Project 2, Path integral formalism. Deadline June 1, Spring 2022

Computational Physics II FYS4411/FYS9411

Department of Physics, University of Oslo, Norway

May 13, 2022

Path Integrals: Concepts and Formalism

The aim of this project is to study the path integral formalism applied first to a Harmonic oscillator problem and then to a spin-like Hamiltonian. This provide us with many of the basic elements for moving to part b). Part a) starts with a field-theoretic approach reformulating the path integral in euclidean time, see for example the text of Gattringer and Lang.

It introduces concepts like the Trotter-Suzuki approximation and how to evaluate the energy for a simple Hamiltonian like the harmonic oscillator in one dimension. It bridges thus the gap between project 1 and the present as well as serving as an introduction to part b). For the second part, due to the structure of the Hamiltonian, one can circumvent the needs for doing a Trotter expansion.

In the path integral formalism the evolution from a state $|x_i\rangle$ to a state $|x_f\rangle$ from time t_i to time t_f is given by:

$$\langle x_f|e^{-\hat{H}(t_f-t_i)}|x_i\rangle = \int \mathcal{D}x(t)e^{-S[x]}$$

On the left-hand side we have the standard quantum mechanical time evolution operator with $\hat{H}$ being the Hamiltonian operator of the system, the right-hand side represents the path integral of all possible paths x(t) from x_i to x_f with $t = t_i \rightarrow t_f$ weighted with the classical action $S[x]$:

$$S[x] = \int_{t_i}^{t_f} dtL(x,\dot{x}) = \int_{t_i}^{t_f} dt\left(\frac{m\dot{x}(t)^2}{2} - V(x(t))\right),$$

and knowledge of the propagator gives full information about the system.
In a simple one dimensional case the path integral can be written as

$$\int \mathcal{D}x(t) \rightarrow \int_{-\infty}^{\infty} dx_1 dx_2 \dots dx_{N-1}$$

where time has been discretized in N slices from $t_i = t_0$ to $t_f = t_N$. Consequently $x_i = x(t_i)$.

Given some functional $\Gamma[x]$ that represents an observable at intermediate times of the evolution of the system we can write its expectation value as:

$$\langle \Gamma[x] \rangle = \frac{1}{Z} \int \mathcal{D}x(t) \Gamma[x] e^{-S[x]}$$

where the normalization Z is given by

$$Z = \int \mathcal{D}x(t) e^{-S[x]}$$

This can be computed using a set of configurations $x^{(k)} = \{x_1^{(k)} x_2^{(k)} \dots x_{N-1}^{(k)}\}$ for $k = 1, 2, \dots N_{cf}$. The Monte Carlo estimator then becomes:

$$\langle \Gamma[x] \rangle \approx \bar{\Gamma} = \frac{1}{N_{cf}} \sum_{k=1}^{k} \Gamma[x^{(k)}]$$

As and aside, if we wish to generalize our theory to a four dimensional euclidean space time an observable that depends on a scalar field can be computed in a very similar fashion

$$\langle \Gamma[\phi] \rangle = \frac{1}{Z} \int \mathcal{D}\phi \Gamma[\phi] e^{-S[\phi]}$$

where the normalization Z is given again by

$$Z = \int \mathcal{D}\phi e^{-S[\phi]}$$

But this time the path integral can be computed numerically by discretizing the 4 dimensional space in a lattice:

$$\int \mathcal{D}\phi \to \int \prod_{x_i \in lattice} d\phi(x_i)$$

The first exercise here follows closely LePage's article. See also Ref. [1] below.

The Harmonic Oscillator: a Simple Example

Project 2 a): A first example of the path integral solution of quantum mechanical systems is the one-dimensional harmonic oscillator. It is particularly well suited because of:

1. simple dimensional discretization
2. well-known results for expectation values
3. fast analytical analysis, implementation and runs

Here we follow LePage in defining the classical action for the harmonic oscillator. This action is given by

$$S[x] = \int_{t_i}^{t_f} \left[\frac{m\dot{x}(t)^2}{2} + \frac{1}{2}\omega x(t)^2 \right] dt.$$

We can discretize the time domain into N slices of step a, which imposes us to discretize the time derivative of the position as well. When considering the integral between time t_j and t_{j+1} the action becomes

$$S_j \approx a \left[\frac{m(x_{j+1} - x_j)^2}{2a^2} + \frac{1}{4}\omega(x_{j+1} - x_j)^2 \right] dt.$$

When summing over all time slices (and applying periodic boundary conditions) we get

$$S_{latt} = \sum_{j=0}^{N-1} \left[\frac{m}{2a}(x_{j+1} - x_j)^2 + \frac{a}{2}x_j^2 \right]$$

In order to extract information from the system we need to construct and study an observable. We can look at a two point correlator of the form $\langle x(t_2)x(t_1)\rangle$. In the continuum this is equivalent to the calculation of the ratio of path integrals

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

In quantum mechanics the numerator (using the definition of the path integral) is equal to

$$v \int dx \langle x|e^{\bar{H}(t_f - t2)}\tilde{x}e^{\bar{H}(t_2 - t1)}\tilde{x}e^{\bar{H}(t_1 - ti)}|x\rangle,$$

If we look at the full integral, let the hamiltionian operator act on the states leaving the spectral decomposition of them, and substitute $T = t_f - Ti$ and $t = t_2 - t_1$ we get to

$$\langle x(t_2)x(t_1)\rangle = \frac{\sum e^{-E_n T}\langle E_n|\tilde{x}e^{-(\bar{H} - E_n)t}\tilde{x}|E_n\rangle}{\sum e^{-E_n T}}$$

and for $T \gg t$ the ground state will dominate the summation:

$$G(t) = \langle x(t_2)x(t_1)\rangle \to \langle E_0|\tilde{x}e^{-(\bar{H} - E_0)t}\tilde{x}|E_0\rangle,$$

and letting t to be large as well the propagator ends up linking the two lowest energy states

$$G(t) \to |\langle E_0|\tilde{x}|E_1\rangle|^2 e^{-(E_1 - E_0)t}.$$

We can now extract the first excitation energy of the quantum mechanical harmonic oscillator, in the limit of t large, as

$$\log\left(\frac{G(t)}{G(t+\Delta t)}\right) = (E_1 - E_0)\Delta t.$$

On our discretized lattice the correlator can be obtained numerically by directly computing the average value of the operator

$$G(t) = \frac{1}{N}\sum_j \langle x(t_j + t)x(t_j)\rangle,$$

for all t in $0, a, 2a \ldots (N-1)a$. This might look trivial at a first glance, but it is the key point of the whole algorithm.

The computational algorithm required to compute the correlator is based on the idea of creating a Markov chain of possible paths for the harmonic oscillator, via subsequent updates. This procedure is called Metropolis algorithm:

1. initialize an array of N position values for each time point (for example set everything to 0)
2. suggest a new random configuration starting from the current one
3. accept or reject the update, based on the action difference
4. compute the correlator for the current configuration
5. repeat the process N_{cf} times, sufficiently large to have a statistically relevant ensemble.

In order to limit the auto correlation between data, the observable is only computed once every N_{corr} updates. The rule for accepting an update is: $\exp(-\Delta S) > \xi$. Where $\xi \in [0, 1]$, a randomly chosen number. This implies that if the classical action is reduced the update is automatically accepted, if it is positive, a random number ξ is generated and compared with the exponential of the action. This condition allows the system to explore all the phase space and not get stuck in a local minimum for example.

Your task is to run for the harmonic oscillator using $10^5 - 10^7$ configurations (N_{conf}), skipping 20 updates between every measurement (N_{corr}). The time axis is discretized into 20 nodes. Data, to improve the estimate of the error can be resampled using the bootstrap or the blocking techniques. The result that is physically relevant is the first excitation energy, which we expect to be $1\hbar\omega$. For simplicity we set $\hbar = \omega = 1$ and the lattice spacing parameter $a = 0.5$.

Implement statistical bootstrapping and blocking. Estimate energy and how it relates with the correlation time. This exercise follows closely LePage's article. You may find it a useful read.

Project 2 b): Here the aim is to explore the Heisenberg-model in one and two dimensions, by use of stochastic series expansion (SSE). Hopefully, we will be able to calculate the critical temperature for the continious phase transition. With this value the critical exponents for the system can also be obtained. The results should be compared with the relevant literature as provided in the references.

Implementation of SSE for 1D Heisenberg model with periodic bound- ary conditions.

1. Add external field and anisotropy dependencies.
2. 2D implementation, again with periodic boundary conditions.
3. Study the temperature dependent phase transition and calculate the critical exponents for the system. By studying the system the follow- ing values should be attainable; M, M^2, E, ξ, C_V , $C(r)$, where M is magnetization, E is the energy, ξ is the magnetic suceptibility, C_V is the heat capacity and $C(r)$ is the correlation length.

Literature.

1. G. P. LePage, Lattice QCD for novices, https://arxiv.org/pdf/hep-lat/0506036.pdf, 2005
2. A. W. Sandvik. Computational Studies of Quantum Spin Systems. https://doi.org/10.1063/1.3518900, Jan. 2011.
3. A. W. Sandvik. Finite-size scaling of the ground-state parameters of the two-dimensional heisenberg model. Phys. Rev. B, 56:11678–11690, Nov 1997.
4. A. W. Sandvik. Stochastic series expansion method with operator-loop update. Phys. Rev. B, 59:R14157–R14160, Jun 1999.
5. O. F. Syljuasen and A. W. Sandvik. Quantum Monte Carlo with directed loops. Phys. Rev. E, 66:046701, Oct 2002.

Introduction to numerical projects

Here follows a brief recipe and recommendation on how to write a report for each project.

- Give a short description of the nature of the problem and the eventual numerical methods you have used.
- Describe the algorithm you have used and/or developed. Here you may find it convenient to use pseudocoding. In many cases you can describe the algorithm in the program itself.

- Include the source code of your program. Comment your program properly.
- If possible, try to find analytic solutions, or known limits in order to test your program when developing the code.
- Include your results either in figure form or in a table. Remember to label your results. All tables and figures should have relevant captions and labels on the axes.
- Try to evaluate the reliabilty and numerical stability/precision of your results. If possible, include a qualitative and/or quantitative discussion of the numerical stability, eventual loss of precision etc.
- Try to give an interpretation of you results in your answers to the problems.
- Critique: if possible include your comments and reflections about the exercise, whether you felt you learnt something, ideas for improvements and other thoughts you've made when solving the exercise. We wish to keep this course at the interactive level and your comments can help us improve it.
- Try to establish a practice where you log your work at the computerlab. You may find such a logbook very handy at later stages in your work, especially when you don't properly remember what a previous test version of your program did. Here you could also record the time spent on solving the exercise, various algorithms you may have tested or other topics which you feel worthy of mentioning.

Format for electronic delivery of report and programs

The preferred format for the report is a PDF file. You can also use DOC or postscript formats or as an ipython notebook file. As programming language we prefer that you choose between C/C++, Fortran2008 or Python. The following prescription should be followed when preparing the report:

- Use canvas to hand in your projects, log in at `http://canvas.uio.no` with your normal UiO username and password.
- Upload **only** the report file! For the source code file(s) you have developed please provide us with your link to your github domain. The report file should include all of your discussions and a list of the codes you have developed. The full version of the codes should be in your github repository.
- In your github repository, please include a folder which contains selected results. These can be in the form of output from your code for a selected set of runs and input parameters.
- Still in your github make a folder where you place your codes.

- In this and all later projects, you should include tests (for example unit tests) of your code(s).
- Comments from us on your projects, approval or not, corrections to be made etc can be found under your Devilry domain and are only visible to you and the teachers of the course.

Finally, we encourage you to work two and two together. Optimal working groups consist of 2-3 students. You can then hand in a common report.