Ecole Normale Supérieure de Lyon — Université de Lyon
Master: Sciences de la Matière, A. A. 2024-2025

Computational Quantum Physics (M2) — T. Roscilde, F. Mezzacapo, F. Caleca, S. Bocini

General rule for the TD sessions: the TD sessions are fully hands-on – namely, in every TD session you are supposed to write computer codes to learn about the phenomenology and efficiency of important algorithms for problems in quantum physics. You should choose a programming platform (Python, Matlab, Mathematica, C, Fortran, etc.), and you should be able to plot your results in the form of two-dimensional functions $y = f(x)$ (using matplotlib in Python, the plotting utilities of Matlab and Mathematica, Gnuplot, etc.), or occasionally in a more complicated form. We assume that you have some familiarity with at least one programming platform; if this is not the case, you should be able to familiarize yourself rapidly e.g. by attending online tutorials.

TD4: Imaginary-time Path Integral Monte Carlo

In this exercise sheet, we apply the Imaginary-time Path Integral Monte Carlo (PIMC) method to investigate the finite-temperature physics of quantum particles subjected to an external harmonic potential. As a starting point we specialize our attention to the case of a single particle in one spatial dimension. The Hamiltonian of the system of our interest is $H = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial y^2} + \frac{1}{2}m\omega^2 y^2$, where ω is the angular frequency, $\hbar$ the reduced Planck constant and y the particle position. By choosing $E_0 = \hbar\omega$ and $y_0 = \sqrt{\frac{\hbar}{m\omega}}$ as units of energy and length, respectively, one obtains the dimensionless form:

$$\mathcal{H} = -\frac{1}{2}\frac{\partial^2}{\partial x^2} + v(x), \text{ with } v(x) = \frac{1}{2}x^2. \tag{1}$$

1 Useful formulas

The partition function of our system at a temperature $1/\beta$ is:

$$\mathcal{Z} = \int dx_0 \left\langle x_0 | e^{-\beta\mathcal{H}} | x_0 \right\rangle. \tag{2}$$

The integrand in Eq. (2) can be analytically evaluated due to the particular Hamiltonian in Eq. (1), however, given the purpose of this TD we will use an approximation which is i) systematically improvable and ii) applicable regardless the details of the Hamiltonian. By inserting $M-1$ identities in Eq. (2) we obtain:

$$\mathcal{Z} = \int \prod_{k=0}^{M-1} dx_k \left\langle x_k | e^{-\tau\mathcal{H}} | x_{k+1} \right\rangle, \tag{3}$$

where $\tau = \beta/M$ is the imaginary time step, M the number of slices, $\boldsymbol{x} = x_0, \cdots, x_{M-1}$ is the path or world line and $x_M = x_0$. The k_{th} imaginary time propagator in Eq. (3) can be approximated as:

$$\left\langle x_k | e^{-\tau\mathcal{H}} | x_{k+1} \right\rangle \simeq (2\pi\tau)^{-\frac{1}{2}} e^{-\frac{(x_{k+1}-x_k)^2}{2\tau}} e^{-\tau v(x_k)}. \tag{4}$$

The above is known as primitive approximation: it is exact in the limit of small τ and

accurate up to order τ^2. Defining

$$P(\boldsymbol{x}) = \frac{\prod_{k=0}^{M-1} \langle x_k | e^{-\tau \mathcal{H}} | x_{k+1} \rangle}{\mathcal{Z}} \simeq \frac{\prod_{k=0}^{M-1} (2\pi\tau)^{-\frac{1}{2}} e^{-\frac{(x_{k+1}-x_k)^2}{2\tau}} e^{-\tau v(x_k)}}{\mathcal{Z}}, \tag{5}$$

the expectation value of an operator $\mathcal{O}$ in terms of its estimator $O(\boldsymbol{x})$ is:

$$EXP[\mathcal{O}] = \int \mathcal{D}(\boldsymbol{x}) P(\boldsymbol{x}) O(\boldsymbol{x}), \text{ where } \mathcal{D}(\boldsymbol{x}) \equiv dx_0 dx_1, \cdots, dx_{M-1}. \tag{6}$$

For general models, Eq. (6) cannot be solved exactly. The PIMC method, in a nutshell, provides an efficient way of estimating the multidimensional integral above. Specifically, via PIMC one generates a large set of configurations $\boldsymbol{x}^1, \cdots, \boldsymbol{x}^{N_c}$ sampled from $P(\boldsymbol{x})$ and approximates Eq. (6) as:

$$EXP[\mathcal{O}] \simeq \frac{1}{N_c} \sum_{i=1}^{N_c} O(\boldsymbol{x}^i) = \langle O(\boldsymbol{x}) \rangle \,. \tag{7}$$

You may have noticed that in this section we have used two approximations: the primitive one for the propagators and that in Eq. (7). They are both "under control", becoming exact in the large M and N_c limit, respectively.

2 Setting up the code

Our PIMC code should be written in terms of a few adjustable parameters and arrays which will be progressively defined and consistently used in the TD sheets.

For example, **you may start using**:

β: dimensionless inverse temperature.

M: number of imaginary time slices, or beads.

$\boldsymbol{x}$: array of size M with the particle position at each slice (i.e., the world line).

N_c: number of configurations generated by our PIMC algorithm. A new configuration is generated starting from the current one, proposing a certain number of single-bead displacements (see below).

Also, you will need to use standard functions included in essentially all programming languages to extract uniformly or normally distributed random numbers.

2.1 Sampling the configuration space

The sampling strategy is one of the fundamental point of a PIMC code. As a first step you should assign the system initial configuration e.g., by uniformly extracting the position of our single particle for each bead in a certain range $[-XMAX, XMAX]$. Then you can modify the initial configuration by proposing M sequential single-bead updates as follows:

for($k = 0$; $k < M$; $k++$)
{
  Randomly select a bead j;
  Propose a new position for the selected bead:
    $x_j^{new} = \frac{x_{j+1}+x_{j-1}}{2} + g(\sqrt{\tau/2})$;
  Perform Metropolis acceptance (rejection) test and update (or not) x_j;
}
new configuration obtained;

The above pseudo-code where $g(\sqrt{\tau/2})$ is a random number distributed according to a gaussian with zero mean and standard deviation equal to $\sqrt{\tau/2}$ provides a simple "recipe" to generate a new global configuration by means of M sequential local updates. In the following, the procedure of attempting M sequential single-bead updates will be referred to as the proposition of a global update.

Within the scheme discussed above the displacement of a single bead j is accepted, for our model in Eq. (1), with probability

$$A(x_j \to x_j^{new}) = min\Big(1, e^{-\frac{\tau}{2}[(x_j^{new})^2-(x_j)^2]}\Big). \tag{8}$$

2.2

After having implemented the sampling strategy introduced in this section you can do a first numerical exercise:
Initialize the system setting e.g., $x_i = 0 \; \forall \; i = 0, \cdots, M-1$. Propose several global updates (e.g., 5×10^5) and plot the final configuration for $\beta = 1.0$ and $\beta = 10^{-2}$, as well as the initial one (use $M = 10$). What do you observe? [You may need to rigidly shift one of the two world lines to visualize both in the same spatial range].

3 Estimating observables: Energy

We can now evaluate physical observables as averages over the sampled configurations of the corresponding estimators. For the total energy E of our particle the estimator reads:

$$\mathcal{E}(\boldsymbol{x}) = \frac{1}{M}\sum_{j=1}^{M}\left[\frac{1}{2\tau} - \frac{1}{2}\left(\frac{x_{j+1}-x_j}{\tau}\right)^2 + v(x_j)\right]. \tag{9}$$

Hence,

$$E \simeq \langle \mathcal{E}(\boldsymbol{x}) \rangle \tag{10}$$

3.1

Set $\beta = 0.5$ and $M = 10$. Propose a large number $NTHERM$ of global updates to thermalize the system. Then, generate via global updates N_c configurations and compute the energy and its uncertainty. Compare your result with the exact one: $E^{ex}(\beta = 0.5) = 2.04149$.

3.2

Here we will explore the dependence of E on M. Set, for example, $\beta = 5.0$ and estimate the energy and its uncertainty for a few values of M in the interval $[1, 50]$. What do you expect to obtain for $M = 1$? Plot $E(M)$ and compare your estimates with the exact result $E^{ex}(\beta = 5.0) = 0.50678$.

3.3

Estimate $E(\beta)$ for a few values of β, for example, in the interval $[0.5, 10]$. Plot $E(\beta)$ and compare it with the exact result:

$$E^{ex}(\beta) = \frac{1}{2\tanh(\beta/2)}. \tag{11}$$

In this exercise you can use $\tau = \beta/M = 0.1$.

4 Generalizing the code

At this point you should have a working code for the case of a single particle. Such a code can be easily extended to the case of 2 and more distinguishable particles. To this goal you should:
1) Add an index to your configuration array $\boldsymbol{x}$: $x_j \to x_{i,j}$ where i, and j are the particle and slice index, respectively.
2) Use the straightforward generalization of the propagator in Eq. (4):

$$\left\langle \boldsymbol{x}_k | e^{-\tau \mathcal{H}_N} | \boldsymbol{x}_{k+1} \right\rangle \simeq (2\pi\tau)^{-\frac{N}{2}} e^{\sum_{i=1}^{N} \left[-\frac{\left(x_{i,k+1} - x_{i,k}\right)^2}{2\tau} - \tau v(x_{i,k}) \right]} \tag{12}$$

where $\boldsymbol{x}_k \equiv \{x_{1,k}, \cdots, x_{N,k}\}$ and $\mathcal{H}_N = \sum_{i=1}^{N} \left(-\frac{1}{2}\frac{\partial^2}{\partial x_i^2} + v(x_i) \right)$.
3) Define a global update as the propsition of $N \times M$ single-bead updates where the particle and the bead to update are both randomly chosen each time.
4) Use the generalized form of Eq. (9), obtainable by simply taking an extra summation over the number of particles. The energy estimator will then read:

$$\mathcal{E}(\boldsymbol{x}) = \sum_{i=1}^{N} \frac{1}{M} \sum_{j=1}^{M} \left[\frac{1}{2\tau} - \frac{1}{2} \left(\frac{x_{i,j+1} - x_{i,j}}{\tau} \right)^2 + v(x_{i,j}) \right] \tag{13}$$

4.1 The swap update

In order to account for particle indistinguishability, we have to introduce an additional update known as the swap update. Here we will illustrate a very basic implementation of this update.

Let us consider two particles 1 and 2, and the corresponding world lines

$$\begin{aligned} \boldsymbol{x}_1 &\equiv x_{1,0}, \cdots, x_{1,M-1} \text{ (with } x_{1,M} \equiv x_{1,0}) \text{ and} \\ \boldsymbol{x}_2 &\equiv x_{2,0}, \cdots, x_{2,M-1} \text{ (with } x_{2,M} \equiv x_{2,0}). \end{aligned} \tag{14}$$

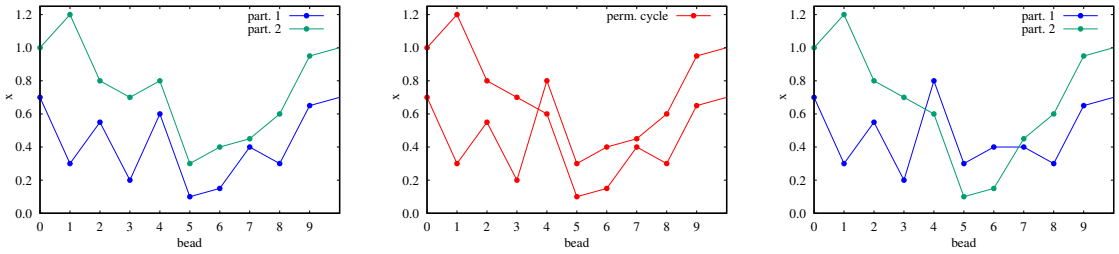

Figure 1: Illustration of the swap update: Two distinct world lines of $M = 10$ beads (left panel) are involved in a permutation cycle after the acceptance of a swap update (middle panel). A subsequent swap destroys the cycle and the resulting world lines are again separate (right panel).

The attempt of a swap update consists in randomly selecting a bead k (with $k \neq 0$) and proposing a new configuration $\boldsymbol{x}^{new} \equiv \{\boldsymbol{x}_1^{new}, \boldsymbol{x}_2^{new}\}$ such that

$$\begin{aligned} \boldsymbol{x}_1^{new} &\equiv x_{1,0}, \cdots, x_{1,k-1}, x_{2,k}, \cdots, x_{2,M-1} \text{ (with } x_{1,M}^{new} \equiv x_{2,0}) \text{ and} \\ \boldsymbol{x}_2^{new} &\equiv x_{2,0}, \cdots, x_{2,k-1}, x_{1,k}, \cdots, x_{1,M-1} \text{ (with } x_{2,M}^{new} \equiv x_{1,0}). \end{aligned} \tag{15}$$

Figure 1 illustrates the effect of two consecutive swap updates on a given two-particle configuration. Starting from two separate world lines (each comprising $M = 10$ beads) a swap has the effect of "gluing" them in a permutation cycle (middle panel). In this peculiar configuration one can come back to a given initial bead following the path only after having visited all the beads of the the two particles. The permutation cycle disappears after the acceptance of a second swap which leads to a configuration where each world line is again a distinct imaginary-time-periodic entity (right panel).

4.2

Implement the swap update explained above and add it to your code. After this, you should be able to investigate the physics of our model of interest in the case of distinguishable, or indistinguishable bosonic particles. You may organize the sampling making use of the following strategy for a global update:

```
for (j = 0; j < N_s; j++)
{
  Randomly select a bead k (with k ≠ 0);
  Extract a random number ε uniformly distributed in [0, 1];
  If (ε < THR) {Propose_swap;}
  else {Propose_displacement;}
  Perform Metropolis test and update (or not) the configuration;
}
new configuration obtained;
```

In the above pseudo-code N_s is the number of single updates proposed (you can set, for example, $N_s = \alpha MN$ with $\alpha = 0.5 - 1.0$), THR is a real number in $[0, 1]$ which determines the probability of proposing a swap or a displacement single update Test your code by computing the energy per particle for $N = 2$ at an inverse temperature $\beta = 0.5$. Compare your result with the exact one: $e_{Bs}^{ex}(\beta = 0.5) = 1.8527$ and that of question 3.1. What can you conclude?