Advanced methods in Simulation: Part IV

Pascal Viot

November 13, 2020

Density of states

Defintion

For a classical systems described by a Hamiltonian $\mathcal{H}$, the microcanonical partition function is given by (see Chap. 1)

$$Z(E) = \sum_{\alpha} \delta(\mathcal{H} - E)$$

where E denotes the energy of the system and the index α runs over all available configurations.

Density of states

Defintion

For a classical systems described by a Hamiltonian $\mathcal{H}$, the microcanonical partition function is given by (see Chap. 1)

$$Z(E) = \sum_{\alpha} \delta(\mathcal{H} - E)$$

where E denotes the energy of the system and the index α runs over all available configurations. Introducing the degeneracy of the system for a given energy E, $g(E)$ The microcanonical partition function is then given as

$$Z(E) = \int dug(u)\delta(u - E) = g(E) = \exp(S(E))$$

where $S(E)$ is the entropy of the system.

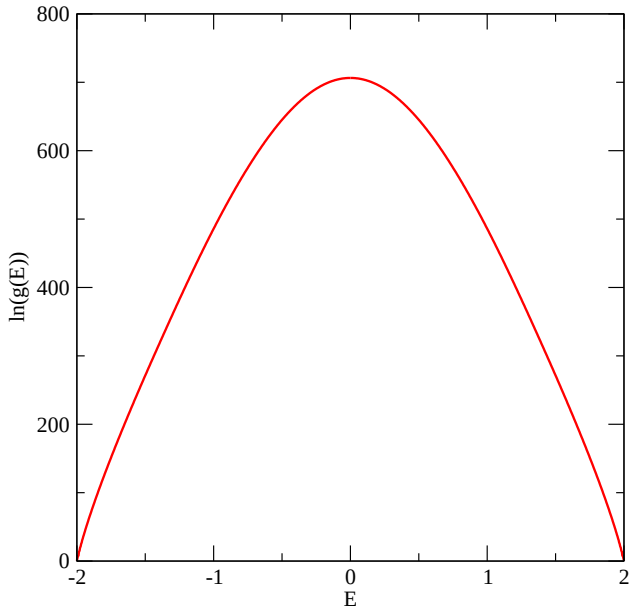
Density of states

For a Ising model, this number of microstates $\mathcal{N}$ is equal to 2^N where N is the total number of spins of the system. This value is then very large and grows exponentially with the system size. From a numerical point of view, one must use the logarithm of this function $\ln(g(E))$ for avoiding overflow problem.

Density of states

For a Ising model, this number of microstates $\mathcal{N}$ is equal to 2^N where N is the total number of spins of the system. This value is then very large and grows exponentially with the system size. From a numerical point of view, one must use the logarithm of this function $\ln(g(E))$ for avoiding overflow problem.

Logarithm of the density of states $\ln(g(E))$ for a $2D$ Ising model on a square lattice of linear size $L = 32$



Simulation in a canonical ensemble

Detailed balance

Monte Carlo algorithms satisfying the detailed balance give

$$\Pi(i \to j) P_{eq}(i) = \Pi(j \to i) P_{eq}(j)$$

$P_{eq}(j)$ is the probability of having the configuration j at equilibrium and $\Pi(i \to j$ is the transition probability of going from the state i towards the state j. One then rewrites detailed balance in energy variable.

Simulation in a canonical ensemble

Detailed balance

Monte Carlo algorithms satisfying the detailed balance give

$$\Pi(i \to j) P_{eq}(i) = \Pi(j \to i) P_{eq}(j)$$

$P_{eq}(j)$ is the probability of having the configuration j at equilibrium and $\Pi(i \to j$ is the transition probability of going from the state i towards the state j. One then rewrites detailed balance in energy variable.

Distribution of energy at equilibrium

Let us denote $H_\beta(E)$ the energy histogram associated to successive visits of the simulation. Consequently, one has

$$H_\beta(E) \propto g(E) \exp(-\beta E) \tag{1}$$

Biased Monte Carlo algorithm

Detailed balance

Introducing a weight function $w(E)$, the the detailed balance equation becomes

$$\Pi(E \to E') \exp(-\beta E + w(E)) = \Pi(E' \to E) \exp(-\beta E' + w(E')) \qquad (2)$$

Biased Monte Carlo algorithm

Detailed balance

Introducing a weight function $w(E)$, the the detailed balance equation becomes

$$\Pi(E \to E') \exp(-\beta E + w(E)) = \Pi(E' \to E) \exp(-\beta E' + w(E')) \tag{2}$$

Distribution of energy at equilibrium

Let us denote $H_\beta(E)$ the energy histogram associated to successive visits of the simulation. Consequently, one has

$$H_\beta(E) \propto g(E) \exp(-\beta E + w(E)) \tag{3}$$

Biased Monte Carlo algorithm

Detailed balance

Introducing a weight function $w(E)$, the the detailed balance equation becomes

$$\Pi(E \to E') \exp(-\beta E + w(E)) = \Pi(E' \to E) \exp(-\beta E' + w(E')) \tag{2}$$

Distribution of energy at equilibrium

Let us denote $H_\beta(E)$ the energy histogram associated to successive visits of the simulation. Consequently, one has

$$H_\beta(E) \propto g(E) \exp(-\beta E + w(E)) \tag{3}$$

Flat histogram H(E)

It immediately appears that, when $w(E) \sim \beta E - \ln(g(E))$, the energy histogram becomes a flat histogram and independent of the temperature. The detailed balance then becomes

$$\frac{\Pi(E \to E')}{g(E)} = \frac{\Pi(E' \to E)}{g(E')}$$

Convergence of the Algorithm

The stochastic process corresponds to a random walk in energy space. The convergence of this method is related to the fact that the available interval of the walker is bounded. For the Ising model, these bounds exist, for other models, it is possible of restricting the range of the energy interval in a simulation

Convergence of the Algorithm

The stochastic process corresponds to a random walk in energy space. The convergence of this method is related to the fact that the available interval of the walker is bounded. For the Ising model, these bounds exist, for other models, it is possible of restricting the range of the energy interval in a simulation

Multicanonical algorithm

This very interesting result is at the origin of multicanonical methods (introduced for the study of first-order transition). However, it is necessary to know the function $w(E)$, namely to know the function $g(E)$. In multicanonical method, a guess for $w(E)$ is provided from the density of states obtained (by simulation) with small system sizes. By extrapolating an expression for larger system sizes, the simulation is performed and by checking the deviations of the flatness of the energy histogram, one can correct the weight function, in order to flatten the energy histogram in a second simulation run.

Wang-Landau algorithm

The basics of the algorithm are:

1. A change of the initial configuration is proposed (generally a local modification)

Wang-Landau algorithm

The basics of the algorithm are:

1. A change of the initial configuration is proposed (generally a local modification)
2. One calculates the energy associated with the modified configuration E'. This configuration is accepted with a Metropolis rule $\Pi(E \to E') = Min(1, g(E)/g(E'))$, if not the configuration is rejected and the old configuration is kept (and added as in an usual Metropolis algorithm)

Wang-Landau algorithm

The basics of the algorithm are:

1. A change of the initial configuration is proposed (generally a local modification)
2. One calculates the energy associated with the modified configuration E'. This configuration is accepted with a Metropolis rule $\Pi(E \to E') = Min(1, g(E)/g(E'))$, if not the configuration is rejected and the old configuration is kept (and added as in an usual Metropolis algorithm)
3. The accepted configuration modifies the density of states as follows: $\ln(g(E')) \leftarrow \ln(g(E')) + \ln(f)$

Wang-Landau algorithm (2)

The basics of the algorithm are:

1. A change of the initial configuration is proposed (generally a local modification)

Wang-Landau algorithm (2)

The basics of the algorithm are:

1. A change of the initial configuration is proposed (generally a local modification)
2. One calculates the energy associated with the modified configuration E'. This configuration is accepted with a Metropolis rule $\Pi(E \to E') = Min(1, g(E)/g(E'))$, if not the configuration is rejected and the old configuration is kept (and added as in an usual Metropolis algorithm)

Wang-Landau algorithm (2)

The basics of the algorithm are:

1. A change of the initial configuration is proposed (generally a local modification)
2. One calculates the energy associated with the modified configuration E'. This configuration is accepted with a Metropolis rule $\Pi(E \to E') = Min(1, g(E)/g(E'))$, if not the configuration is rejected and the old configuration is kept (and added as in an usual Metropolis algorithm)
3. The accepted configuration modifies the density of states as follows: $\ln(g(E')) \leftarrow \ln(g(E')) + \ln(f)$ where f is called the modification factor.

Wang-Landau algorithm (2)

The basics of the algorithm are:

1. A change of the initial configuration is proposed (generally a local modification)
2. One calculates the energy associated with the modified configuration E'. This configuration is accepted with a Metropolis rule $\Pi(E \to E') = Min(1, g(E)/g(E'))$, if not the configuration is rejected and the old configuration is kept (and added as in an usual Metropolis algorithm)
3. The accepted configuration modifies the density of states as follows: $\ln(g(E')) \leftarrow \ln(g(E')) + \ln(f)$ where f is called the modification factor.

Wang-Landau algorithm (2)

The basics of the algorithm are:

1. A change of the initial configuration is proposed (generally a local modification)
2. One calculates the energy associated with the modified configuration E'. This configuration is accepted with a Metropolis rule $\Pi(E \to E') = Min(1, g(E)/g(E'))$, if not the configuration is rejected and the old configuration is kept (and added as in an usual Metropolis algorithm)
3. The accepted configuration modifies the density of states as follows: $\ln(g(E')) \leftarrow \ln(g(E')) + \ln(f)$ where f is called the modification factor.

This iterative scheme is repeated until the energy histogram is sufficiently flat: practically, the original algorithm proposes the histogram is flat when each value of the histogram must be larger than 80% of the mean value of this histogram.

Wang-Landau algorithm (3)

1. Because the density of states changes continuously in time, it is not a stationary quantity. The detailed balance is broken!

Wang-Landau algorithm (3)

1. Because the density of states changes continuously in time, it is not a stationary quantity. The detailed balance is broken!
2. The accuracy of the density of states depends on the statistical quantity of each value of $g(E)$. Because the density of states increases with time, the asymptotic accuracy is given by $\sqrt{\ln(f)}$.

Wang-Landau algorithm (3)

1. Because the density of states changes continuously in time, it is not a stationary quantity. The detailed balance is broken!
2. The accuracy of the density of states depends on the statistical quantity of each value of $g(E)$. Because the density of states increases with time, the asymptotic accuracy is given by $\sqrt{\ln(f)}$.
3. To this first stage, the precision is not sufficient for calculating the thermodynamic quantities.

Wang-Landau algorithm (3)

1. Because the density of states changes continuously in time, it is not a stationary quantity. The detailed balance is broken!
2. The accuracy of the density of states depends on the statistical quantity of each value of $g(E)$. Because the density of states increases with time, the asymptotic accuracy is given by $\sqrt{\ln(f)}$.
3. To this first stage, the precision is not sufficient for calculating the thermodynamic quantities.

Wang-Landau algorithm (3)

1. Because the density of states changes continuously in time, it is not a stationary quantity. The detailed balance is broken!
2. The accuracy of the density of states depends on the statistical quantity of each value of $g(E)$. Because the density of states increases with time, the asymptotic accuracy is given by $\sqrt{\ln(f)}$.
3. To this first stage, the precision is not sufficient for calculating the thermodynamic quantities.

Second part of the algorithm

One resets the energy histogram (but, obviously not the histogram of the density of states). One restart a new simulation described above, but with a new modification factor equal to $\sqrt{f}$. Once this second stage achieved, the density of states is refined with respect to the previous iteration, because the accuracy of the histogram of the density of states is given by $\sqrt{\ln(f)/2}$.

Wang-Landau algorithm (4)

1. In order to have an accurate density of states, the simulation is restarted many times by decreasing the modification factor

$$f \leftarrow \sqrt{f}$$

Wang-Landau algorithm (4)

1. In order to have an accurate density of states, the simulation is restarted many times by decreasing the modification factor

$$f \leftarrow \sqrt{f}$$

2. When $\ln(f) \sim 10^{-8}$, the density of states does not longer evolve, the dynamics becomes a random walk in energy space and the simulation is then stopped.

Wang-Landau algorithm (4)

1. In order to have an accurate density of states, the simulation is restarted many times by decreasing the modification factor

$$f \leftarrow \sqrt{f}$$

2. When $\ln(f) \sim 10^{-8}$, the density of states does not longer evolve, the dynamics becomes a random walk in energy space and the simulation is then stopped.

Wang-Landau algorithm (4)

1. In order to have an accurate density of states, the simulation is restarted many times by decreasing the modification factor

$$f \leftarrow \sqrt{f}$$

2. When $\ln(f) \sim 10^{-8}$, the density of states does not longer evolve, the dynamics becomes a random walk in energy space and the simulation is then stopped.

Once the density of states obtained, one can easily derive all thermodynamic quantities. If the accuracy is not reached, the simulation can be easily prolongated by using the previous density of states until a sufficient accuracy is reached.

Wang-Landau algorithm (5)

A perfect method?

1. If the motion in the energy space is purely diffusive (only true for the last iterations of the algorithm, namely values of the modification factor close to one), the computer time is proportional to $(\Delta E)^2$, where ΔE is the energy interval of the simulation.

Wang-Landau algorithm (5)

A perfect method?

1. If the motion in the energy space is purely diffusive (only true for the last iterations of the algorithm, namely values of the modification factor close to one), the computer time is proportional to $(\Delta E)^2$, where ΔE is the energy interval of the simulation.
2. For decreasing the computing time, one can reduce the energy interval, and perform several simulations for different (overlapping) energy ranges. In the second case, the total duration of all simulations is divided by N_I, the numbers of intervals, compared to an unique simulation over the complete energy space.

   Unfortunately, the matching does not work , if the energy interval belongs to the region of a transition. The density of states is slightly biased at the extremities of the energy interval

Wang-Landau algorithm (6)

A perfect method?

1. The method is very efficient for lattice systems with discrete energy spectra (additional problems occur with continuous energy spectra.

Wang-Landau algorithm (6)

A perfect method?

1. The method is very efficient for lattice systems with discrete energy spectra (additional problems occur with continuous energy spectra.
2. Free energy barriers disappear for first order transition, and it allows for an efficient exploration of phase space. The method is also suitable for continuous phase transitions.

Wang-Landau algorithm (6)

A perfect method?

1. The method is very efficient for lattice systems with discrete energy spectra (additional problems occur with continuous energy spectra.
2. Free energy barriers disappear for first order transition, and it allows for an efficient exploration of phase space. The method is also suitable for continuous phase transitions.

Wang-Landau algorithm (6)

A perfect method?

1. The method is very efficient for lattice systems with discrete energy spectra (additional problems occur with continuous energy spectra.
2. Free energy barriers disappear for first order transition, and it allows for an efficient exploration of phase space. The method is also suitable for continuous phase transitions.

Thermodynamics

Once obtained the density of states (to an additive constant), one obtains the mean energy (in a canonical ensemble) a posteriori by computing the ratio of following integrals

$$< E >= \frac{\int E \exp(-\beta E + \ln(g(E)))dE}{\int \exp(-\beta E + \ln(g(E)))dE}$$

Wang-Landau algorithm (7)

Thermodynamics

1. The arguments of these exponentials can be very large and lead to overflows. For avoiding these problems, let us note that for a given temperature (or β), the expression

$$-\beta E + \ln(g(E)) \tag{4}$$

has one or several maxima (but close) and which decreases rapidly toward negative values when the energy is much smaller or greater to these minima. Without changing the result, one can multiply each integral by $\exp(-Max(-\beta E + \ln(g(E))))$ The exponentials have now an argument always negative or equal to zero.

Wang-Landau algorithm (7)

Thermodynamics

1. The arguments of these exponentials can be very large and lead to overflows. For avoiding these problems, let us note that for a given temperature (or β), the expression

$$-\beta E + \ln(g(E)) \tag{4}$$

has one or several maxima (but close) and which decreases rapidly toward negative values when the energy is much smaller or greater to these minima. Without changing the result, one can multiply each integral by $\exp(-Max(-\beta E + \ln(g(E))))$ The exponentials have now an argument always negative or equal to zero.

2. Truncate the bounds of each integral when the arguments of each exponential become equal to -100. Integration is now safe!

Wang-Landau algorithm (7)

Thermodynamics

1. The arguments of these exponentials can be very large and lead to overflows. For avoiding these problems, let us note that for a given temperature (or β), the expression

$$-\beta E + \ln(g(E)) \tag{4}$$

has one or several maxima (but close) and which decreases rapidly toward negative values when the energy is much smaller or greater to these minima. Without changing the result, one can multiply each integral by $\exp(-Max(-\beta E + \ln(g(E))))$ The exponentials have now an argument always negative or equal to zero.

2. Truncate the bounds of each integral when the arguments of each exponential become equal to -100. Integration is now safe!

3. The specific heat is obtained with the usual formula

$$C_v = k_B \beta^2 \left(\langle E^2 \rangle - \langle E \rangle^2 \right) \tag{5}$$

Wang-Landau algorithm (8)

Thermodynamics

1. A last main advantage of this method is to obtain the free energy of the system for all temperatures as well as the entropy of the system, a different situation of a Monte Carlo simulation in the canonical ensemble.

Wang-Landau algorithm (8)

Thermodynamics

1. A last main advantage of this method is to obtain the free energy of the system for all temperatures as well as the entropy of the system, a different situation of a Monte Carlo simulation in the canonical ensemble.
2. Store the magnetization histogram $M(E)$ as well as the square magnetization $M^2(E)$, even higher moments $M^n(E)$.... From these histograms and the density of states $g(E)$, one can calculate the mean magnetization of the system as a function of the temperature by using the following formula

$$< M(\beta) >= \frac{\int M(E) \exp(-\beta E + \ln(g(E)))dE}{\int \exp(-\beta E + \ln(g(E)))dE}$$

Wang-Landau algorithm (8)

Thermodynamics

1. A last main advantage of this method is to obtain the free energy of the system for all temperatures as well as the entropy of the system, a different situation of a Monte Carlo simulation in the canonical ensemble.
2. Store the magnetization histogram $M(E)$ as well as the square magnetization $M^2(E)$, even higher moments $M^n(E)$.... From these histograms and the density of states $g(E)$, one can calculate the mean magnetization of the system as a function of the temperature by using the following formula

$$< M(\beta) >= \frac{\int M(E) \exp(-\beta E + \ln(g(E)))dE}{\int \exp(-\beta E + \ln(g(E)))dE}$$

3. Wang-Landau simulation in real time.

Replica exchange algorithm

1. The free energy barriers prevent the exploration of the phase diagram in a reasonable computer time. The convergence of Metropolis-like algorithms becomes very slow, because the typical relaxation time is proportional to the exponential of the free energy barriers, which grow with the system size.

Replica exchange algorithm

1. The free energy barriers prevent the exploration of the phase diagram in a reasonable computer time. The convergence of Metropolis-like algorithms becomes very slow, because the typical relaxation time is proportional to the exponential of the free energy barriers, which grow with the system size.
2. The "parallel tempering" method (or replica exchange method) is inspired by the following remark: when the temperature decreases, equilibrium is more and more difficult to reach, due to the presence of large free energy barriers.

Replica exchange algorithm

1. The free energy barriers prevent the exploration of the phase diagram in a reasonable computer time. The convergence of Metropolis-like algorithms becomes very slow, because the typical relaxation time is proportional to the exponential of the free energy barriers, which grow with the system size.
2. The "parallel tempering" method (or replica exchange method) is inspired by the following remark: when the temperature decreases, equilibrium is more and more difficult to reach, due to the presence of large free energy barriers.
3. One consider a system where one only changes the temperature, but the method is easy generalized to other intensive quantities (chemical potential, modification of the Hamiltonian,...): the canonical partition function of a system to the temperature T_i is given by the relation

$$Q_i = \sum_{\alpha} \exp(-\beta_i U(\alpha))$$

where α is an index running over all configurations available to the system and β_i the inverse of the temperature.

Replica exchange algorithm (2)

1. Let us consider the direct product of all systems evolving to different temperatures, the corresponding partition function of this new ensemble is equal to

$$Q_{total} = \prod_{i=1}^{N} Q_i$$

where N denotes the total number of temperatures used in all simulation boxes.

Replica exchange algorithm (2)

1. Let us consider the direct product of all systems evolving to different temperatures, the corresponding partition function of this new ensemble is equal to

$$Q_{total} = \prod_{i=1}^{N} Q_i$$

where N denotes the total number of temperatures used in all simulation boxes.

2. The parallel tempering method introduces an additional Markov chain between simulation boxes, the internal Markov chain of each simulation box are unchanged.

Replica exchange algorithm (2)

1. Let us consider the direct product of all systems evolving to different temperatures, the corresponding partition function of this new ensemble is equal to

$$Q_{total} = \prod_{i=1}^{N} Q_i$$

   where N denotes the total number of temperatures used in all simulation boxes.

2. The parallel tempering method introduces an additional Markov chain between simulation boxes, the internal Markov chain of each simulation box are unchanged.

3. In order to ensure convergence towards equilibrium, a detailed balance rule must be added between two consecutive simulation boxes.

Replica exchange algorithm (3)

1. The detailed balance is expressed as

$$\frac{\Pi((i,\beta_i),(j,\beta_j)\rightarrow(j,\beta_i),(i,\beta_j))}{\Pi((i,\beta_j),(j,\beta_i)\rightarrow(i,\beta_i),(j,\beta_j))}=\frac{\exp\left(-\beta_j U(i)\right)\exp\left(-\beta_i U(j)\right)}{\exp\left(-\beta_i U(i)\right)\exp\left(-\beta_j U(j)\right)}$$

where $U(i)$ et $U_(j)$ are the total energies of the boxes i and j, respectively.

Replica exchange algorithm (3)

1. The detailed balance is expressed as

$$\frac{\Pi((i,\beta_i),(j,\beta_j)\rightarrow(j,\beta_i),(i,\beta_j))}{\Pi((i,\beta_j),(j,\beta_i)\rightarrow(i,\beta_i),(j,\beta_j))}=\frac{\exp\left(-\beta_j U(i)\right)\exp\left(-\beta_i U(j)\right)}{\exp\left(-\beta_i U(i)\right)\exp\left(-\beta_j U(j)\right)}$$

   where $U(i)$ et $U_(j)$ are the total energies of the boxes i and j, respectively.

2. The parallel tempering method introduces an additional Markov chain between simulation boxes, the internal Markov chain of each simulation box are unchanged.

Replica exchange algorithm (4)

The algorithm consists of two kinds of move:

1. Particle motion in each box, which can be performed in parallel according to a Metropolis rule. The acceptance probability for a single move within a simulation box is given by

$$\min(1, \exp(-\beta_i(U_i(n) - U_i(o)))) \tag{6}$$

where n and o denote the new and old configurations, respectively.

Replica exchange algorithm (4)

The algorithm consists of two kinds of move:

1. Particle motion in each box, which can be performed in parallel according to a Metropolis rule. The acceptance probability for a single move within a simulation box is given by

$$\min(1, \exp(-\beta_i(U_i(n) - U_i(o)))) \tag{6}$$

   where n and o denote the new and old configurations, respectively.

2. Particle exchange between two consecutive boxes. The acceptance probability is then given by

$$\min(1, \exp(-(\beta_i - \beta_j)(U_j - U_i))) \tag{7}$$

Replica exchange algorithm (4)

The algorithm consists of two kinds of move:

1. Particle motion in each box, which can be performed in parallel according to a Metropolis rule. The acceptance probability for a single move within a simulation box is given by

$$\min(1, \exp(-\beta_i(U_i(n) - U_i(o)))) \tag{6}$$

   where n and o denote the new and old configurations, respectively.

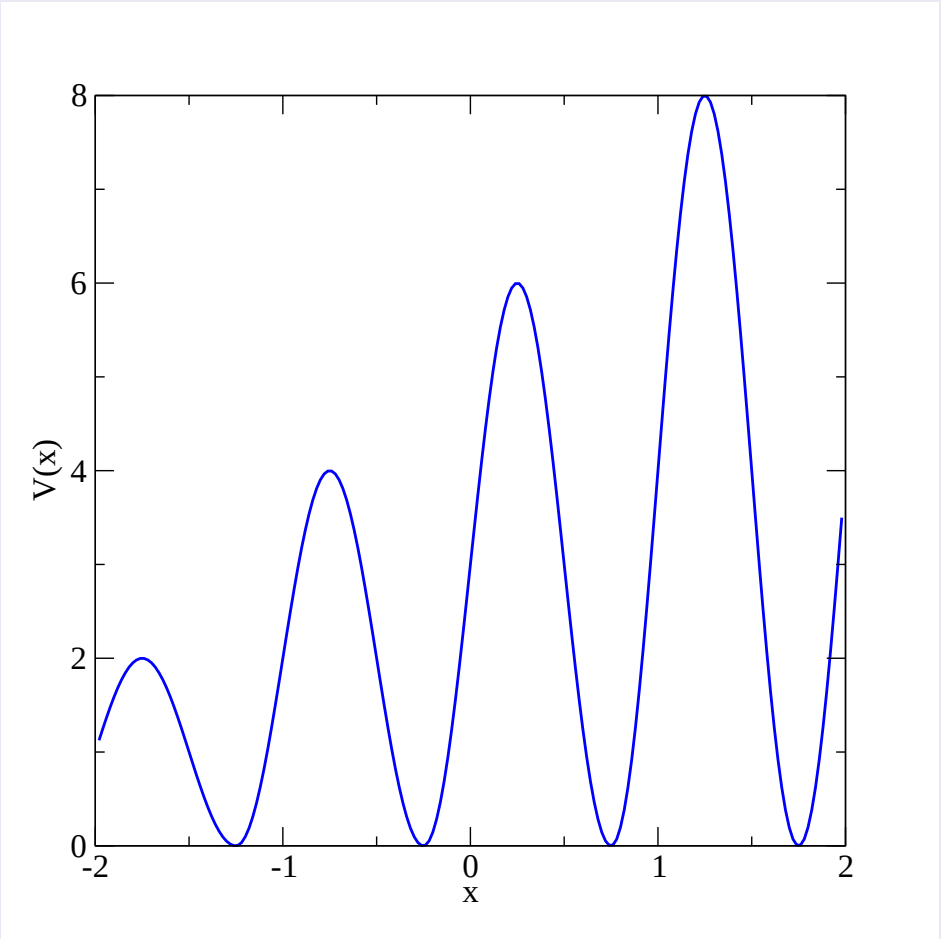
2. Particle exchange between two consecutive boxes. The acceptance probability is then given by

$$\min(1, \exp(-(\beta_i - \beta_j)(U_j - U_i))) \tag{7}$$

3. The exchange fraction beween boxes must to be set to optimize the simulation, as well as the temperature interval.

Replica exchange algorithm (4)

Real-time simulation of a toy model: a particle moves in an finite interval subjected to a external potential $V(x)$



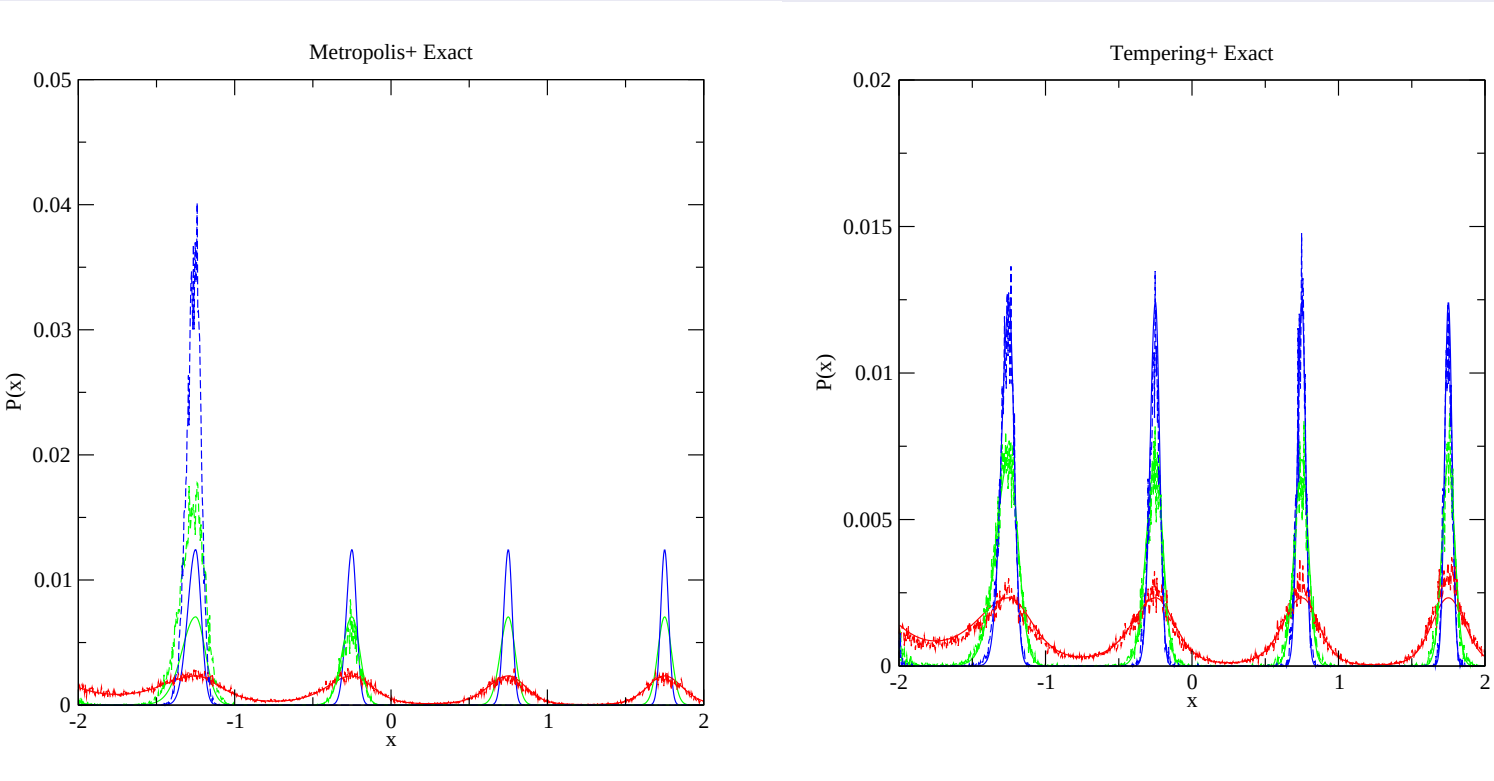
Replica exchange algorithm (5)

$$V(x) = \begin{cases} +\infty & \text{si } x < -2 \\ 1 + \sin(2\pi x) & \text{if } -2 \leq x \leq 1.25 \\ 2 + 2\sin(2\pi x) & \text{if } -1.25 \leq x \leq -0.25 \\ 3 + 3\sin(2\pi x) & \text{if } -0.25 \leq x \leq 0.75 \\ 4 + 4\sin(2\pi x) & \text{if } 0.75 \leq x \leq 1.75 \\ 5 + 5\sin(2\pi x) & \text{if } 1.75 \leq x \leq 2.0 \end{cases}$$

The equilibrium probabilities $P_{eq}(x, \beta)$ can be calculated exactly

$$P_{eq}(x, \beta) = \frac{\exp(-\beta V(x))}{\int_{-2}^{2} dx \exp(-\beta V(x))} \tag{8}$$

Replica exchange algorithm (6)



Simulations with three temperatures: T_1 (blue), T_2 (green) and T_3 (red)