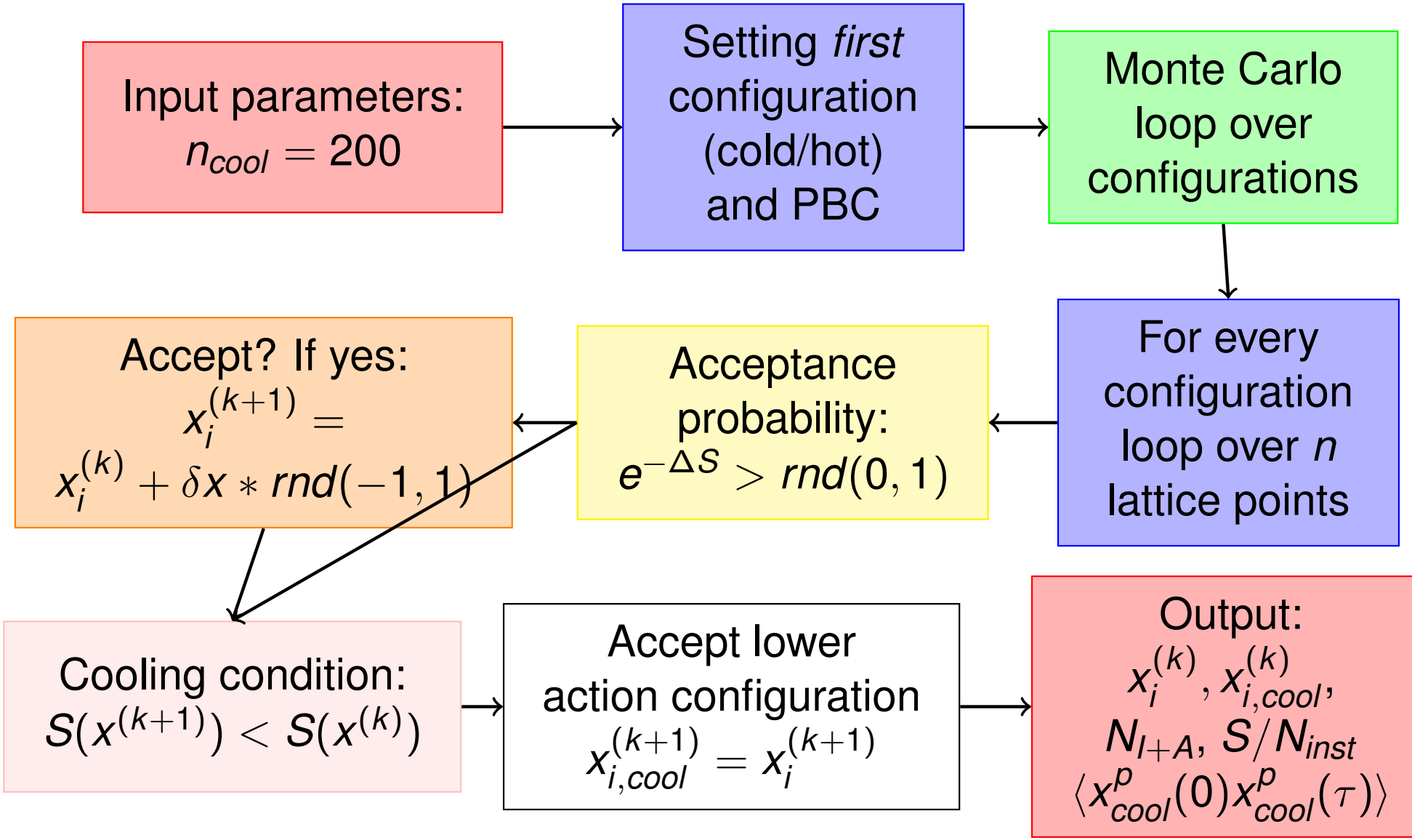
The "new" correlation functions are quite different to the "complete" ones. Short distance fluctuations eliminated by cooling play an important role.

qmcool.py

This programs is identical to qm.py except that expectation values are measured both in the original and in cooled configurations.

- **Inputs**:
  - n_{cool}: number of cooling sweeps ($n \sim 200$)
- **Output** :
  - $\Pi(\tau)$: correlation functions
  - N_{I+A}: total number of instantons extracted from number of zero crossings as a function of the number of cooling sweeps
  - S_{tot}: total action vs. number of cooling sweeps
  - S/N: action per instanton

qmcool.py: algorithm



qmcool.py: results

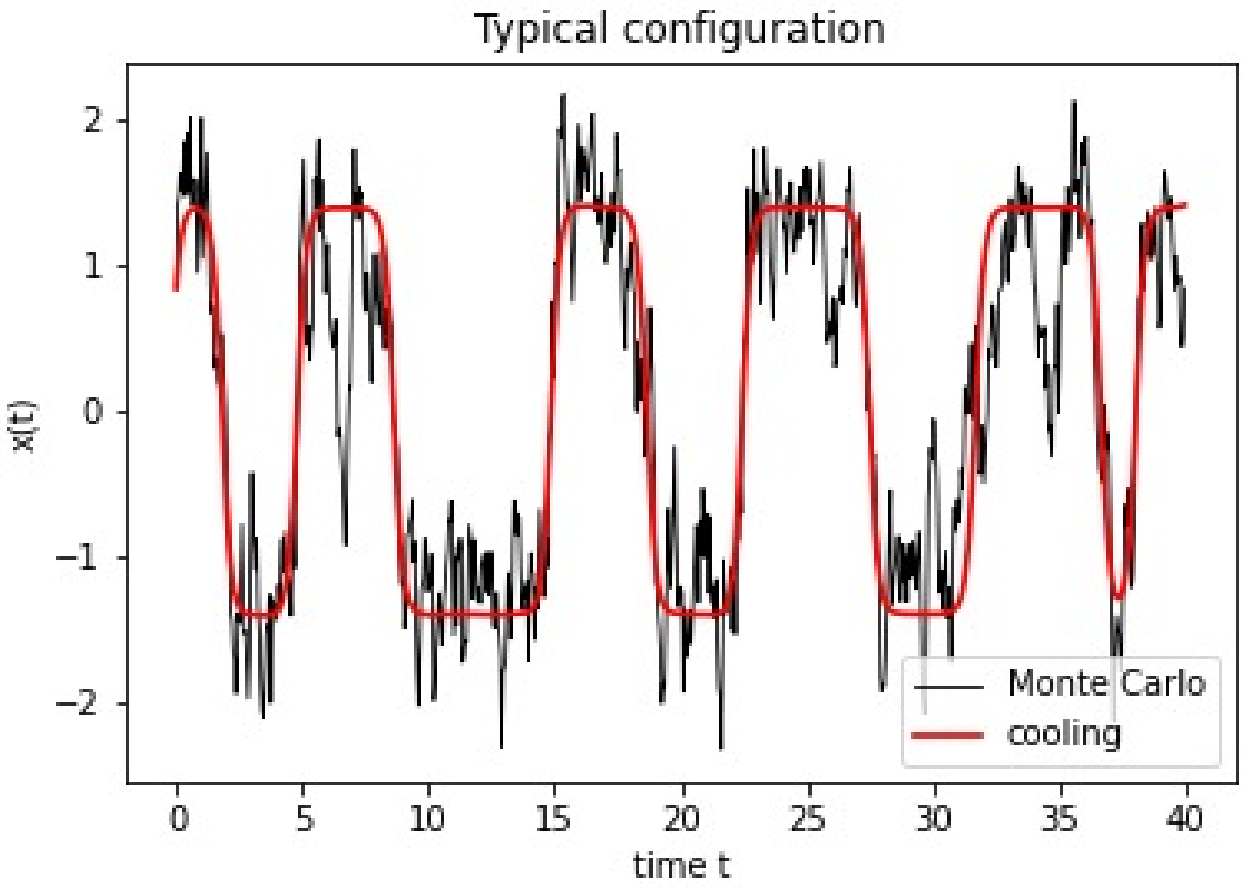


Figure: Typical euclidean path obtained in a Monte Carlo simulation of the discretized euclidean action of the double well potential for $\eta = 1.4$. The red curve shows the smooth path obtained by running 200 cooling sweeps on the original path.

qmcool.py: results

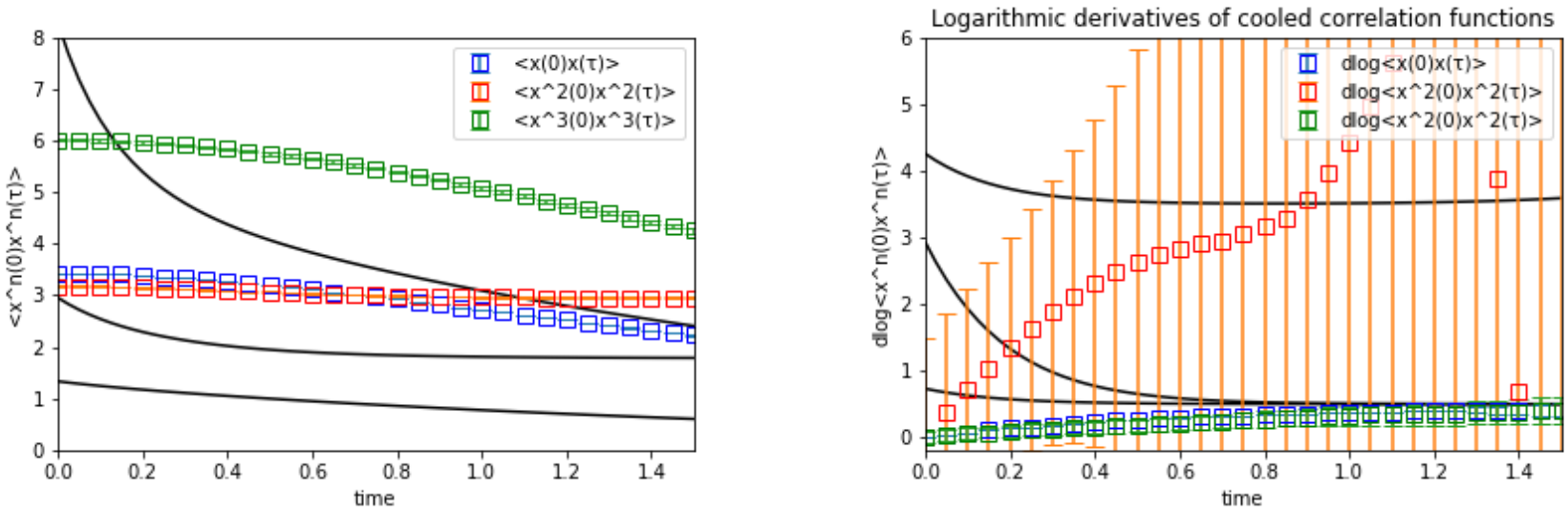


Figure: Correlation functions and their logarithmic derivative after 200 cooling sweeps

Observing the differences respect the non-cooled ones:

- The cooling has deleted short-distance fluctuations
- the level splitting between the ground state and the first excited state is clearly dominated by semiclassical configurations
- when τ increases the solution becomes the classical one

Cooling and instantons Density

Cooling method might be used to estimate the total density of instantons and anti-instantons:

- we cannot distinguish between a large quantum fluctuation and a very close instanton, anti-instanton pair $\rightarrow$ dependence on n_{cool}.
- net topological charge is well defined $= n_I - n_A$.
- Instanton density well defined in the semi-classical limit $\rightarrow$ exponentially large separation of scales between τ_{tun} and τ_{osc}. Expected number of instantons:

$$N_I(n_{cool}) = N_{osc} e^{-n_{cool} a/\tau_{osc}} + N_{tun} e^{-n_{cool} a/\tau_{tun}}$$

Cooling and instantons Density

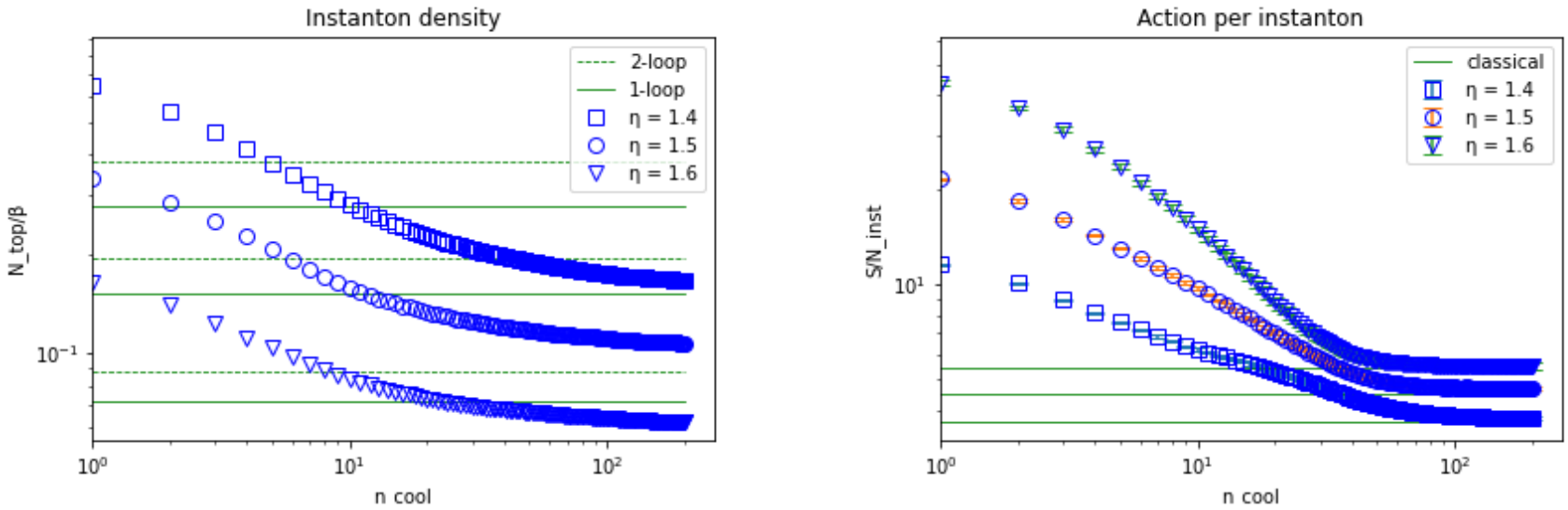


Figure: Instanton density and instanton action as a function of the number of cooling sweeps for $\eta = 1.4, 1.5, 1.6$. See *qmcool_loop.py*

Expected number of instantons:

$$N_I(n_{cool}) = N_{osc} e^{-n_{cool} a/\tau_{osc}} + N_{tun} e^{-n_{cool} a/\tau_{tun}}$$

instantons Density: non-gaussian corrections

- The one-loop calculation of the tunneling rate is based on expanding the action around the classical path to quadratic order

$$S = S_0 + \frac{1}{2}\int d\tau \delta x(\tau) \frac{\delta^2 S}{\delta x^2}|_{x_I(\tau)} \delta x(\tau) + ...$$

  where $\delta x(\tau) = x_I(\tau) - x(\tau)$
- introduce again S_α for the adiabatic switch procedure: $S_\alpha = S_{gauss} + \alpha \Delta S$, with $\Delta S = S - S_{gauss}$.
- Quantum weight of an instanton can be obtained integrating over α:

$$n = n_{gauss} exp\Big[- \int_0^1 d\alpha \big(\langle \Delta S \rangle_\alpha^{(1)} - \langle \Delta S \rangle_\alpha^{(0)} \big) \Big]$$

  where $\langle . \rangle_\alpha^{(n)}$ is an expectation value in the n-instanton sector at coupling α.

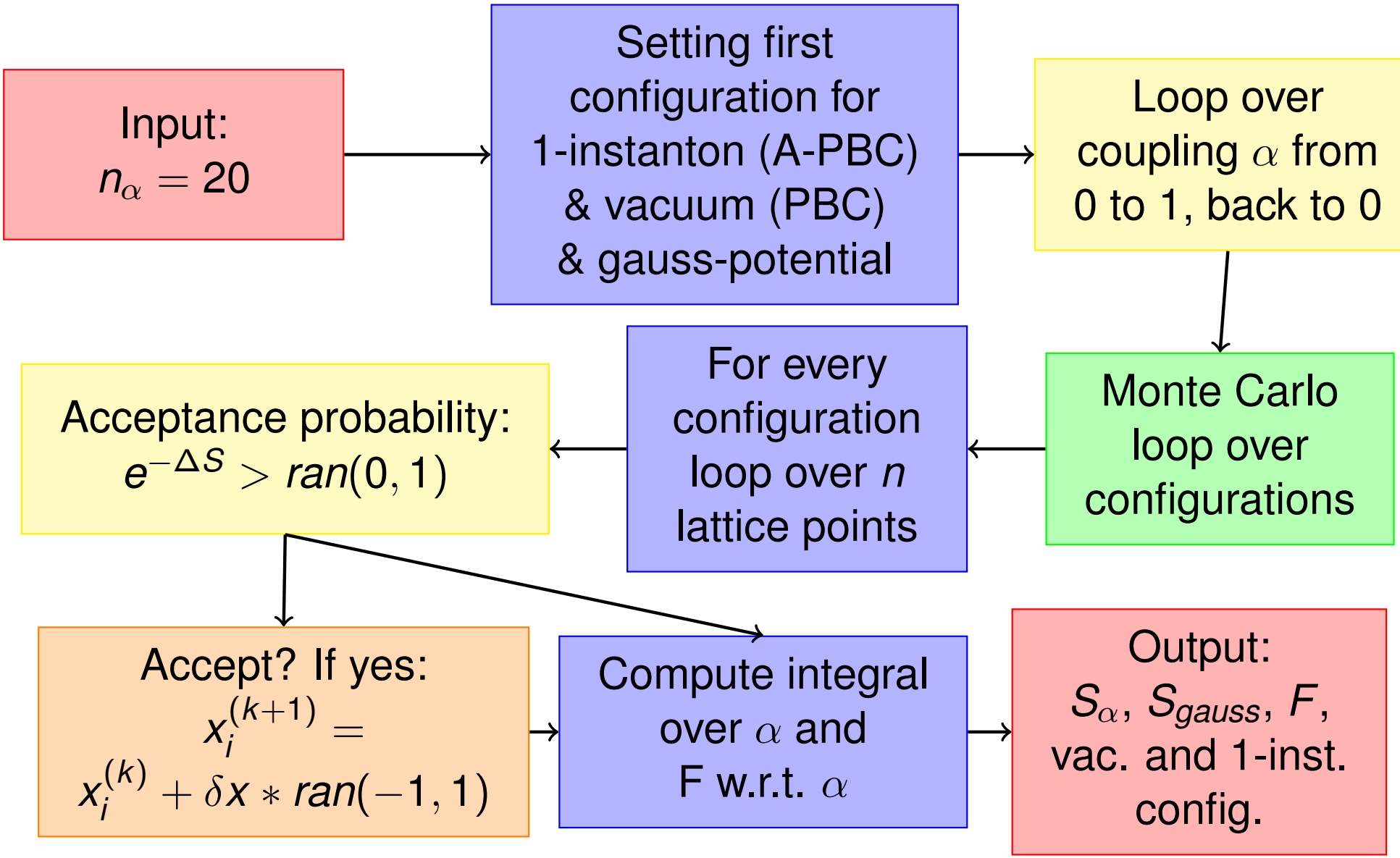
qmidens.py

The program `qmidens.py` calculates non-Gaussian corrections to the instanton density using adiabatic switching between the Gaussian action and the full action. The calculation is performed in both the zero and one-instanton-sector. The details of the adiabatic switching procedure are very similar to the method used in `qm_switch.py`.

- **Input**:
  - n_{switch}: number of adiabatic switchs ($n_{switch} \sim 20$)
- **Output**:
  - Zero-instanton sector configurations
  - One-instanton sector configurations
  - Details of the adiabatic switching procedure.

Note: the method breaks down if η is too small or β is too large.

qmidens.py: algorithm



qmidens.py: results

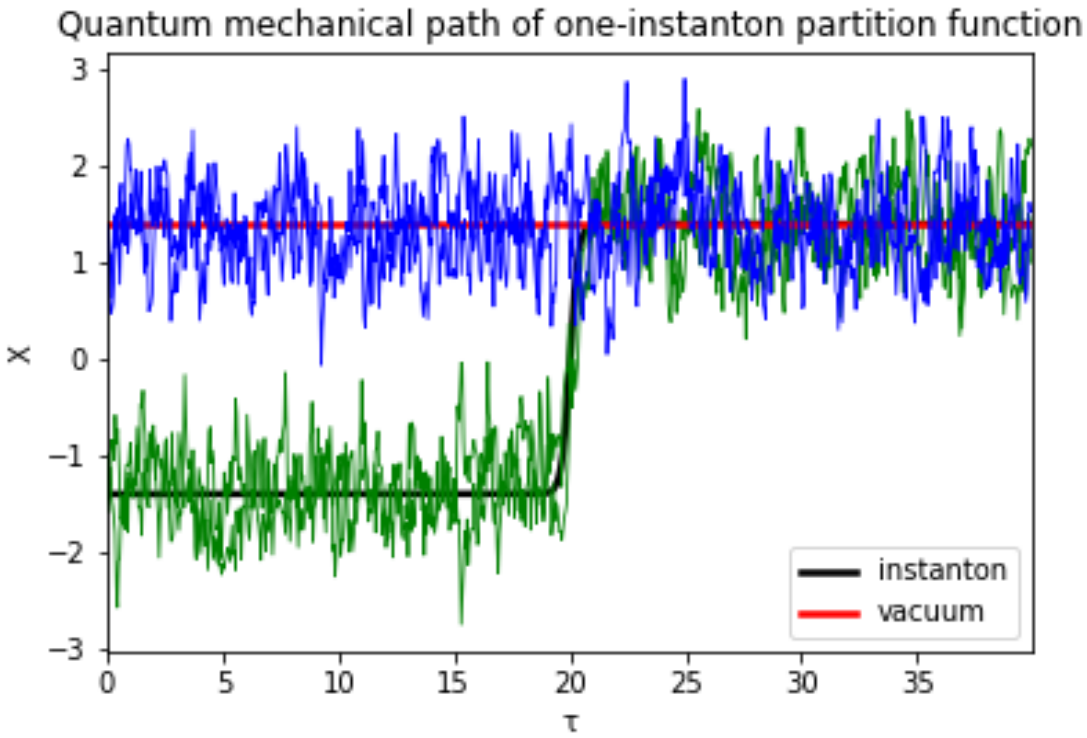


Figure: Quantum mechanical paths in Monte-Carlo calculation of the one-instanton partition function. The calculation involves adiabatic switching between the Gaussian effective potential and the full potential. The smooth curves are the initial configurations. The Monte Carlo updates in the one-instanton sector involve a constraint which keeps the instanton location fixed.

Cooling and instantons Density

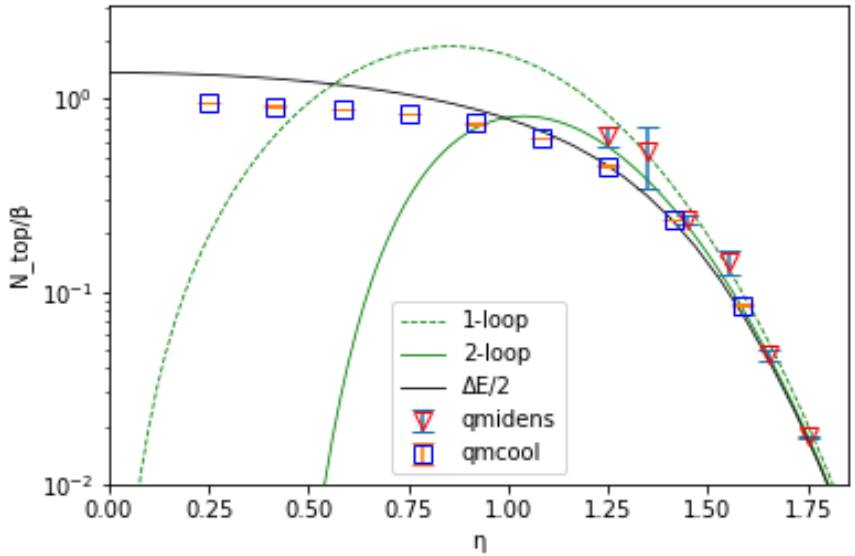


Figure: Instanton density as a function of η.The blue symbols show the instanton density extracted from qmcool.py. The red symbols show the results of non-Gaussian effects. See *qmcool_loop2.py*

- the one-loop result is given by: $n_{I+A} = 8\eta^{5/2}\sqrt{\frac{2}{\pi}}exp(-S_0)$
- the 2-loop result is given by: $n_{I+A} = 8\eta^{5/2}\sqrt{\frac{2}{\pi}}exp(-S_0 - \frac{71}{72}\frac{1}{S_0})$
- $\Delta E/2 = E_1 - E_0$ was computed by `qmdiag.py`.

Instantons Gas Model

We want to study the correlation functions in the semi-classical approximation in more detail. Considering the contribution from the classical path only:

- Partition Function: $Z = \Sigma_{n_I,n_A} \frac{\delta_{n_I,n_A}}{n_I! n_A!} \left(\prod_i \int d\tau_i \right) exp(-S)$
- Coordinate correlation function: $\Pi_{cl}(\tau) = \langle x_{cl}(0) x_{cl}(\tau) \rangle$
  $\langle . \rangle \rightarrow$ ensemble average over collective coordinates τ_i.
- Ignoring interactions between instantons: $S = (n_I + n_A) S_0$
  $\rightarrow$**Instantons Gas Model** (or **Random ensemble of instantons**)
- we construct configurations as a superposition of tunnelling events
  $\rightarrow$ "**sum ansatz**":

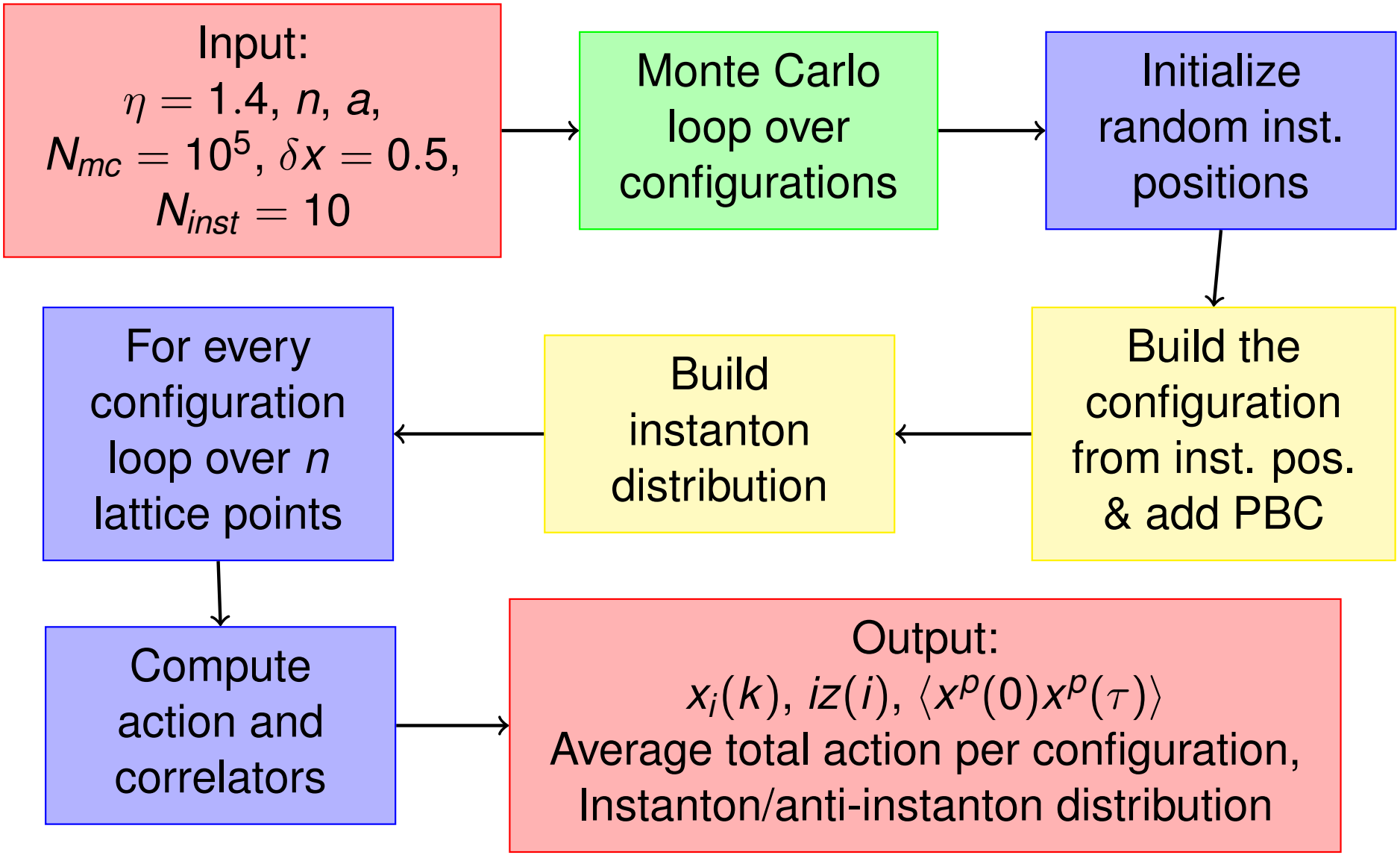
$$x_{sum}(\tau) = \eta\{\Sigma_i(Q_i tanh[\frac{\omega}{2}(\tau - \tau_i)])\} \qquad Q_i = \pm 1$$

rilm.py

This program computes correlation functions of the anharmonic oscillator using a random ensemble of instantons. The multi-instanton configuration is constructed using the sum ansatz. Note that, in contrast to RILM (Random Instantons Liquid Model) calculations in QCD, the fields and correlation functions are computed on a lattice.

- **Inputs**:
  - N_{inst}: number of instantons (has to be even, $N_{inst} \sim 10$).
    The program displays the one and two-loop result for the parameters $(\eta, \beta = na)$
  - $\eta, n, a, N_{nmc}, n_p, n_{mea}$
- **Output**:
  - S_{tot}: total action per configuration
  - S_{av}, V_{av}: average kinetic and potential energy
  - $\langle x^n \rangle$: expectation value $(n = 1, \cdots, 4)$
  - $\Pi(\tau)$: correlation functions$\langle O(0)O(\tau)\rangle$ with $O = x, x^2, x^3$ (also in format $\Delta\Pi(\tau), dlog(\Pi)/(d\tau), \Delta(dlog(\Pi)/(d\tau)))$

rilm.py: algorithm



rilm.py: results

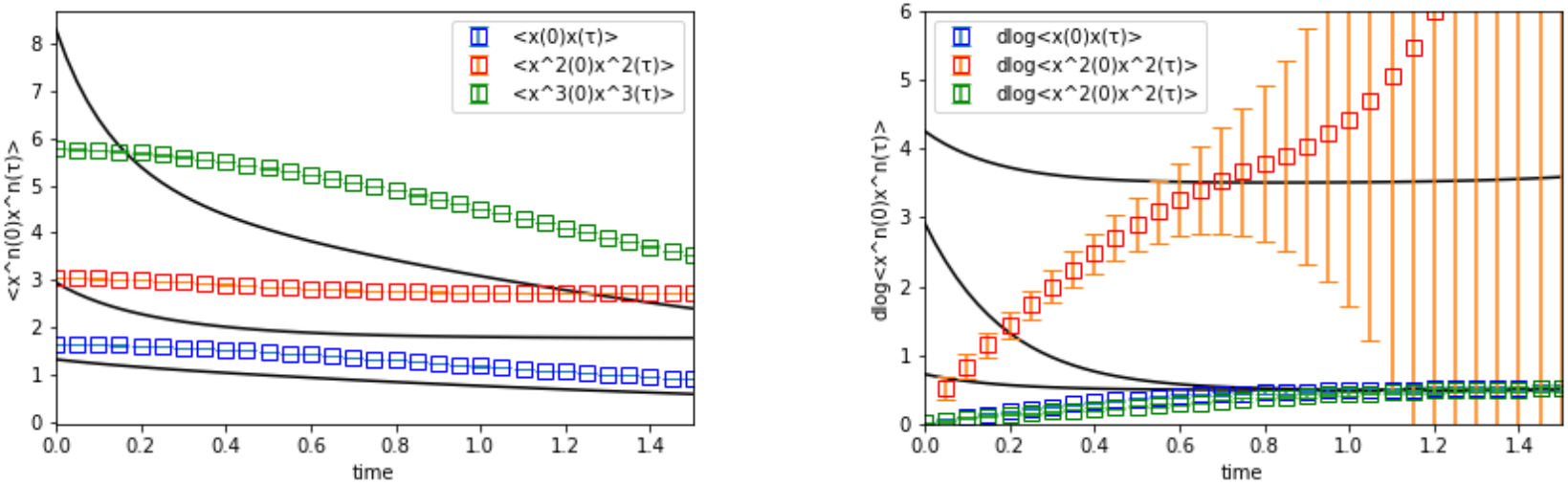


Figure: Correlation functions and logarithmic derivative correlation functions evaluated from a random instanton configuration.

- Correlation functions are similiar to the cooled result
- Splitting between ground state and first excited state is well reproduced, but other aspects of correlations are not.

Quantum Gaussian Fluctuations in RILM

The main feature that is missing from the ensemble of classical paths is quantum fluctuations. Quantum fluctuations appear at next order in the semi-classical approximation.

- quantum fluctuations determine the pre-exponential factor in the tunneling rate:

$$N_{inst} = 8\eta^{5/2}\sqrt{\frac{2}{\pi}}exp(-S_0 - \frac{71}{72}\frac{1}{S_0})$$

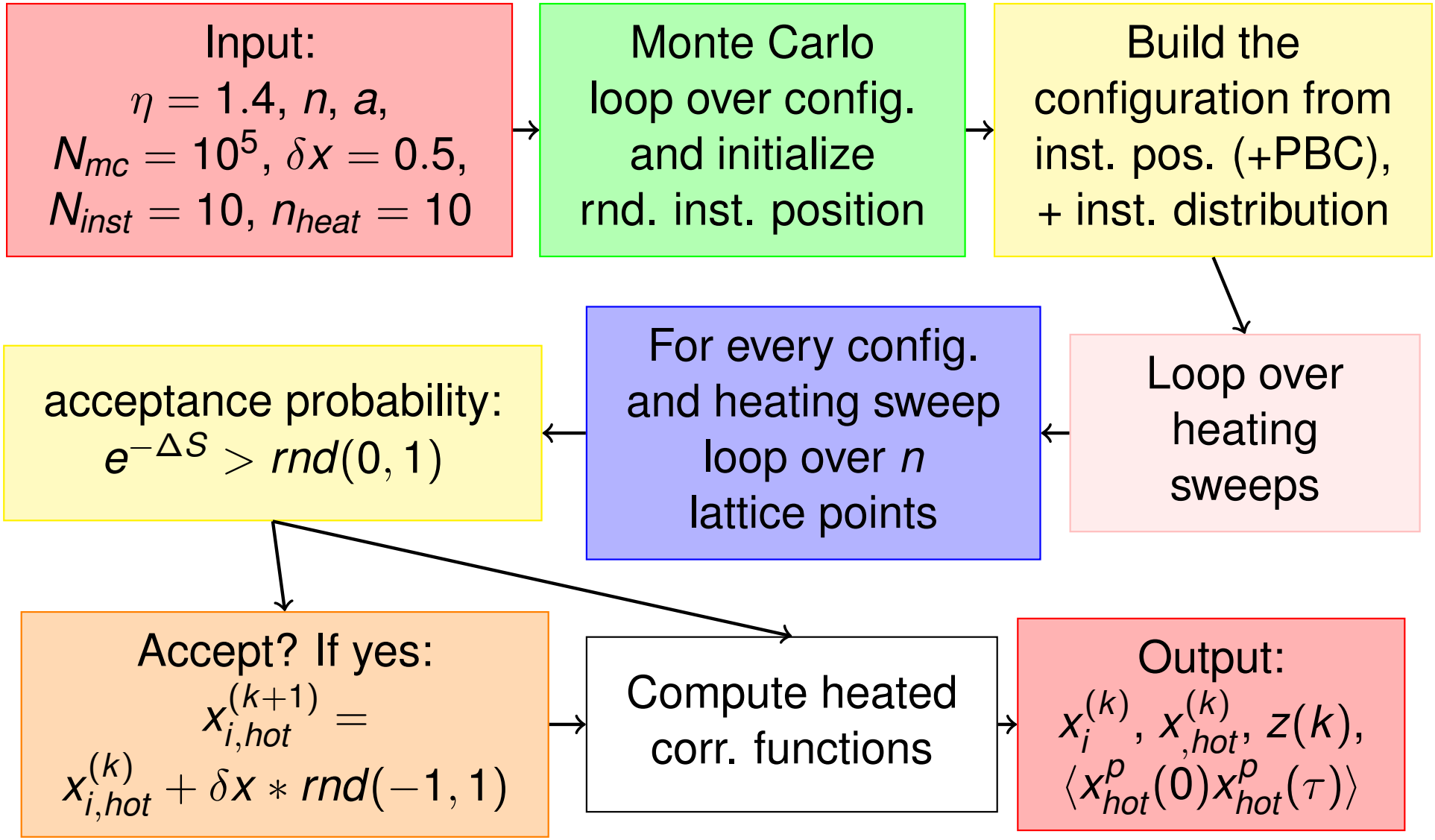
- we can write the path as $x(\tau) = x_{cl}(\tau) + \delta x(\tau)$ where $x_{cl}(\tau)$ is the classical path and $\delta x(\tau)$ is the fluctuating part
- For an ensemble of instantons we can approximate the propagator as a sum of contributions due to individual instantons, or numerically by the *heating* method (opposite of the cooling one).

rilmgauss.py

This program generates the same random instanton ensemble as `rilm.py` but it also includes Gaussian fluctuations around the classical path. This is done by performing a few heating sweeps in the Gaussian effective potential. Most input parameters are defined as in `rilm.py`. Additional input parameters are given below.

- **Input**:
  - n_{heat}: number of heating steps ($n_{heat} \sim 10$)
  - δx: coordinate update ($\delta x \sim 0.5$)
- **Output**:
  - S_{tot}: total action per configuration
  - S_{av}, V_{av}: average kinetic and potential energy
  - $\langle x^n \rangle$: expectation value ($n = 1, \cdots, 4$)
  - $\Pi(\tau)$: correlation functions$\langle O(0)O(\tau) \rangle$ with $O = x, x^2, x^3$ (also in format $\Delta\Pi(\tau), dlog(\Pi)/(d\tau), \Delta(dlog(\Pi)/(d\tau))$)

rilmgauss.py: algorithm



rilmgauss.py: results

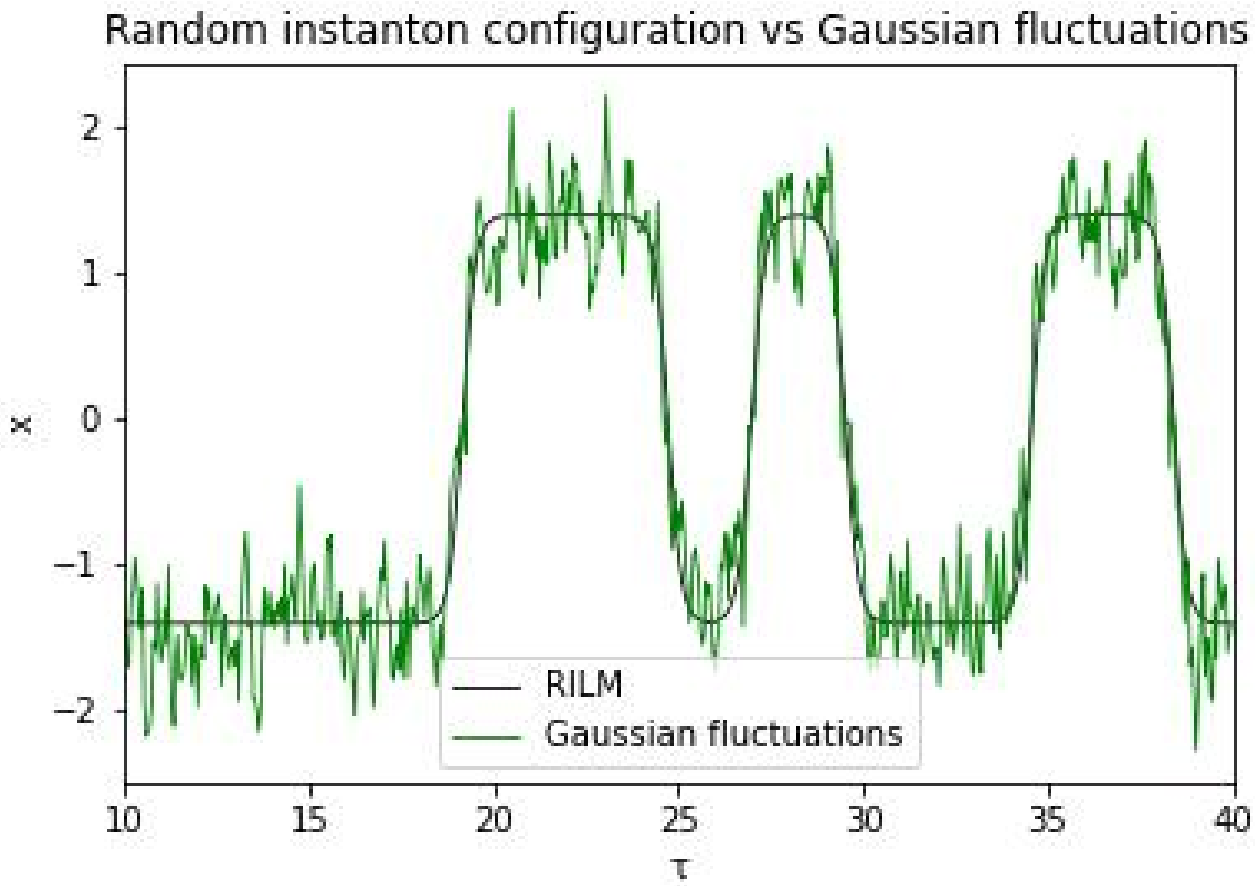


Figure: Typical random instanton configuration and the same configuration with Gaussian fluctuations. The noisy path generated using 10 heating sweeps in the Gaussian potential around the classical path.

rilmgauss.py: results

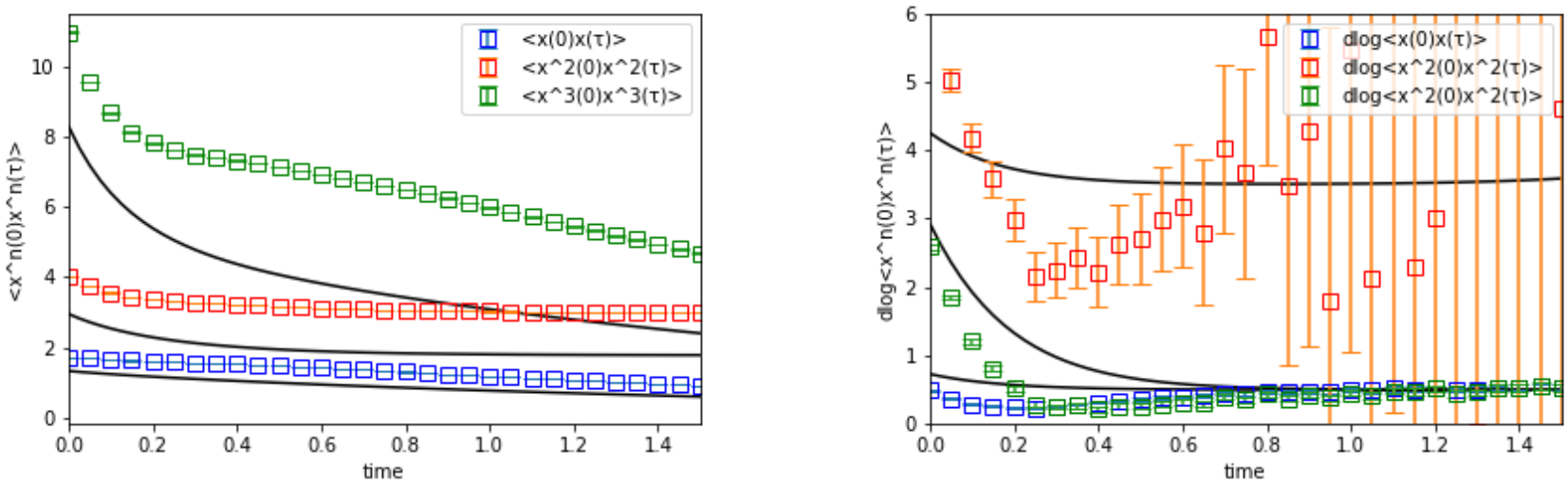


Figure: Correlation functions evaluated in a random instanton ensemble with Gaussian fluctuations.

The correlation functions are in much better agreement with the exact results than the correlators obtained from the classical path only. We also see that the correlators not only describe the splitting between the ground state and the first excited state but also provide a reasonable description of the second excited state.

Instantons Interactions

- Another feature that is missing from the random instanton ensemble is the correlation between tunneling events due to the interaction between instantons.
- The simplest method for studying the instanton-anti-instanton interaction is to construct a trial function and compute its action $\rightarrow$ **sum ansatz**
- Asymptotic Energy: $S_{IA} = 2S_0\,(1 - 6exp(-\eta\tau_{IA}) + ...)$ where $\tau_{IA} = |\tau_I - \tau_A|$ is instanton/anti-instanton separation
  - $\tau_{IA} \rightarrow \infty$ (no interaction) $S_{IA} = 2S_0$
  - $\tau_{IA} \rightarrow 0$ (strong interaction) they annihilate each other $S_{IA} \rightarrow 0$

In this limit the sum ansatz is just an approximate solution to the classical e.o.m, and it is not obvious how the path should be chosen.

Streamline Method

- In the space of all instanton/anti-instanton paths there is one almost flat direction along which the action slowly varies between $2S_0$ and 0. All other directions correspond to perturbative fluctuations
- To force the instanton-anti-instanton path to go along the almost flat direction, we add a Lagrange multiplier:

$$S_\xi = \xi(\lambda) \int d\tau \, [x(\tau) - x_\lambda(\tau)] \, \frac{dx_\lambda(\tau)}{d\lambda}$$

- Streamline configuration: start from a well separated I-A pair and let the system evolve following the principle of the steepest descendent:

$$\xi(\lambda) \frac{dx_\lambda(\tau)}{d\lambda} = \left. \frac{\delta S}{\delta x(\tau)} \right|_{x=x_\lambda}$$

with boundary condition $x_{\lambda=0}(\tau) \simeq x_{sum}$

streamline.py: results

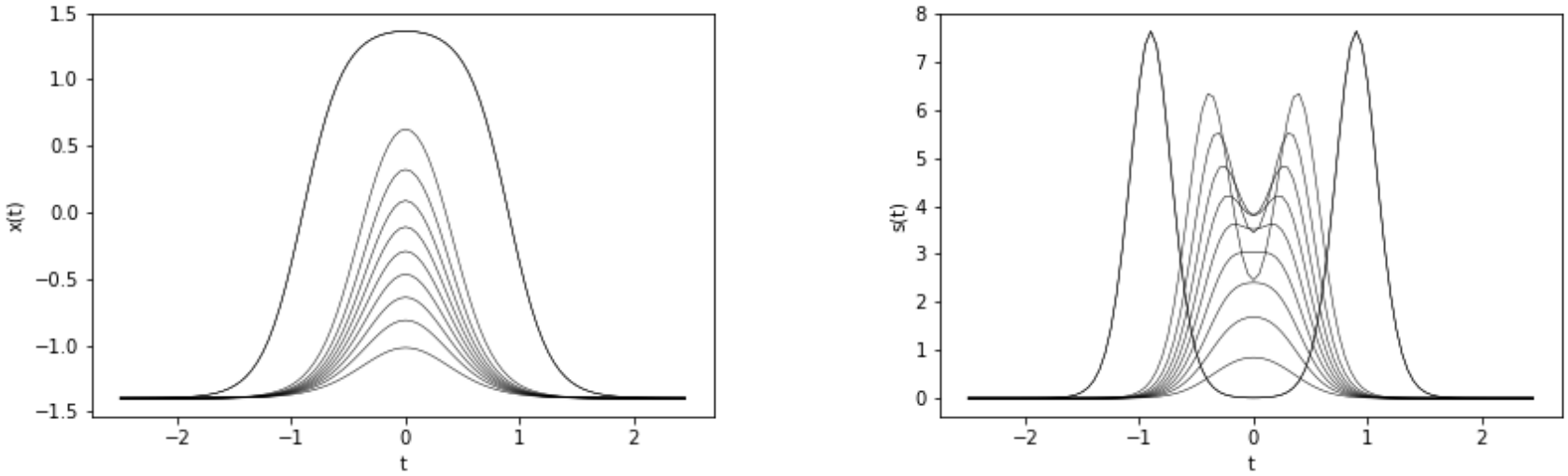


Figure: Solution of the streamline equation for an instanton-anti-instanton pair. Figure a shows the streamline path and the Fig. b shows the action density. The paths correspond to $S/S_0 = 2.0, 1.8, ..., 0.2, 0.1$.

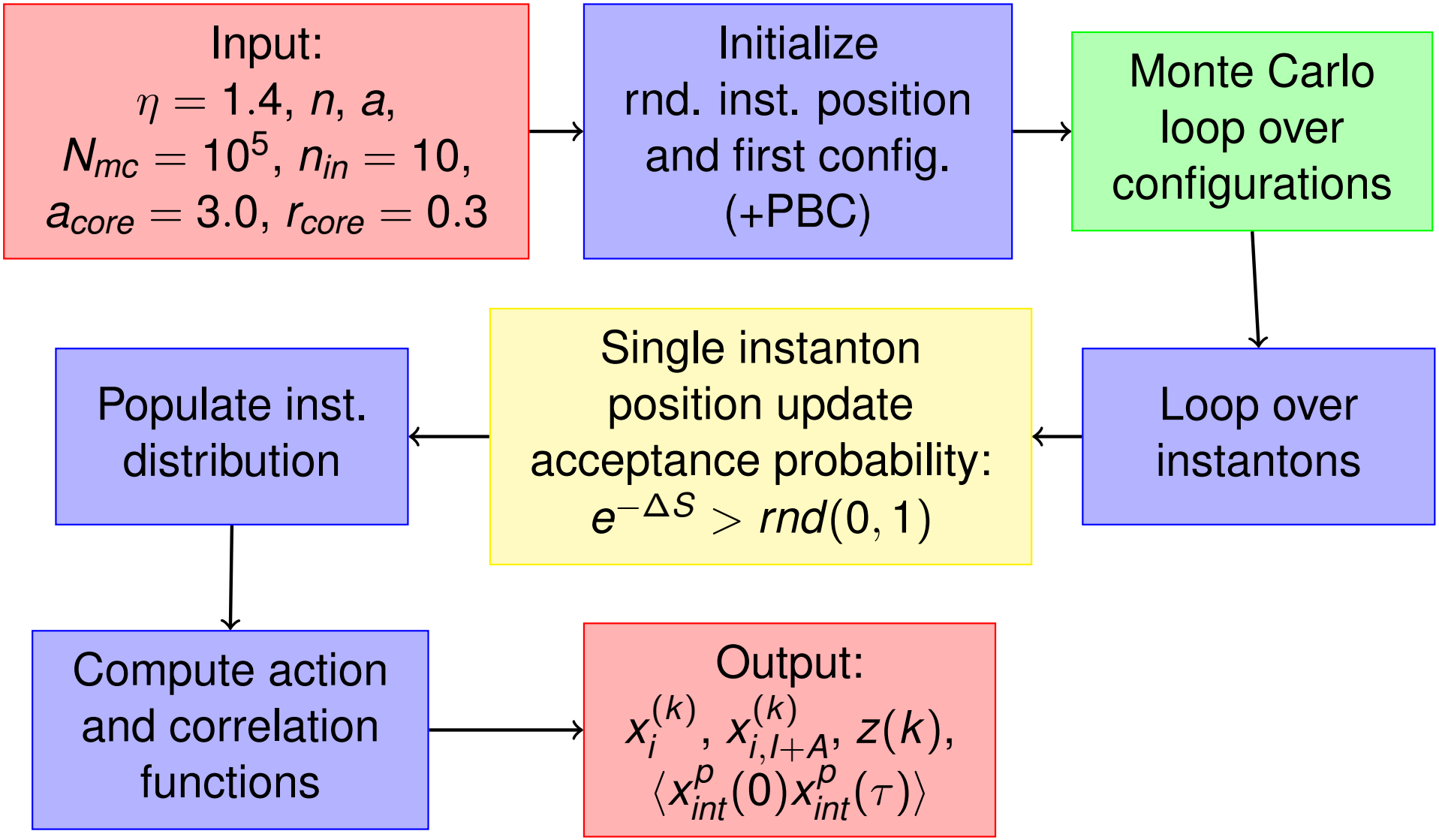
We can see clearly how the two localized solutions merge and eventually disappear as the configuration progresses down the valley.

iilm.py

This program computes correlation functions of the anharmonic oscillator using an interacting ensemble of instantons. The multi-instanton configuration is constructed using the sum ansatz. The configuration is discretized on a lattice and the total action is computed using the discretized lattice action. Very close instanton-anti-instanton pairs are excluded by adding an nearest neighbor interaction with a repulsive core. Most input parameters are the same as in `rilm.py`. Additional input parameters are:

- τ_{core}: range of hard core interactions ($\tau_{core} \sim 0.3$)
- A_{core}: strenght of hard core interactions ($A_{core} \sim 3$)
- dz: average position update ($dz \sim 1$)

iilm.py: algorithm



iilm.py: results

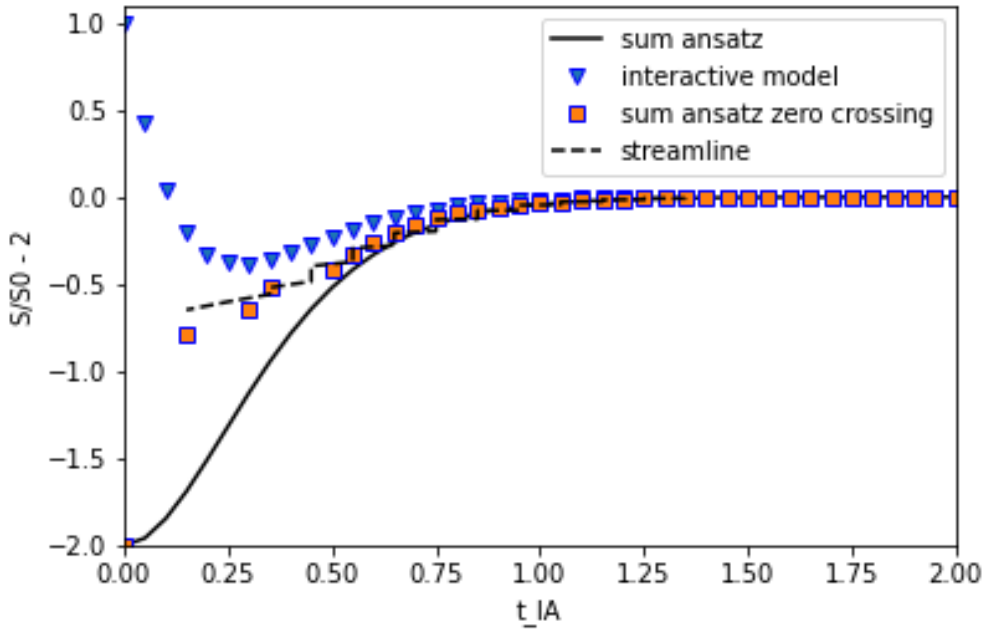


Figure: Instanton-anti-instanton interaction in units of S_0 as a function of the instanton-anti-instanton separation. The solid line is the result in the sum ansatz. The squares show the same data plotted as a function of the zero crossing distance. The streamline interaction is shown as the cutted line and the triangles show the *iilm.py* interaction results

iilm.py: results

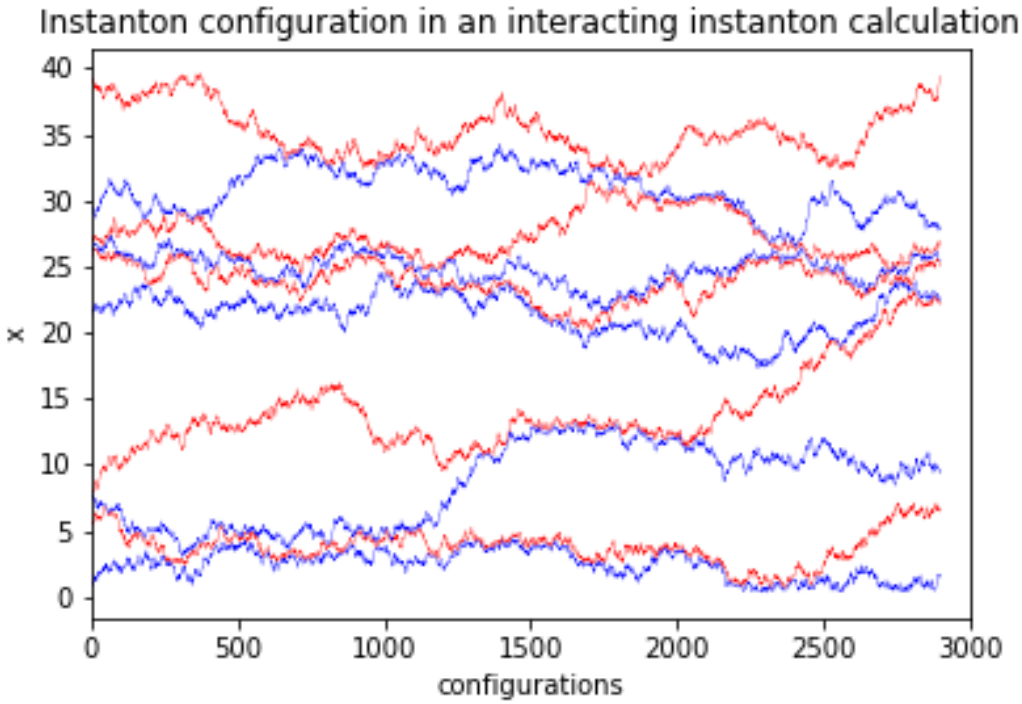


Figure: Typical instanton configuration in an interacting instanton calculation. The figure shows the location x of the first 10 instantons (blue) and anti-instantons (red) over a period of 3000 configurations.

To reproduce the hard-core repulsing interaction

$\rightarrow S_{core}(\tau_{IA}) = A_c e^{-\frac{\tau_{IA}}{\tau_c}}$, where $A_c = 3$ and $\tau_c = 0.3$

iilm.py: results

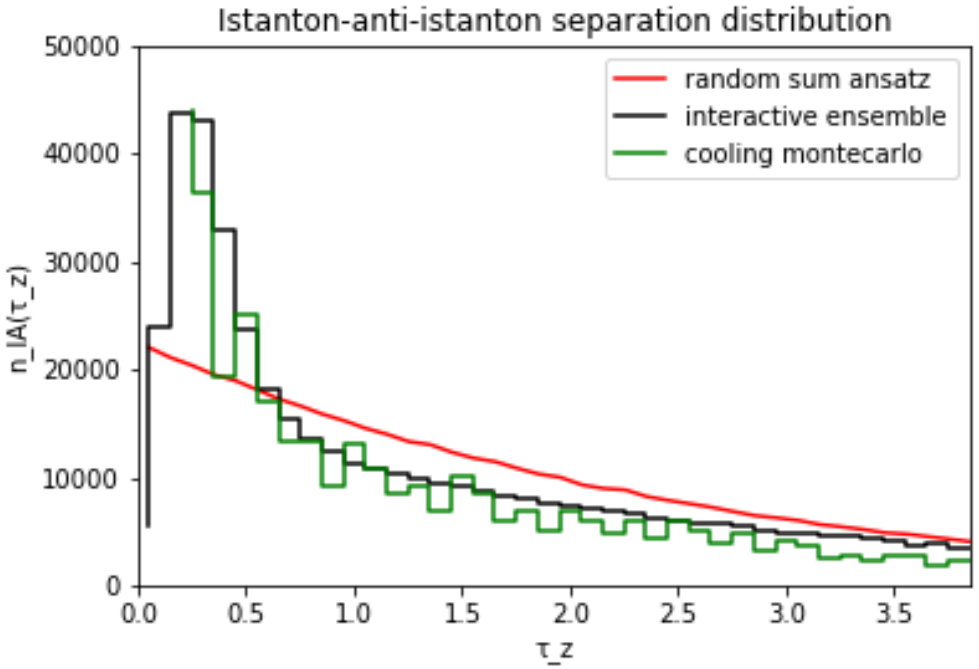


Figure: Distribution of IA separations after 10 cooling sweeps.

At small distance$\rightarrow$ enhancement of close IA pairs$\rightarrow$ attractive IA potential (short range);

iilm.py: results

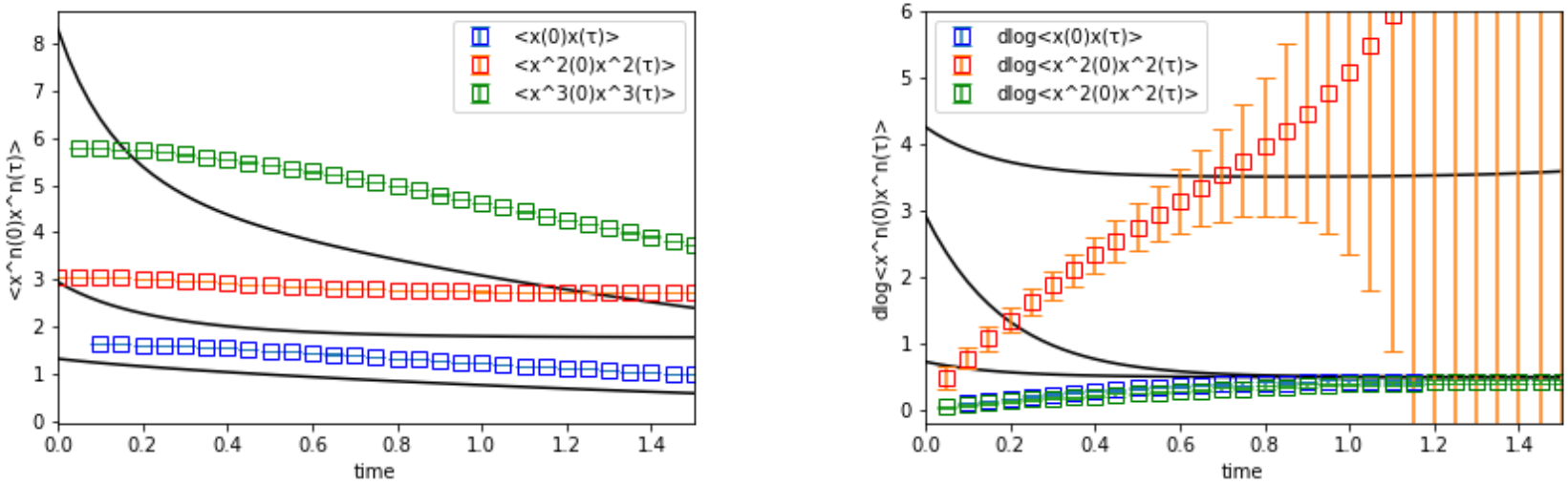


Figure: Correlation functions evaluated in an interacting instanton ensemble.

The correlation functions are similar to the random instanton ensemble because the interaction is small and only short range.

Conclusions

- Monte Carlo calculations of the euclidean path integral are an extremely poor way to compute the spectrum or the correlation functions of the anharmonic oscillator,
- The code based on diagonalizing the Hamiltonian is both much faster and much more accurate than the Monte Carlo codes (only for pedagogical purpose)
- However, in QM systems involving more degrees of freedom (4-dim. field theories) Monte Carlo calculations based on the euclidean path integral provide the most efficient method for computing the spectrum and the correlation functions.
- We focused on $S_0 \sim 3$ (boundary of the semi-classical regime), since:
  - if $S_0 >> 1$: we can identify easily the instantons, but with a low tunnelling rate
  - if $S_0 \sim 1$: instantons are very abundant and is hard to determine an instanton density