united nations educational, scientific and cultural organization

international atomic energy agency

the
abdus salam
international centre for theoretical physics

ICTP 40th Anniversary

SMR 1595 - 23

Joint DEMOCRITOS - ICTP School on
CONTINUUM QUANTUM MONTE CARLO METHODS
12 - 23 January 2004

COMPUTER LABORATORY SESSION

BASIC PATH INTEGRAL EXERCISES: PARTICLE IN AN HARMONIC OSCILLATOR POTENTIAL

Markus HOLZMANN
Univ. de Paris VI (Pierre et Marie Curie), Lab. de Physique Theorique des Liquides
F-75252 Paris, France

These are preliminary lecture notes, intended only for distribution to participants.

I. BASIC PATH INTEGRAL EXERCISES: PARTICLE IN AN HARMONIC OSCILLATOR POTENTIAL

The aim of these exercises is to get familiar with basic Monte Carlo sampling techniques at finite temperature.

Harmonic Oscillator Action

The harmonic oscillator potential is one of the rare cases where the single particle density matrix $\langle \mathbf{R}|e^{-\beta H}|\mathbf{R}'\rangle$ at temperature $T = 1/\beta$ in D dimensions can be calculated analytically (e.g. the book on Statistical Mechanics of Feynman):

$$\langle \mathbf{R}|e^{-\beta H}|\mathbf{R}'\rangle = \prod_{j=1}^{D} \rho(r_j, r_j'; \beta) \tag{1}$$

with

$$H = \frac{\mathbf{P}^2}{2m} + \frac{1}{2}\omega^2 \mathbf{R}^2 \tag{2}$$

and

$$\rho(x, x', \beta) = (2\pi \sinh(\omega\beta)/\omega)^{-1/2} \exp\left\{-\frac{\omega}{4}\left[\tanh(\omega\beta/2)(x + x')^2 + \frac{(x - x')^2}{\tanh(\omega\beta/2)}\right]\right\} \tag{3}$$

In the following we always use units with $\hbar = m = \omega = 1$, length is measured in units of the harmonic oscillator length $a_{ho} = (\hbar/m\omega)^{1/2}$, and energies are measured in unites of $\hbar\omega$).

For a general (N-body) Hamiltonian, an analytical result for the density matrix at low temperature is not known, but one can introduce (imaginary) time slices to express it as a discrete path integral and use high temperature approximations. In this exercise we want to demonstrate Path Integral Monte Carlo, calculating the diagonal part of the density matrix of the harmonic oscilator numerically via path integration. We will use the primitive approximation as high temperature approximation for the action:

$$S_{p.a}(x, x'; \tau) = \frac{(x - x')^2}{2\tau} + \frac{\tau}{4}\omega^2(x^2 + x'^2), \tag{4}$$

valid in the limit $\tau \to 0$. (The action at temperature β is defined by $S(x, x', \beta) = -\log \rho(x, x', \beta)$.)

Statistical weight of a configuration/ Metropolis

The action determines the statistical weight $\Pi_{x(\tau)}$ for a given (discretized) path $x(\tau)$,
$x(\tau) \equiv [\mathbf{R}_0 \equiv \mathbf{R}(0), \mathbf{R}_1 \equiv \mathbf{R}(\tau), \ldots, \mathbf{R}_l \equiv \mathbf{R}(\beta)]$ with $l = \beta/\tau$ discretization points.

$$\Pi_{x(\tau)} = \exp\left\{-\sum_{i=0}^{l-1} S_{p.a}(\mathbf{R}_i, \mathbf{R}_{i+1}; \tau)\right\} \tag{5}$$

To sample the path of the (quantum) particle, we use an a-priori probability $T_{x(\tau)\to x'(\tau)}$ to suggest a new path and use the Metropolis algorithm to accept/reject this proposed move with probability

$$P_{x(\tau)\to x'(\tau)} = \min\left[1, \frac{\Pi_{x'(\tau)}}{\Pi_{x(\tau)}} \frac{T_{x'(\tau)\to x(\tau)}}{T_{x(\tau)\to x'(\tau)}}\right] \tag{6}$$

Sampling a path efficiently/ Levy construction

In the harmonic oscillator program HO the a-priori probability is separated into two parts. First we move the end-coordinates $\mathbf{R}_0 \equiv \mathbf{R}(\tau = 0) = \mathbf{R}(\tau = \beta)$ according to a certain distribution $T_{\rm dia}$, e.g. a gaussian (in an homogeneous system one would rather use a uniform distribution). After we have chosen the end-coordinates, we can use the so-called Levy-construction to create the discretized path, $\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_{l-1}$ with l=Nslice discretization points ($\tau = \beta/l$) to construct the rest of the path with probability T_{Levy}.

The Levy-construction uses the convolution properties of gaussian distributions. It can be regarded as a tool to sample a path with given end-points $\mathbf{R_0}$, $\mathbf{R}_l$ according to a probability $p(\mathbf{R}_1, \mathbf{R}_2, \ldots \mathbf{R}_{l-1})$ determined by a single particle density matrix with Hamiltonian H_0:

$$p(\mathbf{R_1}, \mathbf{R}_2, \ldots \mathbf{R}_{l-1}) \propto \langle \mathbf{R}_0 | e^{-\tau H_0} | \mathbf{R}_1 \rangle \langle \mathbf{R}_1 | e^{-\tau H_0} | \mathbf{R}_2 \rangle \cdots \langle \mathbf{R}_{l-1} | e^{-\tau H_0} | \mathbf{R}_l \rangle \tag{7}$$

$$\propto \frac{\langle \mathbf{R}_0 | e^{-\tau H_0} | \mathbf{R}_1 \rangle \langle \mathbf{R}_1 | e^{-(l-1)\tau H_0} | \mathbf{R}_l \rangle}{\langle \mathbf{R}_0 | e^{-l\tau H_0} | \mathbf{R}_l \rangle} \cdot \frac{\langle \mathbf{R}_1 | e^{-\tau H_0} | \mathbf{R}_2 \rangle \langle \mathbf{R}_2 | e^{-(l-2)\tau H_0} | \mathbf{R}_l \rangle}{\langle \mathbf{R}_1 | e^{-(l-1)\tau H_0} | \mathbf{R}_l \rangle} \cdot \ldots \cdot \frac{\langle \mathbf{R}_{l-2} | e^{-\tau H_0} | \mathbf{R}_{l-1} \rangle \langle \mathbf{R}_{l-1} | e^{-\tau H_0} | \mathbf{R}_l \rangle}{\langle \mathbf{R}_{l-2} | e^{-2\tau H_0} | \mathbf{R}_l \rangle} \tag{8}$$

Since it is of the form

$$p(\mathbf{R}_1, ..., \mathbf{R}_{l-1}) \propto p_1(\mathbf{R}_0, \mathbf{R}_1, \mathbf{R}_l) p_2(\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_l) \cdots p_{l-1}(\mathbf{R}_{l-2}, \mathbf{R}_{l-1}, \mathbf{R}_l) \tag{9}$$

where the p_i's are already properly normalized, $\int d\mathbf{R} p_k(\mathbf{R}_{k-1}, \mathbf{R}, \mathbf{R}_l) = 1$, we can recursively sample the path $\{\mathbf{R}_i\}$ with weight $p(\{\mathbf{R}_i\})$ for fixed endpoints $\mathbf{R}_0$, $\mathbf{R}_l$.

Basic Observables

After a certain number of Monte-Carlo moves, we look at different estimators to calculate the mean energy and the density profile. These two quantities are used to study the convergence of the path integral.

Parameters to be adapted in the code HO

- The harmonic oscillator program is in the directoy HO1. To use the program for our case, check the following parameters in **oszi.1.dat**:
  - We need only one particle: **Nboson**=1, in one spatial dimension: **Idim**=1
  - We only make moves in position space: **plevy**=1., no moves in permutation space: **p2move**=0.
  - we have no interparticle interaction: **a_0**=0.
  - **Nslice** is the number of time slices, **Deltau**=$\tau = \beta/$Nslice.
  - **Maxtime** is the number of Monte-Carlo iterations, should be of the order $10^4 - 10^6$.
- The main program is **oszi.1.f**. Here we have to set the inverse temperature **Beta**, choose a value between 0.1 and 2.0. Use **init**=.true. (.false.) to create an initial configuration (read an already existing configuration from the file **zinput.oszi**).
  At the end of the calculation the output configuration is written into the file **zoutput.oszi**. You can use this file as an input by renaming it ito **zinput.oszi** and choosing **init**=.true..
- In **action_sp.f**, **ichoice**=2 is set to use the primitive approximation for the action.
- To compile type **make**, then start the calculation with **ho.out**.

Exercises

- Calculate the density profile and the mean energy for temperatures ranging around $\beta = 2.$, and study the convergence of the results by doubling the number of time slices, starting with Nslice=1. (Do not forget to recompile after modifying the program.)
  - The density profile is written in the file **zdenscut.ho**. Plot it, and compare it with the analytical formula above. (**Ihmax** and **Xhistomax** control the range and bin-widths used for the density profile; the density profile is calculated in the subroutine **denscut** found in **observables.f**.
  - Look at the estimator of the energy using the primitive approximation and study its convergence vs. Nslice for different temperatures.
  - Look at the kinetic energy and the varianz of the kinetic energy (both calculated using the primitive approximation). What happens? How can one improve?
- Since the analytical expression for the density matrix is known, we can also use the exact action instead of the primitive approximation. This is realized by setting **ichoice**=1 in the subroutine **action_sp**. The energy estimator using the exact harmonic oscillator action is written after the program finished. How many time-slices do you need for convergence?

II. BOSONS IN AN HARMONIC OSCILLATOR POTENTIAL

For the bosonic path integral we have to include a random walk in permutation space. HO tries to permute two randomly chosen particles (subroutine **p2move.f**). The acceptance ratio for a permutation move is typically an order of magnitude less than for moves in position space. If we set **plevy**=1 in **oszi.1.dat**, we should choose **p2perm** an order of magnitude bigger, e.g. **p2perm**=20.

Bose-Einstein condensation of an ideal gas in a harmonic trap

Look at **Nboson**=1000 ideal bosons. Since we are working now with the exact harmonic oscillator single particle action (ichoice=1 in **action_sp.f**), we can put **Nslice**=1 and **Lmax=1000** in **oszi.1.dat** (Lmax restricts the number of new "bridges" sampled in each position move to Nslice*Lmax "beads". I prepared some input files in the subdirectory INPUT, copy the file you want to use to **zinput.oszi** in the directory containing the exectutable, and use **init**=.false. in **oszi.1.f** to read the initial configurations from the input file. The parameters (which you have to adapt in the files **oszi.1.dat** and **oszi.1.f**) of the input files are

- input.0.1.ideal: Nboson=1000, Beta=0.1, Nslice=1, a_0=0.
- input.0.12.ideal: Nboson=1000, Beta=0.12, Nslice=1, a_0=0.
- input.0.15.ideal: Nboson=1000, Beta=0.15, Nslice=1, a_0=0.
- input.0.2.ideal: Nboson=1000, Beta=0.2, Nslice=1, a_0=0.
- input.0.3.ideal: Nboson=1000, Beta=0.3, Nslice=1, a_0=0.

Calculate the density profile for temperatures around and below the transition temperature of the ideal gas. Plot the density profile in **zdenscut.ho**, and the mean cycle distribution (cumulative) in **zicycle.oszi**. (The mean cycle distribution plots the mean number of particles involved in an exchange cycle of lenght smaller or equal k, versus the length of the cycle k) What does happen around and below the transition temperature?

Note, that the maximal cycle length at a certain temperature can be used as an estimate for the condensate fraction for the ideal and very dilute gas.

Choose a temperature far below the transition temperature and look at the superfluid fraction estimator using the area formular. Study its convergence against Nslice. Use smaller number of particles $N \sim 100$ ($\beta_c \sim 0.2$).

Dilute Bose gas

Now consider an interacting ensemble where the interparticle potential is characterized by the s-wave scattering length a_0. In the subdirectory INPUT, I prepared some input-files which are already thermalized. The corresponding parameters are

- set **Lmax**=10 and a_0=0.043 in **oszi.1.dat**
- input.0.1.1000: Nboson=1000, Beta=0.1, Nslice=1, a_0=0.043
- input.0.12.1000: Nboson=1000, Beta=0.12, Nslice=2, a_0=0.043
- input.0.15.1000: Nboson=1000, Beta=0.15, Nslice=2, a_0=0.043
- input.0.2.1000: Nboson=1000, Beta=0.2, Nslice=2, a_0=0.043
- input.0.3.1000: Nboson=1000, Beta=0.3, Nslice=3, a_0=0.043

The parameters are chosen such that they correspond to the first experiment of Bose Einstein condensation which had around 10,000 Rb atoms in a harmonic trap. At this low temperatures the interactions is well described by a scattering length $a\_0 \sim 0.0043$ in units of the harmonic oscillator length. Since $\sim Na\_0$ is the relevant quantity we adapted a_0 for less particles.

For the interactomic interaction action, HO uses an analytical solution to the two-particle density matrix where only changes in the low-energy s-wave channnel are taken into account. For a dilute system it becomes precise at temperatures slightly above the transition temperature of BEC. Therefore the number of time slices is chosen to give $\tau = \beta/\mathrm{Nslice}$ just above the BEC transition.

III. OVERVIEW OVER THE ROUTINES OF THE HARMONIC OSCILLATOR PROGRAM HO

This code is partially taken from "bos98.f" written by Werner Krauth and Markus Holzmann.

- in general $m = \hbar = k_B = 1$
- **oszi.f**: Main program containing the Monte-Carlo loop and initializes the global variables **Beta** (inverse temperature) and the vector **Omega**(Idim=1,2,3) containing the trap frequencies in the different spatial directions. **init=.true.** must be set to create an initial configuration, whereas **init=.false.** reads the initial configuration from the file **zinput.oszi**.
- **oszi.1.dat**: contains all global parameters
  - Nboson: number of Bosons
  - Idim: number of spatial dimensions
  - Nslice: number of time slices, Deltau=Beta/Nslice: time step
  - **R(j,kslice,iboson)** contains the position of iboson at time kslice* Deltau
  - R(j,Nslice+1,iboson)=R(j,1,Ipermute(iboson))
  - Maxtime: number of MC iterations
  - plevy, p2perm: relative weight of position (plevy) and permutation (p2perm) moves in a-priori sampling
  - Ihmax, Xhistomax: used for histograms of the density distribution
- **levymove.f**: picks a particle and try to change its spatial position. If permutations are sampled, up to **Lmax** particles on the same permutation cycle are moved
- **p2move**: tries to permute two particles (pair permutation): starting from the actual permutation stored in the vector Ipermute(1:Nboson), we try to change to a new permutation P_{ij}^{exch} Ipermute(.), or (i,Ipermute(i)),(j,Ipermute(j)) → (i,Ipermute(j)),(j,Ipermute(i))
- **oszidens.f** contains:
  - changeglobal(l,x):
samples the diagonal elements of the density matrix with an a-priori probabilty which is gaussian with standard deviation squared in sigma2_dia (depeding on temperature le*beta)
  - sample_dia(l,x) calculates the a-priori probability to move the particle to position x with changeglobal.
  - levy(l,**r**,tau):
For given r_1 and given r_l samples $r_2, ...r_{l-1}$ according to the non interacting density matrix via the **Levy** construction. sigma2_cm and sigma2_rel determine the standard deviation squared for the center of mass and relative part of the gaussian.
  - sample_levy: calculates the a-priori probabilty corresponding to the Levy move
  - rho(i,j):
calculates the free density matrix elements between particles i and j.
  - pcycle(iboson,ipx(.),l,lmax):
determines the permutation cycle starting from iboson and writes it into the vector ipx(.), ipx(1)=iboson, ipx(2)=Ipermute(iboson), ipx(3)=Ipermute^2(iboson),... ipx(l)=Ipermute^l(iboson)=iboson
if l>lmax only the first lmax elements of ipx() are determined
- **action_sp.f** calculates the single-particle action, either the exact harmonic oscillator action (action_ho, ichoice=1) or the primitive approximation (primitve, ichoice=2). Contains also rhofeyn(x,y,tau), which calculates the density matrix elements of the harmonic oszillator
- **aprioriX_ho** contains various functions determining the stadard deviations of the a-priori sampling, e.q. sigma2_dia, sigma2_cm, sigma2_rel. You have to choose in the **makefile** what routine you want to take.
- **interaction.f** calculates the action due to pair-particle interactions

  - proba(epot): calculates the action and the interparticle interaction energy by summing over all possible pairs. Used to initialize the interparticle interaction energy and to control the action due to interactions
  - interaction: calculates the change of the action and the potential energies due to interactions
  - densmat: calculates the action/energy for a single pair of atims using an analytical formula in pseudopot to describe hard-sphere scattering at low temperatures.

- **random.f** contains:
  - nran(i) : creates an integer random number between 1 and i
  - gauss(Idum,sigma) : creates a gaussian number with std sigma
  - ran3(Idum): creates a random number uniform between 0 and 1, (C) Numerical recipes, **Idum** must be negative for initialization, ran3 must always be real*4.

- **observables.f**: contains various estimator like **denscut()** for calculating the density profile along some axis (integrated over the other axis), **energy()** to calculate the some energy-estimators, **cyclen** to calculate the mean cycle distribution, **rhos_area** to calculate the superfluid density using the area-formula.

- **initial.f**: handles input and output files
  initial(init) creates a new configuration if init=.true. and reads otherwise an existing configuration from the file zinput.oszi
  output(): writes the end configuration in the file zoutput.oszi. This file can afterwords directly be used as the input file if one wants to continue an old simulation.