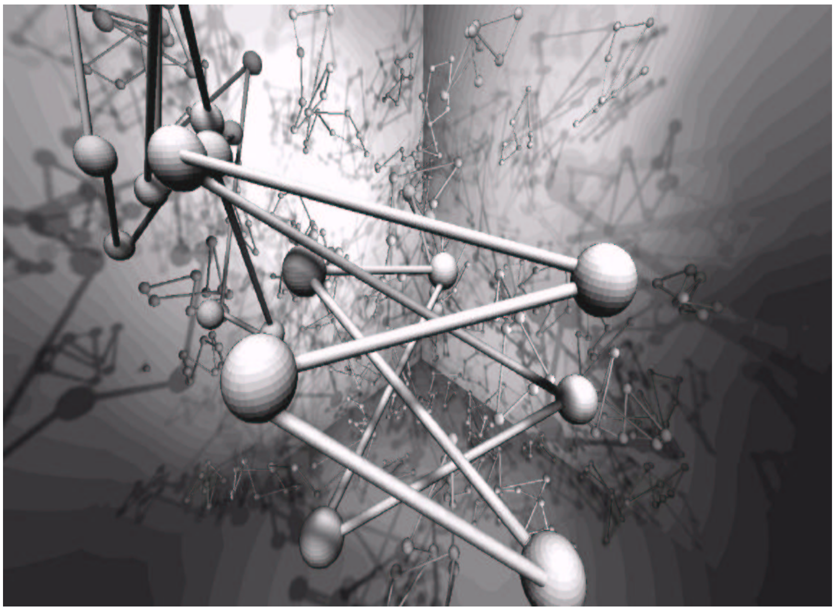
UNIVERSITAT
POLITÈCNICA DE
CATALUNYA
UPC

Path integral Monte Carlo

Algorithms and applications to quantum fluids



Llorenç Brualla i Barberà

Doctoral Dissertation
Departament de Fisica i Enginyeria Nuclear
Barcelona 2002

Path integral Monte Carlo

Algorithms and applications to quantum fluids

Llorenç Brualla i Barberà

Universitat Politècnica de Catalunya

Path integral Monte Carlo
Algorithms and applications to quantum fluids

Path integral Monte Carlo

Algorithms and applications to quantum fluids

Llorenç Brualla i Barberà

Memòria presentada per optar al grau de
Doctor en Ciències Fisiques

July 16, 2002

Universitat Politècnica de Catalunya

The programs in this book have been included for their instructional value. They have been tested with care but are not guaranteed for any particular purpose. The author does not offer any warranties or representations, nor does he accept any liabilities with respect to the programs

Cover: *PIMC simulation of liquid* ^{4}*He at 5 K, 64 atoms with 8 beads per atom*
Typeset by the author with the LaTeX Documentation System

To Teresa

Miserabile. Aveva sopra il capo l'unico luogo stabile del cosmo, l'unico riscatto alla dannazione del *panta rei*, e pensava che fossero affari Suoi, e non suoi. E infatti subito dopo la coppia si allontanò—lui educato su qualche manuale che gli aveva ottenebrato la possibilità di meraviglia, lei inerte, inaccessibile al brivido dell'infinito, entrambi senza aver registrato nella propria memoria l'esperienza terrificante di quel loro incontro—primo e ultimo—con l'Uno, l'En–sof, l'Indicibile. Come non cadere in ginocchio davanti all'altare della certezza?

(Umberto Eco, *Il pendolo di Foucault*, 1988)

ACKNOWLEDGEMENTS

Special thanks are due to Jordi Boronat and Joaquim Casulleras, without whose constant encouragement and fruitful suggestions this thesis would not have been written. First billing goes to Jordi Boronat as the esteemed fulfiller of the supervisory tasks. Alongside the provision of academic freedom, Jordi also manages to pass on the necessary discipline of thinking, as well as an even requirement of quality. The atmosphere of Jordi's group is collegian after his own nature and encourages, if not requires, a great broadening of one's skill base. I am delighted to have had the opportunity to do so. Much gratitude goes also to Joaquim Casulleras for providing the right measures of excitement, encouragement, and doubt. Quim has introduced me to the art of translating the real world into codes, a task he masters with craftsman's ability. His skills straddle well beyond the desktop, and have brought us to spent countless cold stargazing nights.

Heartfelt thankfulness to Romualdo Pastor. He has been unstinting in his efforts to help. I am in debt with him for the many useful discussions we had, all the tricks I have learned from him, and the knowledge of the 'Physics Business' he has given me. We have shared sublime moments with *Il Giardino Armonico*. Besides being a physicist of a capital caliber, he is an excellent chap, 我的老市 .

In the middle part of my research time I was pleased to meet Carmen Gordillo and her unflagging spirit. Being both fellow members of the Quantum Liquids Group gave me the opportunity to have an unreckonable number of scientific discussions. From them, and her experience in path integral Monte Carlo, I learned to perceive that field from a different scope.

The other members of the Department that have made my life much nicer and easier: Estanislau Llanta, whose perception of life came to be, in many senses, sibling of mine; Lluis Ametller, for his relentless support on dealing with the intricacies of Linux, and for reminding me how useful a Palm Pilot can be; Domingo García, with whom I share the passion for fiddling; all of them together with Rossend Rey have been my badminton mates for the last few years.

Joaquim Valls, for showing me his discernment of life through art, for still enjoying my lecture "A gentle introduction to quantum mechanics", no matter how many times he has listen to it, and mostly for being a righteous man. Thanks go to Elvira Guàrdia for both dealing with important amounts of bureaucracy while being my tutor, as well as for putting up with endless sessions of cello playing at her place. Thanks to Joaquim Trullàs who gave me the opportunity to enjoy the rewards of teaching, and with whom I had also several scientific discussions that provided me with his pragmatic point of view. To Gemma Sesé whose ever–lasting smile and good manners have solved many bureaucratic problems I have addressed to her as Head of the Section. Manel Canales's good humour has kept the moral high on this side of the aisle, however. I would also like to thank Jordi Martí for showing continued interest on my thesis, and for helping me with some references.

Ana Calle, Silvia Soriano, and Araceli Gonzalez have provided steady good humour as well as an infinite skill in dealing with the bureaucracy of UPC and elsewhere.

Thanks go to Martin Neumann and Marco Zoppi from whom I had advice on simulation timings, and the ascertainment a path integral Monte Carlo running could take that long.

Of great importance has been the fellowship I have been honoured with from the *Generalitat de Catalunya.*

Teresa is a champ. Brave too, in that our relationship has spanned much of my PhD. Her ability to tackle any situation and get the job done continues to give me clues about the way forward. An infinite supplier of companionship and confidence. A sentence, not even a carefully crafted self–referential one, can capture her influence on my life. Or can it? Well met, indeed.

My mother, who has been asking for the last four years when I was supposed to finish. Thanks to her unstinting generosity I have been able to wander off on this strange, yet wonderful, path. I keep realising every now and then aspects of myself that have been engraved by her wisdom. To my grandmother for being an amazing and continual source of inspiration. Both of them, though many miles away, relentlessly touch me with their love. Also my father, who was the first to show me the curiosity for life and science. His right measure of judgement still guides me. For you the flag is flung.

To those who come with Teresa: Risto and Julia. Their inclusion of me into their family makes things that much more solid.

And some good mates: Gigi Desogus and his family, who have hosted me many times at Trieste. And María Montoya, for her *frasis siempre huidiza de la expresión mecánica.*

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

> It is a curious historical fact that modern quantum mechanics began with two quite different mathematical formulations: the differential equation of Schroedinger, and the matrix algebra of Heisenberg. The two apparently dissimilar approaches, were proved to be mathematically equivalent. [...] a third formulation of non–relativistic quantum theory [...] was suggested by some of Dirac's remarks concerning the relation of classical action to quantum mechanics. A probability amplitude is associated with an entire motion of a particle as a function of time, rather than simply with a position of the particle at a particular time.
> This formulation is mathematically equivalent to the more usual formulations. There are, therefore, no fundamentally new results. However, there is a pleasure in recognizing old things from a new point of view.
>
> (R. P. Feynman, *Rev. Mod. Phys.*, 367, **20**, 1948)

Two score and fourteen years ago Richard Phillip Feynman devised a new formulation of quantum mechanics. At that time this formulation was conceived more as a pleasant intellectual exercise than as a truly practical new stream in quantum theory. Nevertheless, the circumstances were bound to change in a short time. Soon, as computers hove in sight, Feynman's ideas became applicable and a whole new field of knowledge within Physics was developed.

A few years after the publication of the original Feynman's paper, he wrote a book entitled "Quantum Mechanics and Path Integrals" (Feynman and Hibbs, 1965a) in which the whole theory was bounded together and made clear. A traditional way of introducing the postulates of quantum mechanics is by means of the following theoretical experiment: suppose we have an incandescent filament emitting thermal electrons in all directions. In front of the emitter there is wall with two holes, and behind the wall, a receiver. One may wonder from which of the two holes has a given electron come through. That is, which path has the electron followed. In the traditional approach to quantum mechanics the probability amplitude of finding the electron at a given position is obtained by solving a wave equation, which is the Schroedinger equation. The new approach begins postulating the path integral. In order to obtain the probability amplitude of finding the electron at a given position one must sum up the probability of the electron going from the emitter to the receiver over all possible paths. Path integrals give account of that task.

The mathematical expression of a path integral is an exponential of $i/\hbar$ times the classical action between two points. For, the path integral formulation of quantum mechanics states the relation between a quantum mechanical system and its classical counterpart through the classical action. In other words, it is possible to seize quantum mechanical information through a product of classical actions.

Even though we started our disquisition mentioning quantum mechanics, this is not our main motivation. We shall not work on quantum mechanics through path integrals, neither analytically nor numerically. *In lieu*, we shall work on quantum statistical mechanics using a technique named path integral Monte Carlo (PIMC). It is possible to formulate quantum statistical mechanics in terms of path integrals, using the inverse temperature, β, instead of $i/\hbar$, as Feynman (1998) has demonstrated. Through the following lines we shall give a brief introduction to quantum statistical mechanics and recall history.

A key ingredient of statistical mechanics is the density matrix. An essential property of the density matrix is that the convolution of two density matrices at a given temperature T yields a new density matrix at temperature $T/2$. Departing from a density matrix at a given high temperature and applying the previous property many times, it is possible to study systems at lower temperatures. At high temperatures, where the classical regime applies, is possible to neglect commutator algebra.

The form of the low temperature density matrix in terms of a product of high temperature density matrices is formally the same as a path integral. The only difference is that a path integral has no temperature, instead there is an imaginary time factor.

In the late forties quantum and statistical mechanics were reformulated in terms of path integrals. Later, during the eighties, Chandler and Wolynes (1981) introduced the isomorphism between a classical polymer ring and a quantum atom the way it is currently used in simulations. Their paper definitely settled a stream opened some years before by Fosdick and Jordan (1965) with the first path integral Monte Carlo calculation.

PIMC is devised to simulate the behaviour of systems at finite temperature which manifest quantum effects. In particular, it is used for simulating quantum liquids, albeit it is also applied to other fields such as dense matter in neutron stars (Filinov et al., 2001), and electronic structure (Bohm et al., 2002), to name just a few. The isomorphism introduced by Chandler and Wolynes states a mapping between a quantum atom and a classical ring polymer. In plain words, each atom is represented by a necklace formed by beads. Each bead is connected to its neighbours by a couple of springs. The interaction between different necklaces *i.e.* atoms, is given by the potential of the simulated system.

Using properties of the density matrix it is possible to extract physical information from a system. However, it is impossible to do so analytically provided the large number of integrals to perform and their dimensionality. In 1953 a remarkable paper written by Metropolis et al. (1953) appeared. It opened up new paths in calculation with random numbers. The Metropolis algorithm is a method for sampling any known probability distribution. Since the Metropolis algorithm deals with random numbers it is classified as a Monte Carlo technique.

Now, it is convenient to define path integral Monte Carlo as a method that uses the isomorphism between a quantum atom and a classical polymer ring in order to calculate physical properties of quantum liquids at finite temperature, by means of Monte Carlo techniques. This is not the most general definition of PIMC, but for the purposes of this thesis it is perhaps the most useful.

Up until now we have described how to represent quantum atoms in a simulation—in terms of path integrals—and how to make that simulation evolve—by means of Monte Carlo methods. The easiest way to evolve the simulation is using a bead per bead sampling. A bead of a given atom is chosen at random and a random displacement from its initial position is proposed. Using a probabilistic law, given by the choice of action, the movement is accepted or rejected. Sufficiently large iteration of this process will drive the system to its equilibrium state, in which that probability distribution is effectively sampled. After reaching equilibrium, state properties of the system (such as energy, one–body density matrix, or the radial distribution function) can be estimated with further iteration. Note that in this process there is no time variable involved, thus dynamical properties are not accessible.

Any PIMC code encompasses three main parts: action, sampling and properties. The classical action, that is, the one that has only classical terms, is usually called primitive action. However, there are more elaborated actions. Li–Broughton's formulation is an expansion in $\hbar^2$, thus including first–order quantum contributions (Li and Broughton, 1987). Another approach is to determine the exact action for two atoms and then to construct a many–body action from that starting point. This idea was first proposed by Barker (1979) and then fully developed by Pollock and Ceperley (1984)

A naive sampling—the way beads are moved—has been explained already. Nevertheless, there are other ways of moving beads, such as staging (Sprik et al., 1985) or bisection (Ceperley, 1995a) methods. These methods perform smart collective displacements of beads, and better benchmarks are found.

Energy is one of the most demanding properties that can be estimated from a PIMC simulation. The thermodynamic estimator of the energy is obtained using the partition function. Since that estimator yields a large variance, the virial estimator—derived using the virial theorem—was proposed (Herman et al., 1982). It is important to be aware that energy estimators are cunningly interwoven into the action, hence coding a given energy estimator depends on the action chosen. Nevertheless, energy is just one physical property. There are many others also reachable with PIMC, and different methods for each property.

In spite of how complex coding an efficient PIMC program from the description of the problem given so far may seem, we have only addressed quantum liquids at finite temperature from a single point of view. In order to show a more comprehensive perspective of simulations of quantum liquids, we shall briefly describe other approaches.

We have already mentioned that dynamical properties are unreachable through PIMC simulations. Notwithstanding, this problem has been partially surmounted using the centroid theory (Voth, 1996). The centroid approximation is semi–classical, that is, it is well defined within the classical limit approximation. However, it has not been demonstrated that it holds for $T \rightarrow 0$. It assumes the geometric centre of the polymer ring to be the centre of mass of the atom. Once the atom is reduced to a point–like particle, classical dynamics can be performed, assuming that this point evolves via an effective potential.

Monte Carlo techniques are not the only solution for a path integral problem; Newton laws may carry out the same job, yielding the path integral molecular dynamics (PIMD) method (Tuckerman et al., 1993). However, there is a *caveat*: in real systems with a reasonable amount of atoms PIMD becomes useless due to extremely large CPU times. Nevertheless, this problem can be surmounted connecting each bead to a thermostat (Nosè–Hoover chain).

There are even more ways to tackle a PIMC problem. For instance, using terms of a Fourier expansion instead of beads yields the Fourier path integral Monte Carlo method (FPIMC) (Freeman and Doll, 1984). FPIMC represents an improvement on the primitive action, however sampling methods and energy estimators used within FPIMC are the simplest ones, since this kind of algorithms for Fourier space are more difficult. Furthermore, bead exchanges, due to bosonic effects, have not been tried in a FPIMC code (Chakravarty et al., 1998).

The list of different methods and their noteworthy aspects, as well as their unvirtuous attributes, is almost endless and could left us bewildered.

An immediate conclusion from this abridged view of PIMC and related issues is that there are many ways to solve a given problem. For any particular aspect of a path integral simulation there are normally several methods that can be used. However, there are few comparisons available amongst them. Furthermore, some of the available literature regarding methods and algorithms lacks of clarity. Most of the codes available within the scientific community are straight forward approaches that use primitive actions along with a bead per bead sampling. It is worth mentioning the code developed by David M. Ceperley. Nowadays it is highly optimised and amply equipped with add–ons.

The main goal of this thesis was to code a PIMC program. At the moment we have built a library of PIMC codes. Throughout this coding process we have tested and compared many of the methods available. As a result, the codes are optimised for our purposes. We have given detailed algorithms and have told the inside story of the codes. We have also developed new approaches to the energy estimation and the momentum distribution problems (Brualla et al., 2002). For the testing process we have used mainly the harmonic oscillator—as a toy model— and liquid Ne and ^{4}He as real systems. Besides reproducing already known results, there is also a part of the thesis that deals with new results not previously exposed in literature. Up until now the departure from the classical regime of liquid helium and neon has not been fully studied from the momentum distribution point of view. Taking advantage of a new algorithm we have developed for momentum distributions—the *trail* method—we coped with that problem. Finally, we have faced the neon solid–liquid transition studying the behaviour of the energy in terms of the density at a given temperature. Special attention has deserved the possible discontinuity of the kinetic energy at the transition. This point has been studied by other authors, both experimentally and theoretically, in neon, helium, and hydrogen without arriving to a clear conclusion.

In deciding the order of presenting the work done during these years, we faced the dilemma of whether making a historical exposition, following a chronological order of our research of methods and results, or on the other hand, grouping the information in areas of knowledge. We opted for the latter choice, since, from our point of view, it provides a stepwise and more pedagogical approach to PIMC techniques. Therefore, this thesis has been divided in the following way: chapter 2 provides the fundamentals of path integrals, from the quantum mechanical scope, and an introduction to the key features of the density matrix. Towards the end of the chapter we bridge the gap between quantum and statistical mechanics, laying the ground for the path integral Monte Carlo formulation.

From chapter 3 to 5 we introduce the theoretical aspects of PIMC.

Chapter 3 is devoted to the action; we study several actions applied to the one–dimensional harmonic oscillator, and to real systems. In chapter 4 we introduce several sampling methods. PIMC theoretical chapters finish with the one devoted to averaged properties that can be estimated through simulations. It is mainly devoted to the energy and to the momentum distribution. For the former property we keep the same order established for the harmonic oscillator and real systems while studying the action. For the latter we introduce the theoretical aspects of the *trail* method, as well as, those belonging to the other two currently used approaches to the momentum distribution. Throughout these chapters we shall give useful formulæ for coding as well as hints for programming. All of them will be enhanced throughout the appendixes, where lengthy derivations, some codes, and programming tips may be found.

As part of the new results provided by our research, we have made extensive use of the Richardson extrapolation within PIMC. As far as we know, this kind of extrapolation has not been applied to this field. Chapter 6 introduces the theoretical aspects of the Richardson extrapolation, its applications to PIMC results, and a small code for implementing a recursive extrapolation.

Chapter 7 shows the outcomes of the tests we have performed on several systems. We have chosen systems with well known properties, in order to have useful comparisons with the literature. We shall test the accuracy of our codes, as well as, benchmarks of some of the different methods exposed up to that point.

Finally, we shall apply the codes we have developed during the thesis, on problems that represent current open streams of research, hence providing new data to the scientific literature. We have chosen two different problems, briefly sketched above. Chapter 8 deals with the momentum distributions of liquid helium and neon, and chapter 9 is a study of the solid–liquid phase transition of neon at 35 K. Chapter 10 is devoted to conclusions and possible new lines of research.

We hope to give a clear view of PIMC methods along with easy to code recipes that would provide future researchers with appropriate tools for studying quantum liquids at finite temperature in the Boltzmann regime.

2. FUNDAMENTALS

We aim to develop throughout this chapter the fundamental ideas necessary for the path integral Monte Carlo method. In order to do so neatly we shall first give the physical conditions of the problems we would like to solve once we have our PIMC code. We shall simulate three dimensional liquids within a box. The size of the box, as well as the number of particles and the temperature, will be fixed. Thus we work in the canonical ensemble. Furthermore, we will simulate quantum fluids at a low finite temperature but not superfluids. In the range of temperatures chosen, exchanges can be disregarded and the systems can be considered Boltzmann quantum liquids.

In this chapter we shall first present the path integral formalism. Then we shall briefly introduce the thermal density matrix along with some of its properties. Finally, we shall show the parallelism between path integrals and density matrices.

2.1 Path integrals

Parallel to Schrödinger's formulation of quantum mechanics there is another formalism which departs from the principle of least action and arrives to the same conclusions but from a different scope. That formulation was firstly introduced by Feynman (1948) with the name of *path integral.* For the sake of simplicity we shall develop the theory in one dimension. Nevertheless, the extension to three dimensions can be easily done.

2.1.1 Classical action

Let us study a particle travelling in a one dimensional space. It departs from point x_1 at time t_1 and arrives to point x_2 at time t_2. The path actually described by the particle $\bar{x}(t)$, out of all the possible paths is obtained minimising a quantity S given by

$$S \equiv \int_{t_1}^{t_2} dt\, \mathcal{L}(x(t), \dot{x}(t); t) \tag{2.1}$$

S is the classical action and $\mathcal{L}$ is the Lagrangian of the system. For a particle of mass m in a potential $V(x)$, the Lagrangian is

$$\mathcal{L} = \frac{m}{2}\dot{x}(t)^2 - V(x(t); t) \tag{2.2}$$

In classical mechanics only the extreme path contributes to the probability amplitude of finding the particle at a given position. The neighbouring paths are necessary just to determine, by means of variational calculus, the path actually followed by the particle.

2.1.2 The path integral

In quantum mechanics, on the other hand, not just the extreme path contributes to the probability amplitude, but all the possible paths. They all contribute to the total amplitude, every path being weighted with a different phase, given by the action.

The previous statements could be summarised in the following manner. The probability $P(2,1)$ of a particle to go from the point x_1 at time t_1 to a point x_2 at a time t_2 is $P(2,1) = |K(2,1)|^2$, where the amplitude

$$|K(2,1)| = \sum_{\substack{\text{over} \quad \text{all} \\ \text{possible} \\ \text{paths}}} \phi[x(t)] \tag{2.3}$$

is given in terms of the contribution $\phi[x(t)]$ from each path, being

$$\phi[x(t)] = A \exp\left\{\frac{i}{\hbar} S[x(t)]\right\}, \tag{2.4}$$

where A is a normalisation constant and the action is that of the corresponding classical system.

Equation (2.3) is a Riemann integral going to the continuum. Thus, we might slice $\phi[x(t)]$ up into many intervals. The more slices we perform the closest is the approximation to the exact value of the integral. Each slice will be of width τ. Now, taking the limit where $\tau \to 0$ in (2.3) we get

$$K(2,1) = \lim_{\tau\to 0} \frac{1}{A} \int \cdots \int \exp\left(\frac{i}{\hbar} S[2,1]\right) \frac{dx_1}{A} \cdots \frac{dx_{M-1}}{A}, \tag{2.5}$$

where M is the number of slices of length τ, that is, $\tau = |x_{i+1} - x_i|$, and $A = \left(\frac{2\pi i\hbar\tau}{m}\right)^{1/2}$ in case of using the Lagrangian given in equation (2.2). Expression (2.5) is the definition of the sum over all paths. We could express that in a more suitable way as

$$K(2,1) = \int_1^2 \exp\left(\frac{i}{\hbar} S[2,1]\right) \mathcal{D}x(t), \tag{2.6}$$

which is indeed the *Feynman's path integral formula.* $\mathcal{D}x(t)$ is the product of the $M-1$ differentials. We do not pretend a rigorous formulation of the path integral formalism. At a first glance it could seem clumsy to define an integral over all possible paths; this is a high order of infinity and it is not unfeigning to realise that the integral is bounded. Howbeit, what we intend is to give an intuitive idea of the path integral formulation of quantum mechanics, since our main interest is to do quantum statistical mechanics.

Inasmuch as our goal is to produce an algorithm to be run in a computer, we are interested in the translation of path integral formulæ from continuum space into discretised space. It is possible to discretise equation (2.1) taking M slices of size τ, yielding the following summation:

$$S^M = \tau \sum_{j=1}^{M} \left(\frac{m}{2} \left(\frac{x_j - x_{j-1}}{\tau} \right)^2 - V(x_j) \right). \tag{2.7}$$

That approximation to the action is good to second order, as long as we use $V(\frac{x_j + x_{j-1}}{2})$ instead of $V(x_j)$. Equation (2.7) is the discretised action we shall use when coding the PIMC algorithm.

2.1.3 From path integrals to Schrödinger equation

As we said before, the path integral theory is a formulation of quantum mechanics parallel to the well known Schrödinger formulation. Therefore, it is possible to depart from path integrals and retrieve the Schrödinger equation. That is exactly what we are about to do. Moreover, we will demonstrate the normalisation constant appearing in (2.5).

Using the propagator,

$$\psi(x_2,t_2) = \int_{-\infty}^{\infty} K(x_2,t_2;x_1,t_1)\psi(x_1,t_1)\,dx_1, \tag{2.8}$$

we get the wave function ψ at time t_2. It is possible to calculate the time t_2 as a small increment τ of t_1. If we do so, we obtain

$$\psi(x_i,t+\tau) = \int_{-\infty}^{\infty} \frac{1}{A}\exp\left(\tau\frac{i}{\hbar}\mathcal{L}\left(\frac{x_i-x_{i-1}}{\tau},x_i\right)\right)\psi(x_{i-1},t)\,dx_{i-1} \tag{2.9}$$

where A is the normalisation constant to be determined, and $\mathcal{L} = m\dot{x}^2/2 - V(x,t)$. Notice that $\dot{x}$ has been discretised using the small increment τ. Substitution of the Lagrangian in the integral yields

$$\begin{aligned}\psi(x_i,t+\tau) &= \int_{-\infty}^{\infty}\frac{1}{A}\left\{\exp\left[\frac{i}{\hbar}\frac{m(x_i-x_{i-1})^2}{\tau}\right]\right\}\times\\ &\quad\times\left\{\exp\left[-\frac{i}{\hbar}\tau V(x_i,t)\right]\right\}\psi(x_{i-1},t)\,dx_{i-1}.\end{aligned} \tag{2.10}$$

We wish to evaluate that integral for small changes in $(x_i-x_{i-1})^2$. Thus, we substitute $x_{i-1} = x_i + \eta$ in equation (2.10) taking η small enough in preparation for a series expansion:

$$\psi(x_i,t+\tau) = \int_{-\infty}^{\infty}\frac{1}{A}\exp\left[\frac{im\eta^2}{2\hbar\tau}\right]\exp\left[-\frac{i\tau}{\hbar}V(x_i,t)\right]\psi(x_i+\eta,t)\,d\eta. \tag{2.11}$$

Now we expand equation (2.11) in power series to first order in τ

$$\begin{aligned}\psi(x_i,t+\tau)+\tau\frac{\partial}{\partial t}\psi(x_i,t+\tau) &= \int_{-\infty}^{\infty}\frac{1}{A}\exp\left[\frac{im\eta^2}{2\hbar\tau}\right]\left[1-\frac{i\tau}{\hbar}V(x_i,t)\right]\times\\ \times\left[\psi(x_i,t+\tau)+\eta\frac{\partial}{\partial x_i}\psi(x_i,t+\tau)+\frac{1}{2}\eta^2\frac{\partial^2}{\partial {x_i}^2}\psi(x_i,t+\tau)\right]d\eta.\end{aligned} \tag{2.12}$$

If in (2.12) we compare left and right hand sides at the same order in τ, we find the following relations: at order zero

$$\frac{1}{A}\int_{-\infty}^{\infty}\exp\left[\frac{im\eta^2}{2\hbar\tau}\right]d\eta = 1, \tag{2.13}$$

which yields the normalisation constant

$$A = \left(\frac{2\pi i\hbar\tau}{m}\right)^{\frac{1}{2}}. \tag{2.14}$$

And at first order we find

$$\psi + \tau\frac{\partial\psi}{\partial t} = \psi - \frac{i\tau}{\hbar}V\psi - \frac{\hbar\tau}{2im}\frac{\partial^2\psi}{\partial x^2}. \tag{2.15}$$

This holds true to order τ only if ψ satisfies

$$-\frac{\hbar}{i}\frac{\partial\psi}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\psi}{\partial x^2} + V(x,t)\psi \tag{2.16}$$

which is the Schrödinger equation.

2.1.4 Exact solution of the free particle problem

The strength of PIMC roots in the fact that low temperature properties can be calculated performing products of high temperature density matrices. Although, at the moment, we are just working with quantum path integrals in Feynman's sense, that is, at zero temperature, we can still see how the product of classical actions yields the quantum result. In order to do so, we shall calculate the propagator of a quantum free particle by means of path integrals, which is fully analytical.

The classical path of a free particle from $(x_0, 0)$ to (x_M, t) is given by $x(t) = x_0 + \frac{t}{\tau}(x_N - x_0)$ and the classical action is

$$S_{\rm Cl} = \int_0^\tau \frac{m}{2}\left(\frac{x_M - x_0}{\tau}\right)^2 dt = \frac{m}{2}\frac{(x_M - x_0)^2}{\tau}. \tag{2.17}$$

On the other hand, in the path integral sense, we wish to evaluate

$$\begin{aligned} K(x_M, x_0; \tau) &= \int_{x_0}^{x_M} dx(t)\, \exp\left(\frac{i}{\hbar}\int_0^\tau \frac{m}{2}\dot{x}^2\, dt\right) && (2.18)\\ &= \lim_{M\to\infty}\left(\frac{m}{2\pi i\hbar\tau}\right)^{(M+1)/2} \times \\ &\quad \times \int \mathcal{D}x\, \exp\left(\frac{im}{2\hbar\tau}\sum_{j=0}^{M}(x_{j+1} - x_j)^2\right) && (2.19)\end{aligned}$$

with $\tau = t/(M+1)$, $\mathcal{D}x = dx_1 \cdots dx_M$ and $x_{M+1} = x_0$. The normalisation constant appearing in round parentheses has been justified in section (§ 2.1.3). This is again a Fresnel integral, producing

$$K(x_M, x_0; \tau) = \sqrt{\frac{m}{2\pi i\hbar\tau}} \exp\left(\frac{i}{\hbar}\frac{m}{2}\frac{(x_M - x_0)^2}{\tau}\right). \tag{2.20}$$

Notice that the phase in (2.20) is the classical action from (2.17) times a factor $i/\hbar$. This is exactly what we expected from theory. We have seen in this example how the quantum propagator is given in terms of a product of classical actions. For the free particle the quantum mechanical result of the amplitude in terms of the classical action is exact. This is not generally true in other cases since $V(x)$ is not zero. When we integrate the Lagrangian, in order to get the action, what we are integrating indeed is the kinetic energy operator plus the potential energy operator—due to the Baker–Campbell–Hausdorff (2.36) formula and the linearity of the integral. For standard Hamiltonians—those with the kinetic term equal to $p^2/2m$—the result of the kinetic part of the action is always the same. In fact we shall see when studying sampling methods, that some methods take advantage of that fact and solve analytically the kinetic part and the remaining part of the action *i.e.* the potential term, is the only one sampled by means of the Metropolis algorithm. For, it is obvious that problems arise when doing $\int V(x,t)\,dt$. Even in stationary potentials, those with no explicit time dependence $V(x,t) = V(x)$, there are problems, since there is an implicit dependence of $x(t)$ on time. A first approximation is to neglect the dependence of position on time, which yields

$$\int_0^\tau V(x)\,dt = \tau V(x). \tag{2.21}$$

2.2 Density matrix

We have already mentioned that we shall develop our codes in the canonical ensemble. Therefore, all the theory from quantum statistical mechanics shown in this section will be outlined within that ensemble. In the canonical ensemble a state is defined by the fixed quantities N (number of particles), V (volume) and T (temperature), and the variable quantity E (total energy). When we speak of number of particles we refer to number of atoms, that is in the PIMC language, number of polymer rings or necklaces.

The probability that a system, chosen at random from the ensemble, has an energy E_i is given by the Boltzmann statistics $\exp(-\beta E_i)$, where $\beta = 1/kT$ with k the Boltzmann's constant. We must not forget that we are going to work in the Boltzmann quantum regime, that is, at a temperature sufficiently low for quantum effects to arise, but not low enough to incur in indistinguishability effects. Thus, Boltzmann statistics suffices.

Our ensemble is characterised by the Hamiltonian operator H whose eigenfunctions are ϕ_i and the eigenvalues are E_i. In the position–space representation, the density matrix is defined as

$$\rho(R, R'; \beta) =< R|e^{-\beta H}|R' >= \sum_i \phi_i^*(R)\phi_i(R')e^{-\beta E_i}, \tag{2.22}$$

where $R = \mathbf{r}_1, \ldots, \mathbf{r}_N$. The occupation probability of a given state i is $e^{-E_i/k_\mathrm{B}T}$.

The expectation value of an operator O in discretised space is

$$< O >= Z^{-1} \sum_i < \phi_i|O|\phi_i > e^{-\beta E_i}, \tag{2.23}$$

with

$$Z = \sum_i e^{-\beta E_i}, \tag{2.24}$$

the partition function.

In continuum space, summations in equations (2.23) and (2.24) become integrals as

$$< O >= Z^{-1} \int dRdR' \rho(R, R'; \beta) < R|O|R' >, \tag{2.25}$$

$$Z = \int dR \rho(R, R; \beta). \tag{2.26}$$

2.2.1 Properties of the density matrix

The following exact property of the density matrix is the basis of the PIMC method. As a matter of fact that property shows the reason why PIMC—a method from quantum statistical mechanics—is formally identical to the path integral development in quantum mechanics.

The convolution of two density matrices is a density matrix at a lower temperature. In order to derive this property, we depart from the definition of the density matrix,

$$\rho(R_1, R_3; \beta) =< R_1|e^{-\beta H}|R_3 > . \tag{2.27}$$

We also know that $\beta = \beta_1 + \beta_2$. Substitution of this into the definition of the density matrix leads to

$$\rho(R_1, R_3; \beta) =< R_1|e^{-\beta_1 H}e^{-\beta_2 H}|R_3 > . \tag{2.28}$$

Using the identity operator on the right hand side of (2.28) yields

$$\rho(R_1, R_3; \beta_1 + \beta_2) = \int dR_2 \rho(R_1, R_2; \beta_1)\rho(R_2, R_3; \beta_2), \tag{2.29}$$

which is the product property of the density matrix.

The product property can be applied iteratively many times. In order to do so we shall define first the time step τ as:

$$\tau \equiv \beta/M \;\Rightarrow\; e^{-\beta H} = \left(e^{-\tau H}\right)^M . \tag{2.30}$$

Now, taking the product M times,

$$\begin{aligned}\rho(R_0, R_M; \beta) \;\; = \;\; & \int \cdots \int dR_1 dR_2 \cdots dR_{M-1} \times \\ & \rho(R_0, R_1; \tau)\rho(R_1, R_2; \tau) \cdots \rho(R_{M-1}, R_M; \tau)\end{aligned} \tag{2.31}$$

This result is exact for any $M \geq 1$. M is the number of time slices, that is, in our polymer language, is the number of beads of a given atom. Since $\beta \equiv 1/T$ (from now on we shall work in reduced units, where the Boltzmann constant equals to 1) then the density matrices at the right hand side of equation (2.31) are at a higher temperature (MT),than $\rho(R_0, R_M; \beta)$. For, $\rho(R_0, R_1; \tau)$, $\rho(R_1, R_2; \tau)$, $\ldots$, $\rho(R_{M-1}, R_M; \tau)$, should they are at a sufficiently high temperature, they can be considered as classical density matrices. Thereof the parallelism with the path integral formulation of quantum mechanics, where quantum results are obtained via a product of classical actions.

The density matrix is non–negative for all values of its arguments (R, R' and β), as long as we are dealing with Boltzmannons or Bosons. Thus, it can be interpreted as a probability distribution and sampled.

2.2.2 The Trotter formula

Let us assume we are dealing with a standard Hamiltonian of the form

$$H(x,p,t) = T(p,t) + V(x) = \frac{p^2}{2m} + V(x), \tag{2.32}$$

with $V(x)$ sufficiently smooth and lower bounded. The time displacement operator is given by

$$e^{-\tau H/\hbar} = e^{-\tau(T+V)/\hbar}. \tag{2.33}$$

with H, T and V operators. According to the Baker–Campbell–Hausdorff formula, equation (2.33) can be factorised as

$$e^{-\tau(T+V)/\hbar} = e^{-\tau V/\hbar} e^{-\tau T/\hbar} e^{-\tau^2 C/\hbar^2}, \tag{2.34}$$

where the operator C is

$$C \equiv \frac{i}{2}[V,T] - \frac{\tau}{\hbar}\left(\frac{1}{6}[V,[V,T]] - \frac{1}{3}[[V,T],T]\right) + \dots. \tag{2.35}$$

Therefore to second order in τ

$$e^{-\tau(T+V)/\hbar} \approx e^{-\tau T/\hbar} e^{-\tau V/\hbar}. \tag{2.36}$$

In accordance to equation (2.36) we can approximate the exact density matrix in H to the product of density matrices of T and V. Nevertheless, it is not guaranteed *a priori* that repeated application of this process does not lead to a biased result due to cumulative errors. In order to be sure of the exactness of our results we recall the Trotter formula:

$$e^{-\beta(T+V)} = \lim_{M\to\infty} \left[e^{-\tau T} e^{-\tau V}\right]^M. \tag{2.37}$$

The derivation of the Trotter formula is too technical and out of the scope of this thesis (Trotter, 1958). Nevertheless, there are several conditions that the functions involved in that formula must satisfy in order to yield valid results. Although we are using the Trotter formula for the operators T and V, it holds for any two pair of operators, provided the operators are lower bounded. We do not have to worry, since the potentials we shall use are bounded from below. The Trotter formula also requires that the three operators T, V, and $T+V$ be self–adjoint. Nevertheless the Hamiltonians we use for helium and neon also satisfy this condition.

2.2.3 The relation between path integrals and the density matrix

In the introduction we talked about the language around PIMC methods. We said that sometimes the beads are referred as particles in imaginary time. Now we are going to show the relationship between temperature and time in the path integral sense.

In aid of clarity we shall study a single particle in one dimension whose Hamiltonian in space representation is

$$\hat{H} = \frac{\hbar^2}{2m}\frac{\partial^2}{\partial R^2} + V(R). \tag{2.38}$$

In order to make evident the relationship between temperature and time we first derive equation (2.22), therefore we have

$$\frac{\partial}{\partial \beta}\rho(R, R'; \beta) = -\sum_i E_i \phi_i^*(R)\phi_i(R')e^{-\beta E_i}, \tag{2.39}$$

yielding the Bloch equation

$$\frac{\partial}{\partial \beta}\rho(R, R'; \beta) = -\left(-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial R^2} + V(R)\right)\rho(R, R'; \beta). \tag{2.40}$$

On the other hand we have the Schrödinger equation for a wave function $\psi(x, t)$, which is:

$$i\hbar\frac{\partial}{\partial t}\psi(x, t) = \left(-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x)\right)\psi(x, t). \tag{2.41}$$

We also know that the time evolution of the wave function $\psi(x, t)$ is obtained by means of the propagator given by equation (2.8), that is

$$\psi(x, t) = \int dx' K(x, x'; t)\psi(x', 0). \tag{2.42}$$

The propagator satisfies the Schrödinger equation

$$i\hbar\frac{\partial}{\partial t}K(x, x'; t) = \left(-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x)\right)K(x, x'; t), \tag{2.43}$$

along with the boundary condition

$$K(x, x'; 0) = \delta(x - x'). \tag{2.44}$$

If we take $t = -i\beta\hbar$ and $R = x$, then the equivalence between equations (2.40) and (2.41) is evident. In other words, the density matrix is the propagator for evolution in imaginary time. In fact:

$$\rho(R, R'; \beta) = K(x, x'; -i\beta\hbar). \tag{2.45}$$

Expression (2.45) is a formal equivalence, instead of an equation. Nevertheless, it is useful for observing the connection between quantum mechanics and quantum statistical mechanics.

2.2.4 The harmonic oscillator—semper fidelis

The harmonic oscillator is a simple model that we shall use in order to test our algorithms. It is very useful since the energy, and the density matrix, can be calculated analytically. Later on, we shall compare our results of the energy from the PIMC code with the exact result obtained here.

The Hamiltonian of the linear harmonic oscillator is given by

$$\hat{H} = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + \frac{1}{2}m\omega^2 x^2 \tag{2.46}$$

and the trace is

$$\int_{-\infty}^{\infty} \langle x|e^{-\beta\hat{H}}|x\rangle = \frac{e^{-(1/2)\beta\hbar\omega}}{1 - e^{-\beta\hbar\omega}}, \tag{2.47}$$

which is indeed the partition function. From the definition of the mean energy in terms of the trace we know that

$$\langle H\rangle = -\frac{\partial}{\partial\beta}\ln\sum_x \langle x|e^{-\beta\hat{H}}|x\rangle. \tag{2.48}$$

Therefore, after some algebra. we finally get the expression for the mean energy of the linear harmonic oscillator

$$\langle H\rangle = \frac{1}{2}\hbar\omega\coth\left(\frac{1}{2}\beta\hbar\omega\right). \tag{2.49}$$

In chapter § 7 we shall compare it with the mean energy estimated by means of PIMC.

3. ACTION

There are three main ingredients in any PIMC code: action, sampling, and properties. Along this chapter we shall study the action. We shall not directly deal with the quantum many body problem stated above, instead, we shall start with a very well known model problem: the quantum harmonic oscillator. It will give us the opportunity to test different approaches to the action in a PIMC code, and learn how to implement the algorithms in a computer. Therefore, this chapter is divided in two main sections, the first one devoted to the actions for a one dimensional harmonic oscillator, and the second one related to actions of three dimensional real systems.

During the study of the action in the harmonic oscillator we shall test as much as five different actions: the primitive action, three slight improvements over the primitive action, and finally the Li–Broughton action. We shall realise that only the Li–Broughton action represents a real improvement over the primitive, while the other actions studied do not provide substantial betterment.

In order to sample the probability distributions required in a PIMC algorithm we shall use the Metropolis method. The Metropolis method (Metropolis et al., 1953) generates points that are serially correlated. Long correlation times, together with the many degrees of freedom (around 20,000 spatial variables in any real system simulation), yield noisy results when dealing with quantum fluids. Thus, we shall only derive the primitive and the Li–Broughton action (Li and Broughton, 1987) for the real systems, since we will show in chapter 7 that the little improvements observed in the three variants of the primitive in the harmonic oscillator, will not be perceptible in the real system.

In aid of clarity, in this chapter we shall just derive the formulæ related to the actions. We shall leave for the following chapters the algorithms, as well as, the results obtained with the actions found. In the chapter dedicated to the results we will validate the choice of actions taken for studying real systems.

3.1 General considerations

Prior to finding the different actions for both the harmonic oscillator and three dimensional real systems we shall derive some results that will provide us with useful formulæ, as well as a deeper physical insight. For the sake of clarity, and without loss of generality, we shall derive these results for a single particle in a stationary potential.

3.1.1 The primitive action

We have seen in § 2.2.1 how a large number of time slices M implies a path made up of high temperature density matrices. Taking this into account we might approximate the partition function

$$\begin{aligned} Z &= \int dx_1 \cdots dx_M \qquad (3.1) \\ &\times \rho(x_1,x_2;\beta/M)\rho(x_2,x_3;\beta/M)\cdots\rho(x_{M-1},x_M;\beta/M)\rho(x_M,x_1;\beta/M) \end{aligned}$$

for a large number of beads M. In such case we might approximate $V(x)=0$ and, therefore, the Bloch equation (2.40) becomes

$$\frac{\partial}{\partial\beta}\rho(x,x';\beta) = \frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2}\rho(x,x';\beta). \qquad (3.2)$$

Considering (3.2) as a diffusion equation, where β plays the role of time and $\hbar^2/2m$ is the diffusion coefficient. All of the elements of the density matrix $\rho(x,x';\beta)$ are positive–defined and it behaves as a probability density, thus it can be sampled. Returning to the diffusion equation analogy, $\rho(x,x';\beta)$ comports as the probability of finding the diffusing particles at position x provided at $\beta=0$ they were all concentrated at $x=x'$. We shall return to that interpretation of finding all the particles concentrated in one point when discussing the polymer isomorphism. The solution to the diffusion equation is a Gaussian that spreads with time. Thus, the solution to (3.2) is

$$\rho(x,x';\beta) = \left(\frac{m}{2\pi\beta\hbar^2}\right)^{1/2} \exp\left[-\frac{m}{2\beta\hbar^2}(x-x')^2\right]. \tag{3.3}$$

So far the result found is exact. We can still go a little bit further and consider a constant potential V. In that case the solution is still exact and given by

$$\rho(x,x';\beta) = \left(\frac{m}{2\pi\beta\hbar^2}\right)^{1/2} \exp\left[-\frac{m}{2\beta\hbar^2}(x-x')^2 - \beta V\right]. \tag{3.4}$$

Notice that as x and x' get apart from each other, and with β small, the first term of the integral dominates over βV. So, we might consider, even in the case of a potential of the form $V(x)$, the potential constant when the particle propagates from x to x'. The approximation for small β (large M) results in:

$$\rho(x,x';\beta/M) \approx \left(\frac{mM}{2\pi\beta\hbar^2}\right)^{1/2} \exp\left[-\frac{mM}{2\beta\hbar^2}(x-x')^2 - \frac{\beta}{M}V(x)\right]. \tag{3.5}$$

The term in the exponential is the primitive approximation to the action. We have considered only a one dimensional particle; however, this result is easily extendable to three dimensions, and we shall do so in the section devoted to real systems, § 3.3.

3.1.2 The polymer isomorphism

We are now in the position to discuss the polymer isomorphism (Chandler and Wolynes, 1981) presented in the introduction (§1). From equations (3.1) and (3.5) we infer that the approximate partition function is

$$\begin{aligned} Z \quad &\approx \quad Z_M = \left(\frac{mM}{2\pi\beta\hbar^2}\right)^{M/2} \int dx_1 \cdots dx_M \qquad (3.6)\\ &\times \quad \exp\left\{-\beta\sum_{\alpha=1}^{M}\left[\frac{mM}{2\beta^2\hbar^2}(x_{\alpha+1}-x_\alpha)^2 + \frac{1}{M}V(x_\alpha)\right]\right\}. \end{aligned}$$

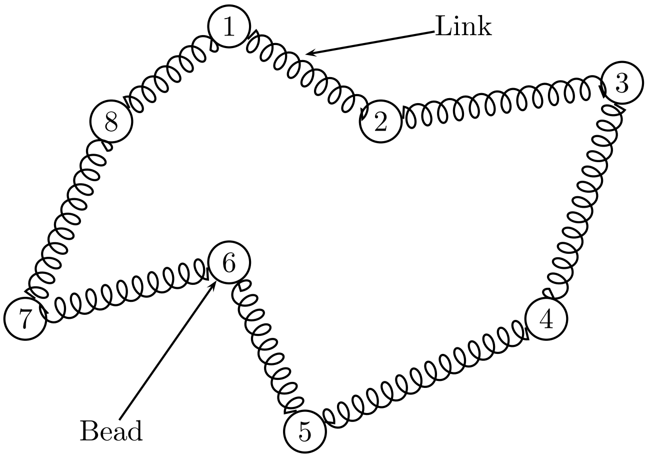


Fig. 3.1: Intuitive picture of the polymer isomorphism

The previous approximation is exact in the case of an infinite number of beads (Feynman, 1948), (Trotter, 1958), that is,

$$Z = \lim_{M \to \infty} Z_M. \tag{3.7}$$

The argument of the exponential is made up of two terms, where the kinetic part is fully separated from the potential part. The kinetic part, which is the spring term, derives from the kinetic energy operator in the Hamiltonian. This form of an interacting potential term plus a spring term, is the same that has been used for modelling polymers Binder (1994). An intuitive picture of the polymer isomorphism can be seen in figure 3.1.2. The coils represent the kinetic term of the action $(x_{\alpha+1} - x_\alpha)^2$, while the beads represent the number of high T density matrices.

3.1.3 The polymer isomorphism seen from the scope of temperature

There are two principal advantages of the polymer isomorphism in PIMC. Firstly, from the numerical calculation point of view: there are many numerical techniques developed within the reference frame of polymer physics that can be applied to quantum liquids through the isomorphism. Secondly, it help us to visualise quantum systems. Perhaps most of the troubles people face when dealing for the first time with quantum mechanics come from the fact that we live in a classical world, and the quantum reality goes against our intuition. In that moment it becomes imperative to develop a new kind of intuition. Nevertheless, it is not easy to do so, since every learning process has to be deeply rooted in a previous known experience, and there is no evident connection between the quantum and the classical world. Path integrals, along with the polymer isomorphism can help to bridge that gap, and offer the opportunity to make a mental mapping of the quantum reality throughout classical mechanics. The trick is to represent one quantum atom by means of a necklace of many classical particles (Fig. 3.1.2). When the atom is at high temperature, the corresponding classical necklace shrinks and becomes a collection of many classical particles one upon each other. In that situation we may say that the atom is localised and distinguishable, and behaves as a classical particle. As the temperature descends, the polymer ring begins to swell occupying a volume of the position space. In that case, the atom begins to delocalise (Fig. 3.2). It is no further possible to give account of a precise position of the atom, instead, there is a a density probability of finding the atom in the space. The atom is delocalised, but still distinguishable, since the polymer ring is well defined, and cannot be confused with another neighbouring necklace. We say that we are dealing with Boltzmannons. The picture on the cover is a snapshot of the positions of the beads taken during a ^{4}He simulation at 5 K. There are 8 beads per atom. From that picture it can be seen how each polymer ring occupies a region of the space.

Now, in order to go to the farthest consequences of the isomorphism, let us assume we are dealing with a collection of ^{4}He atoms, and let us also assume we are going through the lambda transition, that is, we go into the superfluidity regime. Superfluidity can be translated into the classical polymer language in the following terms: there is a probability of bead exchanging between atoms. That probability is very low above the lambda transition, but is not negligible below it. Therefore, below the lambda transition temperature the atoms begin to exchange beads and the result is that some chains become entangled. Now, the entan-

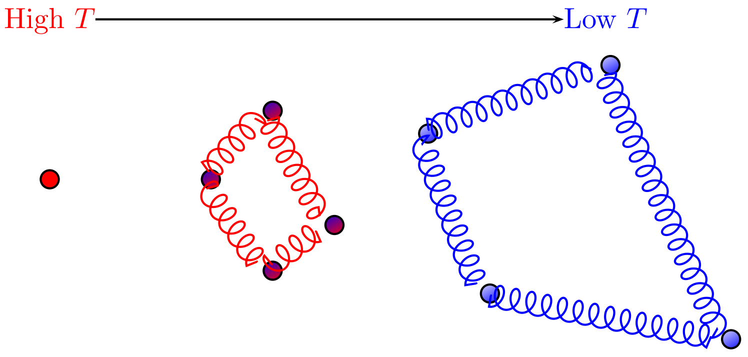


Fig. 3.2: Behaviour of the polymer ring as the temperature descends

gled chains are no longer distinguishable since they have “melted” their constituting beads with beads from other atoms.

The polymer rings and their relation with temperature can be put in a formal way. The easiest analytical case to consider is the free particle. Let us define the root mean square radius of gyration Δ as

$$\Delta^2 \equiv \frac{1}{M}\langle\sum_{\alpha=1}^{M}(\Delta x_\alpha)^2\rangle, \tag{3.8}$$

being

$$\Delta x_\alpha \equiv x_\alpha - \bar{x}, \tag{3.9}$$

and

$$\bar{x} \equiv \frac{1}{M}\sum_{\alpha=1}^{M} x_\alpha. \tag{3.10}$$

Obviously, $\bar{x}$ is the centre of mass of the polymer ring. With the definition we have given of Δ we are giving account of the length of the polymer ring, therefore, a larger value of Δ implies a more spread out polymer.

In the limit of $M \to \infty$, and for the free particle case, equation (3.8) becomes (Gillan, 1990):

$$\Delta^2 = \frac{\beta\hbar^2}{12m} = \frac{\pi}{6}\Lambda^2, \tag{3.11}$$

with Λ the thermal wavelength. In the classical limit (high temperatures, large mass or $\hbar \to 0$) the value of Δ goes to zero, meaning that the necklace shrinks and all the beads of an atom occupy the same position. In the quantum regime, the necklace spreads out becoming infinity at $T = 0$, as we would expect.

In the presence of a potential the spreading is limited by the potential itself. At a given temperature and for a given mass, the limiting factor is the potential, this has a very useful practical consequence. A polymer ring within a real system simulation experiences two opposing forces. From one hand, the beads within an atom want to approach each other, since there are coils connecting them but, on the other hand, there is an inter–atomic potential attracting beads from different atoms and spreading the necklaces. The thermal equilibrium is reached when the mean value of these two forces balances. However, should the number of beads per atom were not sufficiently high, the polymer would not spread as much as required for the given conditions of mass, temperature and potential. The results obtained in that situation would be biased towards the classical limit. But, if we put a sufficient amount of beads—time slices—the polymer would not spread out more than necessary for the given conditions, since there is a potential that limits the length of the necklace. Increasing the number of beads once the sufficient amount has been found would not lead to better results.

In equation (3.7) we said that the exact partition function is reached using an infinite number of beads. For, our results will be exact, provided that a sufficient number of beads is taken, and we only have to worry about the statistical error.

From a physical point of view, a sufficient number of beads is roughly reached when the root mean squared length of the springs is smaller than the relevant length scale of the external potential (Gillan, 1990). This translates into the following relation:

$$M \gg \frac{\beta \hbar^2}{m\sigma^2}, \tag{3.12}$$

with M the number of beads, and σ the length scale of the potential. To a first order, we can say that the number of beads required is inversely proportional to the temperature.

3.2 The harmonic oscillator

In section § 2.2.4 we gave the analytic result for the energy of a quantum harmonic oscillator. In order to test our codes we shall first construct a simple algorithm for estimating the energy of the quantum harmonic oscillator and compare it with the analytic result. For, we need to find the action of a one dimensional harmonic oscillator. We shall first find the primitive approximation to the action, and later several analytic improvements to that result. We intend to compare the behaviour of these different actions in section § 7.1.

The Hamiltonian we shall use through out this section is that one given by equation (2.46).

3.2.1 High–order correction to the Trotter expansion

The partition function in the primitive approximation is

$$\begin{aligned}\rho(x_0, x_M, \beta) &= \int dx_1 \cdots dx_{M-1} (4\pi\lambda\tau)^{-3M/2} \\ &\times \exp\left(- \sum_{\alpha=1}^{M} \left[\frac{(x_{\alpha+1} - x_\alpha)^2}{4\lambda\tau} + \tau V(x_\alpha) \right] \right), \quad (3.13)\end{aligned}$$

with $\lambda = \hbar^2/2m$. Since we are simulating polymer rings, a cyclic condition is assumed, meaning that $x_1 \equiv x_{M+1}$. The mass of the particle is m. This approximation is accurate to order $(\beta/M)^2$ (Raedt and Raedt, 1983), (Fye, 1986). Nevertheless, should we take a sufficiently large number of slices, in theory infinite, the results obtained would be exact (Gillan, 1990). However we might wonder about the effects of analytically improving the action in order to place less reliance upon the sampling. In other words, we hope to reduce the number of beads required by including more terms of the expansion of the partition function.

In preparation for giving this high–order correction to the Trotter expansion we shall give a rearrangement of the Baker–Campbell–Hausdorff formula. The derivation presented here follows closely the one given by Li and Broughton (1987). There is, however, a similar derivation of a fourth–order propagator by Takahashi and Imada (1984), albeit they only produce results for the harmonic oscillator.

A given bead of the polymer ring feels the interaction of the external field (in our present case the harmonic oscillator potential) and also the harmonic potential produced by the springs connecting the bead to its neighbours. These two adding potentials can be included in a single effective potential whose representation is associated with all the interactions the bead experiences. Translating this into a more formal language:

$$V_{\mathrm{eff}}(x_1,\ldots,x_M) = \frac{mM}{2\hbar^2\beta^2}\sum_{\alpha=1}^{M}(x_{\alpha+1}-x_\alpha)^2 + \frac{1}{M}\sum_{\alpha=1}^{M}V(x_\alpha). \tag{3.14}$$

As a consequence of the path integral formulation, the partition function, $Z = \exp(-\beta H)$, for a one dimensional particle can be expressed, in terms of the effective potential as

$$Z = \int dx_1\cdots dx_M \left(\frac{mM}{2\pi\hbar^2\beta^2}\right)^{M/2} \exp[-\beta V_{\mathrm{eff}}(x_1,\ldots,x_M)]. \tag{3.15}$$

Alongside, given A and B any pair of operators, the Baker–Campbell–Hausdorff formula can be expressed as

$$\exp[-\tau(A+B)] = g_4(A,B;\tau) + \mathcal{O}(\tau^5), \tag{3.16}$$

with the usual definition $\tau \equiv \beta/M$, and where

$$g_4(A,B,\tau) = e^{-\tau A/2}e^{-\tau B/2}e^{-\tau^3 C/24}e^{-\tau B/2}e^{-\tau A/2}, \tag{3.17}$$

being C an operator defined as

$$C \equiv [[B,A],A+2B]. \tag{3.18}$$

A correct expansion to order $(\beta/M)^4$ obtained using the previous expressions is

$$Z = \mathrm{Tr}[g_4(A,B,\beta/M)]^M. \tag{3.19}$$

However, equation (3.19) is not very useful for programming purposes (Li and Broughton, 1987), (Raedt and Raedt, 1983).

Equation (3.17) can be expressed equivalently either as

$$g_4(A,B,\tau) = e^{-\tau^3 C/24}e^{-\tau A/2}e^{-\tau B}e^{-\tau A/2}e^{-\tau^4[A+B,C]/48}, \tag{3.20}$$

or as

$$g_4(A,B,\tau) = e^{-\tau A/2} e^{-\tau B} e^{-\tau A/2} e^{-\tau^3 C/24} e^{-\tau^4 [A+B,C]/48}. \qquad (3.21)$$

Substitution of equations (3.20) and (3.21) into (3.19) yield two equivalent expressions for the partition function

$$g_4^{(1)} = e^{-\tau B/2} e^{-\tau A/2} e^{-\tau^3 C/24} e^{-\tau A/2} e^{-\tau B/2} e^{\tau^4 [A+B,C]/48}, \qquad (3.22)$$

and

$$g_4^{(2)} = e^{-\tau B/2} e^{-\tau A/2} e^{-\tau^3 C/24} e^{-\tau A/2} e^{-\tau B/2} e^{-\tau^4 [A+B,C]/48}. \qquad (3.23)$$

Since (3.22) and (3.23) are partition functions, an average of them is also a partition function, therefore a new approximation valid to fourth order is

$$g_4^{(3)} = e^{-\tau B/2} e^{-\tau A/2} e^{-\tau^3 C/24} e^{-\tau A/2} e^{-\tau B/2}. \qquad (3.24)$$

Comparing (3.24) to (3.17) we notice that they are the same equation with the operators A and B exchanged. Now we let $A = T$ and $B = V$ in (3.17), and $A = V$ and $B = T$ in equation (3.24). Making a new linear combination of two partition functions we obtain

$$\begin{aligned} g_4^{(4)} &= \frac{2}{3} g_4(T,V;\tau) + \frac{1}{3} g_4^{(3)}(V,T;\tau) \qquad (3.25) \\ &= e^{-\tau T/2} e^{-\tau V/2} e^{-\tau^3 C'/24} e^{-\tau V/2} e^{-\tau T/2}, \end{aligned}$$

where

$$C' = [[V,T],V] = \frac{\hbar^2}{m} (\nabla V)^2. \qquad (3.26)$$

Substitution of (3.26) into (3.17) gives a useful expression for the new effective potential in terms of the potential defined in (3.14)

$$V_{\mathrm{LB}}(x_1,\ldots,x_M) = V_{\mathrm{eff}}(x_1,\ldots,x_M) + \frac{\beta^2 \hbar^2}{24 M^3 m} \sum_{\alpha=1}^{M} [\partial_\alpha V(x_\alpha)]^2, \quad (3.27)$$

which is the Li–Broughton effective one–dimensional potential for a single–particle, and

$$\frac{\beta^2 \hbar^2}{24 M^3 m} \sum_{\alpha=1}^{M} [\partial_\alpha V(x_\alpha)]^2, \qquad (3.28)$$

is the so–called Li–Broughton correction to the primitive approximation to the single particle one dimensional potential. It is straight forward to convert (3.28) into computer language. However, the partial derivative in (3.28) becomes a gradient for the three dimensional. Also and additional index, labelling atoms, appears when dealing with real systems, in which more than one particle is involved. Therefore, some algebra has to be performed in order to get a codable expression. This job is carried out in detail in the Appendix A.

3.2.2 Li–Broughton correction applied to the harmonic oscillator

For the sake of completeness we shall give the derivation of Li–Broughton's effective potential for the harmonic oscillator. We shall be using the harmonic potential

$$V = \frac{1}{2} m\omega^2 x^2. \tag{3.29}$$

Substitution of (3.29) into (3.27) yields

$$V_{\text{LBH}} = \frac{mM}{2\hbar^2\beta^2} \sum_{\alpha=1}^{M} (x_{\alpha+1} - x_\alpha)^2 + \frac{m\omega^2}{2M} \sum_{\alpha=1}^{M} x_\alpha^2 + \frac{\beta^2\hbar^2}{24M^3} m\omega^4 \sum_{\alpha=1}^{M} x_\alpha^2. \tag{3.30}$$

Using the appropriate units in order to have $\hbar = 1$, which is the value we shall use in our simulation, and substituting for the definition of τ we get

$$V_{\text{LBH}} = \frac{1}{2} \frac{m}{\tau^2} \frac{1}{M} \sum_{\alpha=1}^{M} (x_{\alpha+1} - x_\alpha)^2 + \frac{1}{2} m\omega^2 \left(1 + \frac{\tau^2\omega^2}{12}\right) \frac{1}{M} \sum_{\alpha=1}^{M} x_\alpha^2 \tag{3.31}$$

which is the Li–Broughton potential for the harmonic oscillator. Notice that this is exactly the same potential that would be obtained using the primitive approximation (3.14), but for a corrective term, which is called the Li–Broughton correction.

A similar derivation to ours, with comparisons between the harmonic and anharmonic oscillators can be found in a paper from Burghardt et al. (1998). Even though they use the high order corrections provided by Takahashi and Imada (1984) and Li and Broughton (1987), they do not claim they are fourth order. Cao and Berne (1993) have also performed simulations involving the Li–Broughton correction applied to the harmonic oscillator when studying electron solvation in polarizable fluids.

3.2.3 A slight improvement of the potential

It was suggested by Feynman (Feynman and Hibbs, 1965b) (Feynman, 1998) that integrating the potential over a straight line would result in better approximations to the action. Replacing the potential $V(x_\alpha)$ in (3.14) or (3.27) for

$$\frac{V(x_\alpha)+V(x_{\alpha-1})}{2} \tag{3.32}$$

gives in fact a second–order in the primitive approximation to the action, which otherwise cannot be achieved. It could also provide a betterment to use instead

$$V\left(\frac{x_\alpha+x_{\alpha-1}}{2}\right). \tag{3.33}$$

3.2.4 Variational improvement over the classical action

In §3.2.3 we integrated along a straight line. An improvement over that first approach is to deviate from the classical path. This idea combined with the variation theorem allowed Feynman to give an improved formula for the potential (Feynman and Hibbs, 1965b).

From the Gibbs–Delbrück–Moliere minimum principle it can be derived the following relationship

$$E \leq E_0 + \frac{1}{\beta_b}\langle S - S_0\rangle, \tag{3.34}$$

where β_b is the inverse of the temperature per bead. Making use of variational calculus leads to a expression for the potential, which is

$$V_{\mathrm{F}}(y) = \sqrt{\frac{6mkT_b}{\pi\hbar^2}} \int dz\, V(z)\, \exp\left[-\frac{6mkT_b(y-z)^2}{\hbar^2}\right]. \tag{3.35}$$

Equation (3.35) is indeed the given potential smeared out by a Gaussian. Since this potential have been found using (3.34) we got an upper bound for the potential energy. Furthermore, the variational principle gives us the better choice we could find for the single bead energy. That means we are performing the path integral just for one bead i.e., not using the Trotter formula. Thus, temperatures appearing in (3.35) and (3.34) are indeed $\beta_b = M\beta$ and $T_b = MT$.

3.2.5 Variational theorem applied to the harmonic oscillator

Application of (3.35) to $V(z) = m\omega^2 z^2/2$ leads us to an integral of the kind

$$\int_{-\infty}^{\infty} x^n e^{-px^2+2qx} dx = \frac{1}{2^{n-1}p}\sqrt{\frac{\pi}{p}}\frac{d^{n-1}}{dq^{n-1}}\left(qe^{q^2/p}\right). \tag{3.36}$$

Applying (3.36) to the harmonic potential yields

$$V_{\mathrm{FH}}(x) = \frac{m\omega^2}{2}\left(x^2 + \frac{\hbar^2\beta}{12Mm}\right), \tag{3.37}$$

where V_{FH} stands for Feynman harmonic potential.

3.2.6 Li–Broughton correction to Feynman's variational principle applied to the harmonic oscillator

Up until now we have derived the primitive approximation to the action, the Li–Broughton correction, and Feynman's variational improvement. Li–Broughton's correction and Feynman's variational approach are independent improvements to the action, therefore they can be applied successively. That is exactly what we are going to do in this and the next section. In this section (§3.2.6) we are going to apply the Li–Broughton correction to the action obtained using Feynman's variational principle in the harmonic oscillator, that is, to equation (3.37). In the next section (§3.2.8) we shall use Feynman's principle upon a Li–Broughton action for the harmonic oscillator, that is, upon equation (3.30).

We have seen so far that Feynman's variational principle when applied to the harmonic potential gives a new constant term $\omega^2\hbar^2\beta/24M$. That term is added to the classical harmonic potential i.e., is just a correction in temperature. On the other hand, Li–Broughton's work proposes a correction in x^2, that is, to deal with a different parabola—not just a shift in the y axis—which gives better accountant of the quantum nature of the problem. When a classical particle reaches a potential barrier only 'feels' the potential once it has hit it. Otherwise, quantum particles 'feel' the effects of potential barriers before arriving to them.

Now we are going to see what happens if a Li–Broughton correction scheme is applied to a Feynman variational theorem in the case of the harmonic oscillator. Our starting point is (3.37). We substitute it in (3.27) in order to get

$$V_{\mathrm{LBFH}} = \frac{m\omega^2}{2}\left(1+\frac{\beta^2\hbar^2\omega^2}{12M}\right)x^2 + \frac{\omega^2\hbar^2\beta}{24M}. \tag{3.38}$$

3.2.7 Feynman's variational principle applied to the Li–Broughton correction of the harmonic oscillator

We are obviously trying to explore all the possible combinations amongst the primitive action, Li–Broughton's correction and Feynman's variational principle. The last possible choice is to apply Feynman's variational principle to an already Li–Broughton corrected potential. The result obtained is:

$$V_{\mathrm{FLBH}} = \left(\frac{m\omega^2}{2}+\frac{m\omega^4\hbar^2\beta^2}{24M^2}\right)x^2 + \frac{\omega^2\hbar^2\beta}{24}\left(1+\frac{\omega^2\hbar^2\beta^2}{12M}\right), \tag{3.39}$$

Notice that in the high temperature limit i.e., $\beta \to 0$ the harmonic classical potential V is retrieved, as expected. We see how Feynman variational theorem always improves the accuracy with a higher order in temperature. Since V_{LBH} has a β^2 dependence, the application of the variational theorem yields a β^3 order.

3.2.8 Feynman and Li–Broughton corrections applied to the anharmonic oscillator

Now we are going to play the same game we did before, but applied to the anharmonic oscillator. Since the procedure for finding each potential and potential energy estimator is the same we shall only give the formulæ. The behaviour of the results we obtained using the anharmonic oscillator in our simulations was the same we observed for the harmonic case.

$$V_{\mathrm{A}} = \frac{m\omega^2}{2}x^2 + \frac{\lambda m\omega^4}{4}x^4 \tag{3.40}$$

$$\begin{aligned} V_{\mathrm{FA}} &= \frac{m\omega^2}{2}\left(x^2 + \frac{\hbar^2\beta}{12Mm}\right) \\ &+ \frac{\lambda m\omega^4}{16}\left(\frac{\hbar^2\beta}{6Mm}\right)^2\left[\left(3 + \frac{12mM}{\hbar^2\beta}x^2\right)^2 - 6\right] \end{aligned} \tag{3.41}$$

$$\begin{aligned} V_{\mathrm{LBA}} &= \frac{m\omega^2}{2}\left(1 + \frac{\omega^2\beta^2\hbar^2}{12M^2}\right)x^2 \\ &+ \frac{\lambda m\omega^4}{4}\left(1 + \frac{\omega^2\beta^2\hbar^2}{3M^2}\right)x^4 + \frac{\lambda^2 m\omega^8\beta^2\hbar^2}{24M^2}x^6 \end{aligned} \tag{3.42}$$

3.3 *Going towards real systems*

So far we have derived several actions for one dimensional systems, in particular, the free particle, the harmonic oscillator and the anharmonic oscillator. In forthcoming chapters the corresponding energy estimators for these actions will be derived, also tests will be carried out. However, we might say in advance that only the primitive and the Li–Broughton action will be worth to use in real systems simulations. In the light of this conclusion we shall derive these two both actions including a three dimensional space and several atoms within the simulation box. Remember that up until now we have only dealt with a single particle—constituted by many beads—within a one dimensional potential. From now on we shall be using Greek indexes for labelling beads and Latin indexes for labelling atoms, e.g. $\mathbf{r}_{i\alpha}$ would be the three dimensional position vector representing the α–th bead of the i–th atom.

3.3.1 Primitive approximation to the action in the many particle problem

Equation (3.14) is the primitive approximation to the action for a single one–dimensional particle. That equation becomes

$$V_{\text{eff}} = \frac{1}{4\lambda\tau^2}\frac{1}{M}\sum_{\alpha=1}^{M}(\mathbf{r}_{\alpha+1} - \mathbf{r}_{\alpha})^2 + \frac{1}{M}\sum_{\alpha=1}^{M} V(\mathbf{r}_{\alpha}), \qquad (3.43)$$

for the case of a single three–dimensional particle. If we are dealing with more than one atom, then (3.43) has to be averaged out with the number of atoms, yielding

$$V_{\text{eff}} = \frac{1}{4\lambda\tau^2}\frac{1}{NM}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}(\mathbf{r}_{i\alpha+1} - \mathbf{r}_{i\alpha})^2 + \frac{1}{NM}\sum_{i=1}^{N}\sum_{\alpha=1}^{M} V(\mathbf{r}_{i\alpha}), \qquad (3.44)$$

where N is the number of atoms.

3.3.2 Li–Broughton's correction applied to the many particle problem

We have already found the correction for the one dimensional single particle case, which is given by (3.27). That expression can be translated into a three dimensional formula as

$$V_{\text{LB}} = \frac{1}{4\lambda\tau^2}\frac{1}{M}\sum_{\alpha=1}^{M}(\mathbf{r}_{\alpha+1}-\mathbf{r}_{\alpha})^2+\frac{1}{M}\sum_{\alpha=1}^{M} V(\mathbf{r}_{\alpha})+\frac{\lambda\tau^2}{12}\frac{1}{M}\sum_{\alpha=1}^{M}\left[\nabla_{\alpha}V(\mathbf{r}_{\alpha})\right]^2 . \qquad (3.45)$$

Equation (3.45) is a three dimensional formula and yet for a single particle.

When dealing with a many particle problem the squared gradient of the potential that appears in the Li–Broughton correction to the kinetic part of the energy must be treated with care due to the chain rule. Here, the potential is not an external potential *ad hoc* as in the case of the harmonic oscillator. The potential experimented by a given bead is produced by the force (the gradient of the potential) exerted over that bead by all the other beads that coexist in the same given imaginary time step. In this section we will give Li–Broughton's expression for the action Then we shall give an explicit expression for the squared gradient of the potential that is suitable for programming. The derivation, for a given potential, is not complicated but rather lengthy, thus in this section we shall only give the results that are going to be used when programming and we address the reader to the Appendix A for the full derivation.

Li and Broughton showed that the commutator

$$\begin{aligned} C' = [[V,T],V] &= (VT - TV)V - V(VT - TV) \\ &= -(TV)V - V(VT) + 2VTV, \end{aligned} \tag{3.46}$$

gives the right correction to fourth order of the action. Making use of (3.46) and knowing that the quantum kinetic and potential operators are

$$T = -\frac{\hbar^2}{2m}\frac{1}{M}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}\nabla_{i\alpha}^2, \tag{3.47}$$

and

$$V = \frac{1}{M}\sum_{\alpha=1}^{M}\sum_{i<j}V\left(|r_{i\alpha} - r_{j\alpha}|\right) \tag{3.48}$$

respectively. The number of atoms is given by N. Therefore, the expression for the Li–Broughton correction to the effective potential is

$$V_{LB} = \frac{\hbar^2}{24m}\frac{\beta^2}{M^3}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}\mathbf{F}_{i\alpha}\cdot\mathbf{F}_{i\alpha}. \tag{3.49}$$

The 'forces' $\mathbf{F}_{i\alpha}$ are defined as

$$\begin{aligned} F_{i\alpha}^a &= \sum_{k\neq i}\nabla_{i\alpha}^a V\left(|r_{i\alpha} - r_{k\alpha}|\right) \\ &= \sum_{k\neq i}\frac{dV\left(|r_{i\alpha} - r_{k\alpha}|\right)}{d\left(|r_{i\alpha} - r_{k\alpha}|\right)}\frac{\left(r_{i\alpha} - r_{k\alpha}\right)^a}{|r_{i\alpha} - r_{k\alpha}|}, \end{aligned} \tag{3.50}$$

where a refers to the coordinate index i.e., x, y, z.

Li and Broughton (1987) claim that their propagator is accurate to fourth order. The results of our numerical experiments with the harmonic oscillator agree with that, however, the numerical results we obtained with real systems show only a fourth–order behaviour in some restricted regime. We will discuss about this in more detail in the chapter devoted to testing.

3.3.3 Addendum

Shortly before submitting this thesis we realised, thanks to a private communication with Siu A. Chin, about the existence of a recently published paper (Jang et al., 2001) that shows a different expression for a fourth–order propagator. The results showed therein are based on a publication did some years before by Chin (1997). In the light of these references it seems that the propagator proposed by Li and Broughton (1987) is not fully fourth–order, which would be in agreement with some of our numerical results for real systems.

4. SAMPLING

The concept of sampling was introduced at the very beginning of the thesis saying that it was an essential ingredient of any path integral code. In addition, we pointed out that sampling was the way beads were moved. Now, we are in the position to discuss sampling in a more formal way and to explain the problems arisen and the solutions given.

In classical liquid simulations atoms are depicted by point–like particles within a box. The time evolution of these particles is obtained by iteratively applying Newton's laws and solving all the interactions. When this process is repeated many times the system reaches thermal equilibrium. In a quantum liquid simulation the picture is similar, being an important difference the way atoms are described. Due to the polymer isomorphism atoms are represented by necklaces made up of beads, and each bead is connected to its neighbours by a couple of kinetic springs. Thus, instead of representing an atom with a single point–like particle, it is represented by a collection of interacting point–like particles. It is intuitive to think that this gathering of necklaces should also be moved in order to reach thermal equilibrium. The sampling method is the way the beads within the necklaces are displaced.

A more formal perspective is to think of the density matrix as a probability distribution that should be sampled following the distribution given by the action. Therefore, we must perform a large number of multidimensional integrations and stochastic methods are the only useful choice for this task. The distribution is different from that of a classical liquid, since the points on the path are connected with kinetic springs. This causes a very slow convergence due to the long correlation times and, therefore, the traditional methods used with classical liquids become useless. However, the classical polymer isomorphism can be exploited and some simulation techniques used in that field can be employed.

In this chapter we shall study three main sampling methods. The first one is a simple approach to sampling, the bead per bead method. It is very easy to code and it will provide us with useful concepts for developing better sampling methods. We will observe, however, that bead per bead sampling is doomed to fail. If the atoms are made up of more than sixteen beads it may become very slow and convergence can hardly be reached. For going lower in temperature we need to provide a better sampling method, one that is able to sample longer chains. A way to overcome the strong correlation amongst beads is to propose collective movements. There are several sampling schemes that make thorough use of that idea: staging, bisection and multigrid. We are going to study the first two of them.

Staging (Sprik et al., 1985) will be the second sampling method we shall study. It is based on smart collective movements. We performed few tests on it, being the benchmarks obtained similar to those of bisection. Since both methods are very similar, this is what we would expect

The last method—bisection (Ceperley and Pollock, 1992)—is similar to staging in the sense of being a smart collective displacement scheme, that means that several beads are independently displaced at a time. It also shares with staging the capability of exactly sampling the free particle density matrix, and using the Metropolis algorithm just for sampling the potential interaction. This is going to be our method of choice. There is a negligible bias in the bisection method, that in all cases falls within the statistical error. Notwithstanding, we shall present a small variant to the standard bisection method that we have developed.

4.1 Bead per bead sampling

Bead per bead sampling is the easiest way to sample a probability distribution. The algorithm is very simple: first the action of the whole system is calculated. Then a random displacement around the actual position of a bead is performed. After the random displacement the action of the new configuration space is calculated and compared via the Metropolis algorithm (Metropolis et al., 1953) to the old action. If the displacement is accepted the positions in configuration space are updated; if not the positions remain equal to the beginning of the algorithm. Accepted and rejected moves contribute to thermal averages in the same way. Once in a while all the centre of masses of the polymer rings are displaced with and acceptance also guided by the Metropolis algorithm. That is, movements of chains as a whole—without changing their inner structure—are proposed and accepted or rejected using Metropolis. Repetition of this process will eventually drive the system to equilibrium. The sampling equilibrium reached can be understood in the sense of a microscopic equilibrium of probability, known as the detailed balance condition.

In every sampling process there are several concepts we should bear in mind. There are many references that provide a formal presentation of the subject (Kalos M. H. and Whitlock P. A., 1986), (Binder, 1979), and there are also ones that give easy approaches to the subject along with codes and algorithms (Frenkel and Smit, 1996), (Gould and Tobochnik, 1996). Thus, we shall only sketch those ideas of stochastic processes directly related to PIMC within the frame of bead per bead sampling.

The Metropolis algorithm is a method for solving the inverse problem in a Markov chain. Knowing the probabilities of each state, the Metropolis algorithm, asymptotically builds up the transition probabilities between different states. To construct a Markov chain the configuration space of a system has to be changed according to a transition rule, which is indeed a probability distribution. Let us define $P(s \to s')$ as a probability distribution to go from state s to state s'. Repeated application of this transition rule will generate a random walk. A transition is ergodic if it is possible to access any given state in a finite number of states. For a full discussion on this subject see Krauth (1998)

Let us define the probability distribution

$$\pi(s) = \frac{\exp[-\sum_{\alpha=1}^{M} S_\alpha]}{Z}, \tag{4.1}$$

with Z the partition function that normalises $\pi(s)$ and S_α the action of the α–th link. Detailed balance can be put in a formal way as

$$\pi(s)P(s \to s') = \pi(s')P(s' \to s). \tag{4.2}$$

To satisfy detailed balance is a sufficient condition for an ergodic system in order to guarantee that the probability distribution $\pi(s)$ will be sampled in a finite number of steps.

According to the generalised Metropolis criterion the transition probability can be split as

$$P(s \to s') = T(s \to s')A(s \to s') \tag{4.3}$$

into a *a priori* sampling distribution $T(s \to s')$ and an acceptance probability $A(s \to s')$. In the original Metropolis algorithm, $T(s \to s') = T(s' \to s)$, however $T(s \to s')$ might have other definitions. The trial moves are accepted or rejected according to

$$A(s \to s') = \min \left[1, \frac{T(s' \to s)\pi(s')}{T(s \to s')\pi(s)} \right], \tag{4.4}$$

which is the Metropolis method. Equation (4.4) verifies detailed balance, therefore, convergence is guaranteed.

The bead per bead sampling method satisfies all these conditions, and hence, is essentially correct. However, we have to remember that in order for the primitive approximation to be valid, one has to start with a classical density matrix, that is, a high temperature density matrix. If we want to decrease in temperature, down to the regime where quantum effects arise, a large number of density matrices has to be convoluted. This means more than sixteen beads in our polymer rings. With these conditions this kind of sampling becomes slow and convergence requires large CPU times. In fact, it has been proved that autocorrelation times diverge quadratically with the number of beads (Janke and Sauer, 1997), (Nightingale, 1999). That is the reason why other sampling schemes have been proposed.

4.2 Staging

Due to the severe slowing down of a bead per bead sampling a better sampling method is required. There are several methods that surmount critical slowing down. Some of them come from numerical differential equation theories like the W–cycle in multigrid techniques (Goodman and Sokal, 1986). Janke and Sauer (1993) have tested it in several academic models Janke and Sauer (1996), and they claim to reduce critical slowing down to a constant. However, our tests yielded a linear dependence with the number of beads.

Another sampling method uses terms of a Fourier expansion instead of beads. In this way paths become smoother, and also critical slowing down is reduced. A comparison between discrete space PIMC and Fourier space PIMC can be found in (Chakravarty et al., 1998).

Other methods propose smart collective movements of beads, such as staging or bisection. In this section we shall focus on staging. It will serve us for introducing some ideas common to bisection which will become our method of choice. We have tried V–cycle and the W–cycle, as well as Fourier and staging in our preliminary tests with the harmonic oscillator, and overall bisection turn out to be a better choice. Nevertheless the equations and algorithms presented here will be useful for anyone interested in coding a staging sampling since we shall give formulæ that are straight forward to code.

For a large number of beads the kinetic term in the action dominates over the potential term, thus, the main cause of slowing down comes out from the inter–bead interaction. If we could remove the springs coupling the beads, slowing down would not be a problem. The main issue of staging, as well as bisection, is to separate the kinetic and potential parts of the action and to include the kinetic part within the sampling. Should not exist a potential interaction, that is, were we sampling the free particle density matrix, the acceptance would be one hundred per cent and there were no need of a Metropolis algorithm. Furthermore, the staging method proposes a collective smart displacement of the beads. It takes a segment of the polymer ring and moves it in a way that beads within the segment can be considered independent and not coupled. This is achieved redefining the position coordinate and the mass of the beads in terms of new positions and masses denoted by starred variables. For the sake of clarity we shall develop the theory only for a single atom, thus all the indexes appearing are bead labels. In this section we shall use Latin indexes for labelling beads.

The staging method is a sampling strategy that takes advantage from the fact that the kinetic part of the action may be exactly sampled. The same strategy used for the primitive approximation is employed, and the kinetic part of the action is separated from the potential one. Random numbers are generated with the kinetic probability distribution function and then the Metropolis criteria is only applied to the potential part, since the kinetic part has been sampled exactly.

4.2.1 The staging variables

Let us consider a segment, of length j, of the polymer chain. That is, the number of beads within the segment is going to be j, which is the only free adjustable parameter regarding to sampling

$$\cdots \rho_0(x_{i-1}, x_i; \tau) \underbrace{\rho_0(x_i, x_{i+1}; \tau) \cdots \rho_0(x_{i+(j-1)}, x_{i+j}; \tau)}_{j-1} \rho_0(x_{i+j}, x_{i+j+1}; \tau) \cdots \tag{4.5}$$

We are going to work out the terms inside the braces. The following identity can be stated

$$\begin{aligned} &\rho_0(x_i, x_{i+1}; \tau) \cdots \rho_0(x_{i+(j-1)}, x_{i+j}; \tau) = \rho_0(x_i, x_{i+j}; j\tau) \times \\ &\left[\frac{\rho_0(x_i, x_{i+1}; \tau)\rho_0(x_{i+1}, x_{i+j}; (j-1)\tau)}{\rho_0(x_i, x_{i+j}; j\tau)}\right] \times \\ &\left[\frac{\rho_0(x_{i+1}, x_{i+2}; \tau)\rho_0(x_{i+2}, x_{i+j}; (j-2)\tau)}{\rho_0(x_{i+1}, x_{i+j}; (j-1)\tau)}\right] \times \cdots \\ &\left[\frac{\rho_0(x_{i+(j-2)}, x_{i+(j-1)}; \tau)\rho_0(x_{i+(j-1)}, x_{i+j}; \tau)}{\rho_0(x_{i+(j-2)}, x_{i+j}; 2\tau)}\right]. \end{aligned} \tag{4.6}$$

Now we can split each one of the terms in square brackets in an easy form. Let us take a generic one

$$\begin{aligned} &\frac{\rho_0(x_{i+k}, x_{i+k+1}; \tau)\rho_0(x_{i+k+1}, x_{i+j}; (j-(k+1)\tau)}{\rho_0(x_{i+k}, x_{i+j}; (j-k)\tau)} = \\ &\frac{\exp\left[-\frac{m}{2\hbar^2(j-(k+1))\tau}(x_{i+k+1}-x_{i+j})^2\right]\exp\left[-\frac{m}{2\hbar^2\tau}(x_{i+k}-x_{i+k+1})^2\right]}{\exp\left[-\frac{m}{2\hbar^2(j-k)\tau}(x_{i+k}-x_{i+j})^2\right]}, \end{aligned} \tag{4.7}$$

where i, j are fixed indexes and $k = 0, \ldots, (j-2)$. The arguments of the exponentials in (4.7) can be expanded and sorted in the following way

$$
\begin{aligned}
& - && \frac{1}{j-(k+1)}(x_{i+k+1}-x_{i+j})^2 - (x_{i+k}-x_{i+k+1})^2 \\
& + && \frac{1}{j-k}(x_{i+k}-x_{i+j})^2 = \frac{j-k}{j-(k+1)} \times \\
\left[\vphantom{\frac{1}{j}}\right. & - && \frac{1}{j-k}(x^2_{i+k+1}+x^2_{i+j}-2x_{i+k+1}x_{i+j}) \\
& - && \frac{j-(k+1)}{j-k}(x^2_{i+k}+x^2_{i+k+1}-2x_{i+k}x_{i+k+1}) \\
& - && \frac{j-(k+1)}{(j-k)^2}(x^2_{i+k}+x^2_{i+j}-2x_{i+k}x_{i+j}) \left.\vphantom{\frac{1}{j}}\right] = \\
& && \frac{j-k}{j-(k+1)}\left[x^2_{i+k+1} \underbrace{\left(\frac{-1}{j-k} - \frac{j-(k+1)}{j-k}\right)}_{\text{A}} \right. \\
& + && 2x_{i+k+1} \underbrace{\left(\frac{x_{i+j}}{j-k} + \frac{x_{i+k}(j-(k+1))}{j-k}\right)}_{\text{B}} \\
& + && \left. \underbrace{\left(-\frac{x^2_{i+j}}{j-k} - \frac{j-(k+1)}{j-k}x^2_{i+k} + \frac{j-(k+1)}{(j-k)^2}\left(x^2_{i+k}+x^2_{i+j}-2x_{i+k}x_{i+j}\right)\right)}_{\text{C}} \right].
\end{aligned}
$$

The three parenthesis (A, B, and C) can be rearranged and yield the definition of the new position variables

$$
\text{A} = -1, \tag{4.8}
$$

$$
\text{B} = \frac{x_{i+j} + x_{i+k}(j-(k+1))}{j-k} \equiv x^*_{i+k+1}, \tag{4.9}
$$

$$
\begin{aligned}
\text{C} \quad = \quad & x^2_{i+j}\frac{1}{(j-k)^2} - x^2_{i+k}\left(\frac{j-(k+1)}{j-k}\right)^2 \\
- \quad & 2x_{i+k}x_{i+j}\frac{j-(k+1)}{(j-k)^2} \equiv -x^{*^2}_{i+k+1}.
\end{aligned} \tag{4.10}
$$

Using the definitions (4.9) and (4.10) it is possible to rewrite the arguments of the exponentials in (4.7) as

$$\begin{aligned} & \frac{j-k}{j-(k+1)}\left[-x_{i+k+1}^2+2x_{i+k+1}x_{i+k+1}^*-x_{i+k+1}^{*2}\right] \\ = \quad & -\frac{j-k}{j-(k+1)}(x_{i+k+1}-x_{i+k+1})^2. \end{aligned} \tag{4.11}$$

Once the new staging positions are defined we define a k mass by

$$m_k \equiv m\left(\frac{j-k}{j-(k+1)}\right). \tag{4.12}$$

Each one of the square brackets in the action is written as

$$\exp\left[-\frac{m_k}{2\hbar^2\tau}\left(x_{i+k+1}-x_{i+k+1}^*\right)^2\right], \tag{4.13}$$

where the staging coordinates are given by

$$x_{i+k+1}^* = \frac{x_{i+j}+x_{i+k}(j-(k+1))}{j-k}. \tag{4.14}$$

In a similar fashion the prefactor of each term yields

$$\frac{\left(\frac{m}{2\pi\hbar^2\tau}\right)^{1/2}\left(\frac{m}{2\pi\hbar^2(j-(k+1))\tau}\right)^{1/2}}{\left(\frac{m}{2\pi\hbar^2(j-k)\tau}\right)^{1/2}} = \left(\frac{m_k}{2\pi\hbar^2\tau}\right)^{1/2}. \tag{4.15}$$

Putting all together one arrives to the following identity

$$\begin{aligned} & \rho_0(x_i,x_{i+1};\tau)\cdots\rho_0(x_{i+j-1},x_{i+j};\tau) = \\ & \left(\frac{m}{2\pi\hbar^2 j\tau}\right)^{1/2}\exp\left[-\frac{m}{2\hbar^2 j\tau}(x_i-x_{i+j})^2\right]\times \\ & \prod_{k=0}^{j-2}\left(\frac{m_k}{2\pi\hbar^2\tau}\right)^{1/2}\exp\left[-\frac{m_k}{2\hbar^2\tau}(x_{i+k+1}-x_{i+k+1}^*)^2\right]. \end{aligned} \tag{4.16}$$

From the term inside the product of (4.16) we see the usefulness of the staging method. With the new staging variables we have obtained a system of uncoupled springs, and therefore, they are independent. This allows us to perform collective uncorrelated movements of beads.

4.2.2 The algorithm

Let us think now the way in which an efficient sampling of this "free" density matrix can be carried out. First, note that we can fix the first and last point of the chain i.e., the coordinates x_i and x_{i+j} do not change. The movements correspond to the $j-1$ beads in between. The sampling has to be performed in an iterative scheme since, as the index k increases, all the x^* depend on the precedent x.

1) $k = 0$.

$$m_0 = m\frac{j}{j-1} \tag{4.17}$$

$$x^*_{i+1} = \frac{x_{i+j} + x_i(j-1)}{j} \tag{4.18}$$

$$x'_{i+1} = x^*_{i+1} + \eta\sqrt{\frac{\hbar^2\tau}{m_0}} \tag{4.19}$$

2) $k = 1$.

$$m_1 = m\frac{j-1}{j-2} \tag{4.20}$$

$$x^*_{i+2} = \frac{x_{i+j} + x'_{i+1}(j-2)}{j-1} \tag{4.21}$$

$$x'_{i+2} = x^*_{i+2} + \eta\sqrt{\frac{\hbar^2\tau}{m_1}} \tag{4.22}$$

$\vdots$

$j-1$) $k = j-2$

$$m_{j-2} = m\frac{1}{2} \tag{4.23}$$

$$x^*_{i+j-1} = \frac{x_{i+j} + x'_{i+j-2}}{2} \tag{4.24}$$

$$x'_{i+j-1} = x^*_{i+j-1} + \eta\sqrt{\frac{\hbar^2\tau}{m_{j-2}}}, \tag{4.25}$$

where η is a normally distributed random number.

As it corresponds to an exact sampling of the kinetic action, the Metropolis criteria has to be applied only on the interaction part.

4.3 Bisection

In this section we shall explain the bisection algorithm. We are going to give a general explanation of the method. For the sake of clarity we shall give the formulas for a particular case of bisection. A code written in FORTRAN 77 that performs a general bisection is included in Appendix B. The purpose of the code is both, to illustrate the ideas exposed in the general algorithm explanation given below. There are also parts of the following algorithms that make use of FORTRAN instructions. It can be helpful for understanding the structure of the bisection code, a basic PIMC algorithm we have included in Appendix E. That appendix also contains some tricks that might be useful for coding and running a PIMC simulation.

The bisection algorithm proposes a faster approach to the sampling problem (Ceperley, 1995b), (Chakravarty et al., 1998). It only calculates the full action for movements that are very likely to be accepted. Movements that will not lead to an update of positions are rejected in early steps of the algorithm with an approximate action, much simpler to compute than the real one. Moreover, only the potential part of the action enters in the Metropolis acceptance–rejection scheme, since the kinetic part has an analytic solution which is the solution to the free particle density matrix given in equation (3.5). It is based on the Lévy construction (Lévy, 1939), an algorithm for building a Brownian bridge.

4.3.1 Lévy construction

It is the algorithm on which is based the bisection method. Its scope is to sample a free particle path. In the Lévy construction of a Brownian bridge one starts with to fixed end points, R_0 and R_β. The middle point between $\mathbf{R}_0$ and $\mathbf{R}_\beta$, namely $\mathbf{R}_{\beta/2}$, is sampled exactly via

$$\mathbf{R}'_{\beta/2} = \frac{\mathbf{R}_0 + \mathbf{R}_\beta}{2} + \boldsymbol{\eta}\sqrt{\frac{\lambda\beta}{2}}, \tag{4.26}$$

being $\boldsymbol{\eta}$ a normally distributed random vector, with mean zero and unit variance, and $\lambda = \hbar^2/2m$.

This choice exactly samples the free particle density matrix, so there are no rejections. Applying this algorithm recursively, the next two intervals to bisect are $(0, \beta/2)$ and $(\beta/2, \beta)$, yielding the points $\mathbf{R}_{\beta/4}$ and $\mathbf{R}_{3\beta/4}$. The algorithm continues until the difference between the intervals is τ.

4.3.2 Multilevel Metropolis

The multilevel Metropolis method is a general algorithm that can be used for sampling any distribution function having the convolution properties of the exponential operators. Since the bisection algorithm is a kind of multilevel Metropolis, we shall explain the latter and demonstrate that the algorithm asymptotically converges to the distribution function sampled.

Let us recall the definition of the probability distribution given in (4.1). We shall partition the configuration s in l levels as $s = (s_0, s_1, \ldots, s_l)$. The coordinates belonging to s_0 are to remain fixed, while those of s_1 will be sampled in the first level, and so on, until sampling the s_l coordinates in the l–th level. Thus the action can be partitioned as

$$\pi(s) = \prod_{k=0}^{l} \pi_k(s_0, s_1, \ldots, s_k). \tag{4.27}$$

Therefore, at any given level k the sampling probability, $T_k(s'_k)$ can only depend on $s_0, s_1, \ldots, s_{k-1}, s'_1, \ldots, s'_{k-1}$, but neither on $s_k, \ldots, s_l$, nor on $s'_{k+1}, \ldots, s'_l$. The primed coordinates indicate new trial positions.

Provided we have reached level k, the probability to go forward is given by

$$A_k(s') = \min\left\{1, \frac{T_k(s_k)\pi_k(s')}{T_k(s'_k)\pi_k(s)}\right\}. \tag{4.28}$$

This acceptance probability satisfies detailed balance in the following way

$$\pi_k(s)T_k(s'_k)A_k(s') = \pi_k(s')T_k(s_k)A_k(s). \tag{4.29}$$

A trial displacement is accepted only if it is accepted throughout all the levels, therefore the total probability of accepting a given trial displacement is given by the product of the acceptance probabilities at each level

$$P(s \to s') = \prod_{k=1}^{l} T_k(s')A_k(s'). \tag{4.30}$$

Since the algorithm satisfies detailed balance, it will asymptotically sample the distribution function desired, independently of T_k and π_k.

What we have not explained while deriving multilevel Metropolis is how to partition the levels, and the way the displacements are proposed, since this is a general algorithm. For the bisection method, this task is carried out by the Lévy construction. The bisection algorithm, which combines both techniques, is explained in the following section.

So far we have demonstrated that in general the multilevel Metropolis, and particularly bisection, asymptotically samples the required distribution function. However, we have not shown that the Lévy construction exactly samples the free particle density matrix. In order to do so we shall use a demonstration we already did for the staging algorithm. We observed in equation (4.16) that with the proposed staging coordinates the free particle density matrix can be sampled exactly. We also notice that bisection is equivalent in the finest level to staging since for $j = 2$ equation (4.19) becomes

$$x_1' = \frac{x_2 + x_0}{2} + \eta\sqrt{\frac{\hbar^2\tau}{2m}} \tag{4.31}$$

which is indeed the Lévy construction (4.26). Therefore, the Lévy construction also samples the free particle density matrix exactly.

4.3.3 The bisection algorithm

Bisection roughly consists in taking up a segment of a polymer chain (Ceperley and Pollock, 1992). The extremes of the segment are not to be moved. Then the bead in the middle of the extremes is assumed to be directly connected, with kinetic springs, to the extremes. In other words, it is assumed that there are no more beads, in the segment, other than the extremes and the middle one. A trial displacement for the middle bead is proposed. This is a coarse move that is Metropolis tested. If rejected the algorithm goes back to the beginning again. If accepted the middle bead goes to its new position, and two new segments are formed: the first bead of the original segment along with the middle bead, as well as the middle bead with the last one. Now, two new trial displacements for the new middle beads of the two new segments is proposed. The two displacements must be accepted together, otherwise they are rejected. The algorithm is applied recursively until we have moved all the beads in the segment (except the first and the last one which are fixed) or until a given proposed movement is rejected in which case we restore all the old positions, even those positions that have been accepted in previous steps, and we go over the whole algorithm again. On the other hand, if all the proposed displacements until the finest level are accepted, they are accepted as a whole. The number of steps in which this algorithm goes through from the coarsest to the finest move is called the level of the bisection method.

The coarsest level

We can illustrate the algorithm with a particular example (Chakravarty et al., 1998). Let us suppose we are moving a polymer ring with more than nine beads, and we have decided to use a bisection sampling of level three ($l = 3$). The number of beads contained in the polymer chain segment for $l = 3$ is $M_b = 2^3 + 1$, but since the extremes will remain unmoved only 7 beads are to be displaced. First we choose a bead of the polymer ring at random. That is going to be one of the extremes of the polymer chain segment, namely r_0 (Fig. 4.1, Line 1).

Now, we must go from the coarsest to the finest level. We will start at level $l = 3$ and then go downwards until we reach $l = 1$. We must choose the middle bead, in this case r_4. The distance between two contiguous beads is given by τ. For $l = 3$ we assume to have a reduced chain, that is, we take one every other four beads (for a bisection of any given level l, we should take one every 2^{l-1} beads). Notice that the distance between r_0 and r_4 is 4τ, as well as the distance between r_4 and r_8 (Fig. 4.1, Line 2).

We propose a trial move of the middle bead, from r_4 to r'_4 (Fig. 4.1, Line 3). The displacement of r_4 is proposed using

$$\mathbf{r}'_4 = \frac{1}{2}(\mathbf{r}_0 + \mathbf{r}_8) + \boldsymbol{\eta}\sqrt{4\lambda\tau}, \tag{4.32}$$

where $\boldsymbol{\eta}$ is a three dimensional normally distributed random vector with zero mean and unit variance, and $\lambda = \hbar^2/2m$. Within a real code (4.32) cannot be used in that raw form due to periodic boundary conditions. In order to take them into account the movement is proposed in the following way

$$\begin{aligned} \mathbf{r}_{0,4} &= \mathbf{r}_4 - \mathbf{r}_0 \\ \mathbf{r}_{4,8} &= \mathbf{r}_8 - \mathbf{r}_4 \end{aligned} \tag{4.33}$$

$$\begin{aligned} \mathbf{r}_{0,4} &\leftarrow \mathbf{r}_{0,4} - L * \texttt{ANINT}(\mathbf{r}_{0,4}/L) \\ \mathbf{r}_{4,8} &\leftarrow \mathbf{r}_{4,8} - L * \texttt{ANINT}(\mathbf{r}_{4,8}/L) \end{aligned} \tag{4.34}$$

with `ANINT(A)` a function that returns the nearest integer number to `A`. L is the side of the simulation box which goes from $-L/2$ to $L/2$. If the side would expand from 0 to L the function `ANINT(A)` could not be used that way. Notice that $\mathbf{r}$ is a three dimensional vector thus every equation really means three equations, one for every space direction. Then we define

$$\begin{aligned} \mathbf{r}^4_{0,4} &= \mathbf{r}_4 - \mathbf{r}_{0,4} \\ \mathbf{r}^4_{4,8} &= \mathbf{r}_4 + \mathbf{r}_{4,8} \end{aligned} \tag{4.35}$$

and finally the new trial position is given by

$$\mathbf{r}'_4 = \frac{1}{2}(\mathbf{r}^4_{0,4} + \mathbf{r}^4_{4,8}) + \boldsymbol{\eta}\sqrt{4\lambda\tau}. \tag{4.36}$$

The proposed displacement exactly samples the kinetic part of the action at $2^{l-1}\tau$, therefore we should only evaluate the change in the potential part of the action. Of course, we will calculate the change in the action using a time step of 4τ.

$$\delta A_{4\tau} = A(r_i^\alpha; 4\tau) - A(r_i'^\alpha; 4\tau). \tag{4.37}$$

Then the proposed movement is Metropolis tested by comparing

$$\Delta_{4\tau} = \min[1, \exp(-\delta A_{4\tau})] \tag{4.38}$$

with a uniformly distributed random number in $[0, 1]$. If the movement is accepted the algorithm proceeds to the next level. If not, the trial position is rejected, the original position of r_4 is restored, and the algorithm moves forward to the next atom, in order to go all over again.

The key point is that the middle bead (in our example r_4) has been displaced a large distance from its original position, since we have not used the exact action with imaginary time τ, but an action with a time step 4τ. Therefore, we are not sampling at the required low temperature, but at a temperature four times higher. For, our approximate polymer ring is much freer to move and to explore broader regions of configuration space.

Medium levels

Let us suppose the coarsest level has been accepted (in our example $l = 3$), so we are able to move forward to a finer level (in the example to $l = 2$). The proposed movements, for the particular example we are developing (Fig. 4.1, Line 4), are

$$\begin{aligned} \mathbf{r}_2' &= \frac{1}{2}(\mathbf{r}_0 + \mathbf{r}_4') + \boldsymbol{\eta}\sqrt{2\lambda\tau} \\ \mathbf{r}_6' &= \frac{1}{2}(\mathbf{r}_4' + \mathbf{r}_8) + \boldsymbol{\eta}\sqrt{2\lambda\tau}. \end{aligned} \tag{4.39}$$

Of course, due to periodic boundary conditions the equations (4.39) can not be applied directly and the same procedure used for (4.32) must be followed. Acceptance is based on

$$\Delta_{2\tau} = \min[1, \exp(-\delta A_{2\tau} + \delta A_{4\tau})]. \tag{4.40}$$

Notice we are sampling now at a temperature 2τ, which yields an approximate action to the real temperature. At a first glance we might think that at a temperature 2τ the acceptance probability is lower than

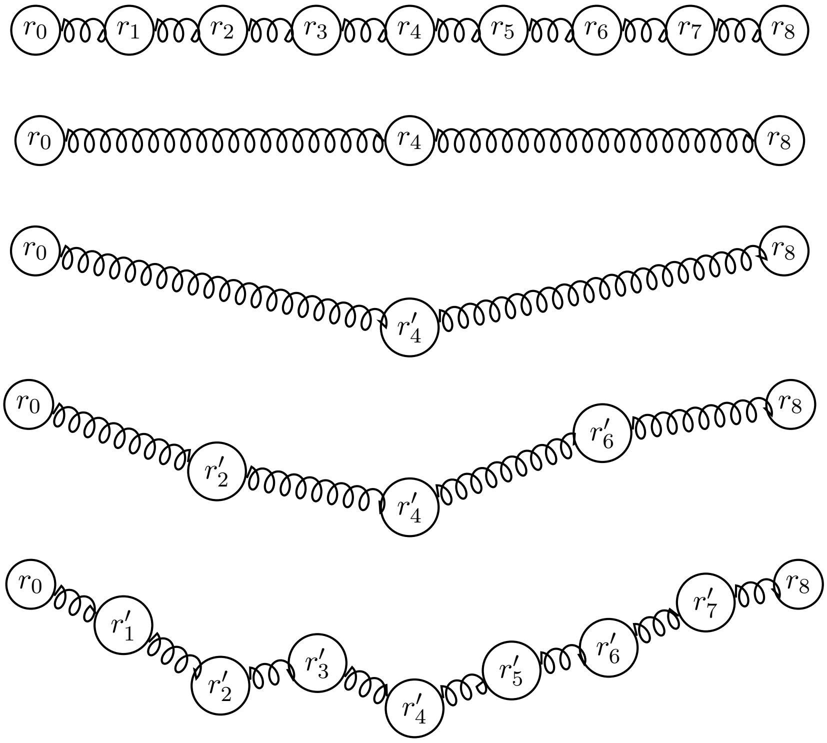


Fig. 4.1: Graphical explanation of bisection algorithm

at a temperature 4τ. However, within bisection this is not true, since the Brownian bridge we are building between r_0 and r'_4 (and another between r'_4 and r_8) is prone to be accepted since r'_4 has been placed in a favourable position.

Also notice that the Metropolis criteria used in (4.40) takes only into account the change in action produced by the displacements of the two beads moved; therefore, the contribution to the action of the previous levels is discarded.

As before, if this level is accepted we will go to the next level. Should not, we return to the very beginning of the algorithm, that is, we restore all the movements performed in this level—and all previous levels—with their original positions.

The finest level

Finally, in level one we evaluate the exact action, using the time step τ and not an approximate one with a directly proportional imaginary time step $(2^{l-1}\tau)$. Now, all the beads that have not been moved are displaced, except the beginning (r_0), and the end (r_8) of the polymer segment (Fig. 4.1, Line 5). The proposed displacements are

$$\begin{aligned}
\mathbf{r}'_1 &= \frac{1}{2}(\mathbf{r}_0 + \mathbf{r}'_2) + \boldsymbol{\eta}\sqrt{\lambda\tau} \\
\mathbf{r}'_3 &= \frac{1}{2}(\mathbf{r}'_2 + \mathbf{r}'_4) + \boldsymbol{\eta}\sqrt{\lambda\tau} \\
\mathbf{r}'_5 &= \frac{1}{2}(\mathbf{r}'_4 + \mathbf{r}'_6) + \boldsymbol{\eta}\sqrt{\lambda\tau} \\
\mathbf{r}'_7 &= \frac{1}{2}(\mathbf{r}'_6 + \mathbf{r}_8) + \boldsymbol{\eta}\sqrt{\lambda\tau}. \qquad (4.41)
\end{aligned}$$

Level one is accepted based on

$$\Delta_\tau = \min[1, \exp(-\delta A_\tau + \delta A_{2\tau})]. \qquad (4.42)$$

Only if level one is accepted the old path is definitively restored with the new one. If it is rejected the algorithm goes to the beginning and all the unprimed positions are kept.

After discounting the approximate actions used in all but the finest level, equations (4.40) and (4.42), we sample indeed the exact action, that is

$$\left(e^{-\delta A_{4\tau}}\right)\left(e^{-\delta A_{2\tau}}e^{\delta A_{4\tau}}\right)\left(e^{-\delta A_{\tau}}e^{\delta A_{2\tau}}\right) = e^{-\delta A_{\tau}}. \tag{4.43}$$

Estimation of thermal averages is only done when a movement, in any level, is rejected, or when the finest level is accepted. There is no calculation of thermal averages while accepting intermediate levels. For this reason we are sampling the exact action. We have only used the approximate actions for constructing the path, but the real test is done with the real action. The advantage of this algorithm is that the major part of the computational effort is done in the finest level (more movements of beads and more actions to be calculated) which is the level with more chances of being accepted. On the other hand, in the coarsest level the chances to accept the movement are very low, but the computational effort is very small (only one bead moved and one action calculated). Nevertheless, if the movement is accepted it represents a big jump into a favourable region and, therefore, the chances that subsequent moves are accepted increase.

In order to make clearer the algorithm described above, we have appended a general level bisection code. One can construct a bisection code for a given level, which is easier than a general level one. However, a general level code is much more succinct, flexible, and elegant. The code, along with its explanation, can be found in Appendix B.

4.3.4 Useful remarks

When using the traditional sampling, there is a free parameter that can be adjusted in order to control the acceptance. This parameter bounds the length of the displacement proposed using a random number. In fact, we use this method when proposing centre of mass displacements. However, there is no such free parameter in a bisection algorithm. The only parameter that can be adjusted is the level, and is an integer number. Furthermore, the maximum level that can be chosen is conditioned by the number of beads used in a given simulation. For instance, were we performing a simulation using chains of sixteen beads, the maximum level allowed would be three. Someone may wonder if a free parameter like l, that gives such a little freedom, delivers sufficient flexibility and power. The answer is yes. A level three bisection yields roughly a twenty per cent global acceptance, which is far below the fifty per cent usually expected in any sampling. However, the movements accepted are considerably larger than that ones that would be obtained with a free parameter. Moreover, the computational effort spent on all the movements wasted is very small.

One may also wonder if using the higher possible level for the given polymer length chosen is a good idea. We quote the following excerpt from (Ceperley and Pollock, 1992): "Running at even larger levels [more than $l = 3$] produces results which converge much more quickly even though the acceptance ratios are quite small." Despite the previously ascertained phrase, the tests we have carried out show that level three is the more advisable. Furthermore, Ceperley's collaborators assure that for any chain length above eight beads they only use level three in their calculations (Gordillo, private communication).

Bisection is useful if one wants to simulate closed polymer rings, that is those which comply the closeness condition $x_1 \equiv x_{M+1}$. If one has to sample an open polymer ring, bisection is not suitable due to technical problems. This follows from the fact that for the last 2^l beads of the chain there are no sufficient beads right hand in order to bisect the segment properly. A possible case in which one would desire open chains are single particle density matrices calculations (all but one polymer are closed).

4.3.5 Unbiased bisection

The action calculated in all the levels, but the finest one, is an approximate action. It is a legitimate concern to wonder whether using approximate actions for setting up a movement would lead to a bias. Well, as a matter of fact, this is not the case, because of the action used for finally accepting the movement is the exact one for the singled–out temperature. Therefore, if we would have erroneously accepted a movement in a coarser level—due to the fact we were not using the appropriate action—it would be rejected in a finer, or the finest, level. Hence, there are no unacceptable movements that can get through the whole algorithm. Remember that a rejection at any level implies to go back to the very beginning and to restore the old positions again.

However, there is a more profound and subtle bias within bisection. It comes not from the accepted movements, but from the rejected movements. Let us suppose we propose a very awkward movement. In most cases it would be rejected by a coarse action, as well as by the finest one. Notwithstanding, it is not impossible to happen that a coarser action rejects that displacement and a more refined one would accept it. But, as soon as a proposed displacement is rejected the algorithm goes to the beginning, reestablishing all positions, and never giving the chance to the finest action to really Metropolis–test the proposed change in action.

Although the previous explanation is not found anywhere in literature, there is a paper (Ceperley and Pollock, 1992) that gives some hint of the bias, stating that "as the density matrices approach the exact density matrix and as the size of the move approaches the total system size, π_k approaches a probability distribution", with π_k the action used for level k. The true meaning is that π_k, even at the finest level, asymptotically approaches the action for that temperature, but the lost opportunities of would–be–accepted levels keep it apart from the exact value.

In light of this, we coded a modified bisection algorithm, that went through the whole process, even though a proposed movement in a coarser level was rejected, in order to check if at the end could be accepted. Of course, this algorithm is by no means an optimal approach from the CPU time point of view; however, we wanted to check the validity of the results yielded by bisection. The result was that the bias was so small that it was always within error bars, even for very long runs. We will show these results in a forthcoming chapter devoted to testing the methods.

5. PROPERTIES

The theory and algorithms developed hitherto allow us to write down a PIMC code. We learned how to simulate a quantum liquid at a finite temperature, which means that we can make a discrete representation of a real quantum system and we also know how to evolve that representation in order to reach equilibrium. Now we need to know what is going on inside the liquid, which properties does it have, and how it behaves. All in all, this is a goal of a simulation, to grasp information without going into a laboratory.

Each section of this chapter is devoted to a physical property. The properties that our codes include are: energy, momentum distribution, single particle density matrix, radial distribution function, and the structure factor. The sections within the chapter are divided in the following way: Section §5.1 is devoted to the energy. The different energy estimators are deeply rooted in the action, therefore, for every action tested we shall derive the energy estimators. The single–particle density matrix will be treated in §5.2. The momentum distribution, as well as the Compton profile, are obtained from the single–particle density matrix. They are also studied in §5.2. Section §5.3 is devoted to the radial distribution function. Finally, the structure factor is studied in section §5.4.

5.1 Energy

Undiscerning choices of energy estimators can mar this struggle for efficient codes. The choice of energy estimator can affect largely the variance, and despite of an optimal sampling method and an improved action, CPU times may become extremely long.

In our pursue for an optimal energy estimator we have tried the thermodynamic and the virial energy estimators, along with all the possible combinations with the different actions tested.

Due to the dependence of the energy estimators on the action, we shall follow the same order of Chapter 3. Thus, we shall start with the harmonic oscillator and then we will move onto energy estimators derived for real systems.

5.1.1 Thermodynamic estimator and primitive action

In preparation for deriving the thermodynamic estimator we start from the definition of the average energy

$$< E >= -\frac{\partial \ln Z}{\partial \beta} = \frac{1}{Z}\mathrm{Tr} H e^{-\beta H} = \frac{\sum_n E_n e^{-\beta E_n}}{\sum_n e^{-\beta E_n}}, \qquad (5.1)$$

where E_n are the eigenvalues of the Hamiltonian.

Acting (5.1) upon

$$S_0 = \sum_{\alpha=1}^{M}\left[\frac{m}{2}\frac{(x_\alpha - x_{\alpha-1})^2}{\hbar^2\tau} + \tau V(x_\alpha)\right], \qquad (5.2)$$

and also using

$$\frac{\partial}{\partial \beta} = \frac{1}{M}\frac{\partial}{\partial \tau}, \qquad (5.3)$$

it is possible to write up the thermodynamic estimator of the energy for the primitive approximation to the action in the case of a one–dimensional single–particle as

$$< E >= \frac{M}{2\beta} - \frac{m}{2\hbar^2 M}\sum_{\alpha=1}^{M}\left(\frac{x_\alpha - x_{\alpha-1}}{\tau}\right)^2 + \frac{1}{M}\sum_{\alpha=1}^{M} V(x_\alpha), \qquad (5.4)$$

where $< E >$ is the energy per atom. Observe that, in the case of $M = 1$, equation (5.4) returns the classical value of the energy. The exact quantum value of the energy is obtained when $M \to \infty$.

Let us inspect the form of (5.4). The last term clearly yields the potential energy. The first term is the classical value of the kinetic energy (if the formula were derived in three dimensions the term would be $3M/2\beta$) times the number of beads. The second term is a negative correction to the first term. Therefore, it is not possible to separate the kinetic energy in a classical term plus a quantum correction. This is the main reason for the large variance associated with the thermodynamic energy estimator. This is not the case for the virial energy estimator that we will consider afterwards.

For completeness we introduce the thermodynamic energy estimator for a single–particle for any dimensionality

$$< E >= \frac{DM}{2\beta} - \frac{mM}{2\hbar^2\beta^2}\left\langle \sum_{\alpha=1}^{M} (\mathbf{r}_{\alpha+1} - \mathbf{r}_{\alpha})^2 \right\rangle + \frac{1}{M}\left\langle \sum_{\alpha=1}^{M} V(\mathbf{r}_{\alpha}) \right\rangle , \quad (5.5)$$

with D the number of dimensions.

5.1.2 Thermodynamic estimator and Li–Broughton action

In the same way we found the thermodynamic energy estimator for the primitive action, it is possible to do the same for the Li–Broughton action.

We start with the partition function of a three–dimensional system, given by

$$Z = \left(\frac{mP}{2\pi\hbar^2\beta}\right)^{3M/2} e^{-\beta V_{\text{eff}}} \quad (5.6)$$

where V_{eff} is the effective potential and M the number of beads of the polymer chain. We have not written the M integrals that appear in the partition function explicitly since we will derive (5.6) and due to the linearity of the integral operator the derivatives can go inside of the integral symbol.

$$\begin{aligned} E &= -\frac{1}{Z}\frac{\partial Z}{\partial \beta} \\ &= -\frac{1}{Z}\left\{ e^{-\beta V_{\text{eff}}} \left(\frac{mM}{2\pi\hbar^2\beta}\right)^{3M/2} \left(-\frac{3M}{2}\right) \beta^{-\frac{3M}{2}-1} \right. \\ &+ \left. e^{-\beta V_{\text{eff}}} \left(\frac{mM}{2\pi\hbar^2\beta}\right)^{3M/2} \left[-V_{\text{eff}}\, e^{-\beta V_{\text{eff}}} - \beta \left(\frac{\partial V_{\text{eff}}}{\partial \beta}\right) e^{-\beta V_{\text{eff}}} \right] \right\} \\ &= \frac{3M}{2\beta} + V_{\text{eff}} + \beta\left(\frac{\partial V_{\text{eff}}}{\partial \beta}\right) . \end{aligned} \quad (5.7)$$

However, from Section §3.3.2 we know that Li–Broughton's effective potential is given by

$$\begin{aligned} V_{\text{eff}} &= \frac{mM}{2\hbar^2\beta^2}\sum_{\alpha=1}^{M}\left(\mathbf{r}_{\alpha+1}-\mathbf{r}_{\alpha}\right)^2+\frac{1}{M}\sum_{\alpha=1}^{M}V(\mathbf{r}_{\alpha}) \\ &+ \frac{\beta^2\hbar^2}{24M^3m}\sum_{\alpha=1}^{M}\left(\nabla_{\alpha}V(\mathbf{r}_{\alpha})\right)^2, \end{aligned} \tag{5.8}$$

therefore, the derivative is

$$\frac{\partial V_{\text{eff}}}{\partial\beta}=-\frac{mM}{\hbar^2\beta^3}\sum_{\alpha=1}^{M}\left(\mathbf{r}_{\alpha+1}-\mathbf{r}_{\alpha}\right)^2+\frac{\beta\hbar^2}{12M^3m}\sum_{\alpha=1}^{M}\left(\nabla_{\alpha}V(\mathbf{r}_{\alpha})\right)^2. \tag{5.9}$$

Using the previous equations, the thermodynamic estimator of the energy for a Li–Broughton corrected potential yields, in the single–particle three–dimensional case

$$\begin{aligned} < E_{\text{LB}} > &= \frac{3M}{2\beta}-\frac{mM}{2\hbar^2\beta^2}\left\langle\sum_{\alpha=1}^{M}(\mathbf{r}_{\alpha+1}-\mathbf{r}_{\alpha})^2\right\rangle \\ &+ \frac{1}{M}\left\langle\sum_{\alpha=1}^{M}\left\{V(\mathbf{r}_{\alpha})+\frac{\beta^2\hbar^2}{8M^2m}\left[\nabla_{\alpha}V(\mathbf{r}_{\alpha})\right]^2\right\}\right\rangle. \end{aligned} \tag{5.10}$$

Close inspection of (5.10) shows that it is the same expression than (5.5) but for a corrective factor.

5.1.3 Thermodynamic estimator of a Li–Broughton corrected harmonic oscillator potential

An immediate application of the Li–Broughton correction (3.27) is to find a better potential for estimating the energy of the harmonic oscillator

$$V_{\text{LBH}}=\frac{m\omega^2}{2}\left(1+\frac{\beta^2\hbar^2\omega^2}{12M^2}\right)x^2. \tag{5.11}$$

In order to derive the energy estimator for the V_{LBH} potential we shall use expression (5.10) substituting V by the harmonic potential $V_{\text{H}} = m\omega^2 x^2/2$. Notice that the same result might be achieved using equation (5.14) that will be derived below and inserting V_{LBH} as the potential V. The energy estimator for the Li–Broughton correction of the harmonic oscillator is

$$\langle E_{\text{LBH}} \rangle = \left\langle \frac{m\omega^2}{2M} \left(1 + \frac{\beta^2 \hbar^2 \omega^2}{4M^2} \right) \sum_{\alpha=1}^{M} {x_\alpha}^2 \right\rangle . \tag{5.12}$$

5.1.4 *Thermodynamic estimator of a harmonic potential variationally improved*

In section §3.2.5 we found the action for the harmonic oscillator potential using Feynman's variational principle, now we shall find the thermodynamic energy estimator for that action. We know that

$$\langle H \rangle = -\frac{1}{M} \frac{\partial \ln Z}{\partial \tau} . \tag{5.13}$$

Using (5.13) upon (3.37) we derive the thermodynamic energy estimator for the Feynman harmonic potential V_{FH}. First we obtain

$$\begin{aligned} \langle V \rangle &= -\frac{1}{M} \left\{ -\sum_{\alpha=1}^{M} \frac{\partial}{\partial \tau} [\tau V(x_\alpha, \tau)] \right\} = \\ & \frac{1}{M} \left\{ \sum_{\alpha=1}^{M} V(x_\alpha, \tau) + \tau \sum_{\alpha=1}^{M} \frac{\partial V(x_\alpha, \tau)}{\partial \tau} \right\} . \end{aligned} \tag{5.14}$$

Now we insert Feynman's harmonic potential in the previous equation, and we obtain the desired estimator for the potential contribution

$$\langle E_{\text{FH}} \rangle = \frac{1}{M} \left\langle \sum_{\alpha=1}^{M} \frac{m\omega^2}{2} {x_\alpha}^2 \right\rangle + \frac{\omega^2 \tau}{12} . \tag{5.15}$$

5.1.5 Thermodynamic estimator of the Li–Broughton correction applied to Feynman's variational principle for the harmonic oscillator

Obviously, what we are doing is following the same path we did when studying the action, that is explore all the possible combinations between Li–Broughton's correction and Feynman's variational principle. When we did that we did not state any formula for the energy, and this is precisely what we are doing now.

Using (3.38) we obtain

$$\langle E_{\mathrm{LBFH}} \rangle = \left\langle \frac{m\omega^2}{2M} \left(1 + \frac{\omega^2 \tau^2 M \hbar^2}{4} \right) \sum_{\alpha=1}^{M} {x_\alpha}^2 \right\rangle + \frac{\omega^2 \hbar^2 \tau}{12}. \tag{5.16}$$

5.1.6 Thermodynamic estimator of the Feynman's variational principle applied to the Li–Broughton correction for the harmonic oscillator

The last combination to be done is finding the thermodynamic estimator obtained by application of Feynman's variational principle upon a Li–Broughton corrected harmonic potential. In order to do that we use (3.39)

$$\begin{aligned} \langle E_{\mathrm{FLBH}} \rangle &= \left\langle \frac{m\omega^2}{2M} \left(1 + \frac{\omega^2 \hbar^2 \beta^2}{4M^2} \right) \sum_{\alpha=1}^{M} {x_\alpha}^2 \right\rangle \\ &+ \frac{\omega^2 \hbar^2 \beta}{12} \left(1 + \frac{\omega^2 \hbar^2 \beta^2}{16M} \right). \end{aligned} \tag{5.17}$$

The same process done in this, and sections above, can be easily repeated for the anharmonic oscillator.

5.1.7 Thermodynamic estimator of the energy for real systems

So far, we have derived the thermodynamic estimator of the energy for single–particle systems, and we have applied it to several actions. That process has been useful in preparation for the study of the harmonic oscillator. However, our ultimate goal is the study of real systems where many atoms are involved. From now on we shall deal with this kind of systems and, therefore, our formulæ will include labels for atoms (up to N atoms), and for beads (up to M beads).

At a first glance, inclusion of more than one atom, just implies averaging out the energy by the number of atoms. This is true only if there are no gradients involved in the energy estimator, which is the case of the thermodynamic estimator along with a primitive action. Notwithstanding, should there appear gradients, due to a Li–Broughton action or a virial estimator, some algebra must be performed.

Nevertheless, at this point, we shall only introduce the thermodynamic energy estimator for a primitive action. We already noticed that the thermodynamic estimator for a primitive action is made up by three terms, the last one being the potential energy, and the other two take account of the kinetic energy. Since the potential energy part is straight forward we just give the thermodynamic kinetic energy estimator for a primitive action (Barker, 1979), which yields

$$K^{\mathrm{T}} = \frac{3}{2}\frac{M}{\beta} - \frac{1}{N}\frac{Mm}{2(\beta\hbar)^2}\sum_{\alpha=1}^{M}\sum_{i=1}^{N}\left|\mathbf{r}_i^{\alpha} - \mathbf{r}_i^{\alpha+1}\right|^2 . \tag{5.18}$$

The superscript T stands for thermodynamic.

5.1.8 Li–Broughton's correction to the kinetic energy for a Lennard–Jones potential

In our simulations of real systems we do not use Lennard–Jones potentials, however, for the testing process this kind of potentials are useful for comparison purposes. Therefore, we introduce the derivation of the Li–Broughton correction to the Lennard–Jones potential.

We already know that Li–Broughton's correction to the total energy is given by

$$\left(\frac{\hbar\beta}{M}\right)^2 \sum_{i=1}^{N}\sum_{\alpha=1}^{M}\left(\nabla_{i\alpha}U(\mathbf{r}_{i\alpha})\right)^2 \frac{1}{8mM}, \tag{5.19}$$

where M is the number of beads, N the number of atoms, and

$$U(\mathbf{r}_{i\alpha}) = \sum_{j\neq i}^{N} u\left(\left|\mathbf{r}_{i\alpha} - \mathbf{r}_{j\alpha}\right|\right). \tag{5.20}$$

Now we apply the gradient operator on both sides of the previous equation

$$\nabla_{i\alpha} U(\mathbf{r}_{i\alpha}) = \sum_{j \neq i}^{N} \nabla_{i\alpha} u \left(|\mathbf{r}_{i\alpha} - \mathbf{r}_{j\alpha}| \right). \tag{5.21}$$

Substitution of the Lennard–Jones potential $u(r) = \epsilon[(\sigma/r)^{12} - (\sigma/r)^6]$ in (5.21) yields

$$\begin{aligned} \nabla_{i\alpha} u \left(|\mathbf{r}_{i\alpha} - \mathbf{r}_{j\alpha}| \right) &= \epsilon \sigma^{12} \frac{\partial}{\partial \mathbf{r}_{i\alpha}} \left(\frac{1}{|\mathbf{r}_{i\alpha} - \mathbf{r}_{j\alpha}|} \right)^{12} \\ &- \epsilon \sigma^{6} \frac{\partial}{\partial \mathbf{r}_{i\alpha}} \left(\frac{1}{|\mathbf{r}_{i\alpha} - \mathbf{r}_{j\alpha}|} \right)^{6}. \end{aligned} \tag{5.22}$$

Finally, inserting (5.22) in (5.19) gives the Li–Broughton correction to the kinetic energy for the Lennard–Jones potential

$$\begin{aligned} & \frac{\lambda \tau^2}{4} \sum_{i=1}^{N} \sum_{\alpha=1}^{M} \sum_{j \neq i}^{N} \frac{\epsilon^2}{|\mathbf{r}_{i\alpha} - \mathbf{r}_{j\alpha}|^2} \\ \times & \left[-12 \left(\frac{\sigma}{|\mathbf{r}_{i\alpha} - \mathbf{r}_{j\alpha}|} \right)^{12} + 6 \left(\frac{\sigma}{|\mathbf{r}_{i\alpha} - \mathbf{r}_{j\alpha}|} \right)^{6} \right]^2. \end{aligned} \tag{5.23}$$

5.1.9 The virial estimator

Even though the results of the tests performed with the thermodynamic energy estimator have not been shown yet, we must say that there is a large variance in the results obtained with it, which increases with M. Close inspection show that most of the variance accounts for the kinetic energy, while the potential energy contributes roughly with 10% of the total variance. In the light of these conclusions Herman et al. (1982) proposed a new energy estimator. They were looking for an estimator that would depend only on the potential energy and its derivatives. The virial theorem applied to path integrals was the response. The virial theorem states that the kinetic energy is proportional to the scalar product of position and the gradient of the potential, therefore, the total energy in one dimension becomes

$$E = V(x) + \frac{1}{2} x \frac{\partial V(x)}{\partial x}. \tag{5.24}$$

Equation (5.24) yields the virial estimator of the kinetic energy

$$K^{\mathrm{V}} = \frac{1}{2NM} \sum_{i=1}^{N} \sum_{\alpha=1}^{M} \mathbf{r}_i^{\alpha} \cdot \frac{\partial V\left(\mathbf{r}_i^{\alpha}\right)}{\partial \mathbf{r}_i^{\alpha}}. \tag{5.25}$$

Adding a constant boundary term to (5.25) reduces the variance and keeps it constant with the number of beads. If the term added is the centre of mass of the polymer ring we obtain the centroid virial estimator (Herman et al., 1982), (Glaesemann and Fried, 2002). However, same accuracy is obtained using any given bead of the polymer ring as the boundary term. The virial estimator in the form of (5.25) is seldom used, and the common practice is to name as virial estimator of the kinetic energy the following equation:

$$K^{\mathrm{V}} = \frac{3}{2}T + \frac{1}{2NM} \sum_{i=1}^{N} \sum_{\alpha=1}^{M} \left(\mathbf{r}_i^{\alpha} - \mathbf{r}_i^{M}\right) \cdot \frac{\partial V\left(\mathbf{r}_i^{\alpha}\right)}{\partial \mathbf{r}_i^{\alpha}}. \tag{5.26}$$

Notice that the kinetic energy is now obtained as the classical energy of the system, plus a corrective term. This is particularly useful when performing simulations. Let us suppose we start a simulation from a classical configuration, that is, with all the polymer rings collapsed, that has been thermalised classically. If we estimate the energy prior to any PIMC step using the virial estimator (5.26) we would obtain the classical value of the energy for that system, *i.e.,* $3/2T$, since the contribution of the corrective term will be zero (the value of $\left(\mathbf{r}_i^{\alpha} - \mathbf{r}_i^{M}\right)$ equals zero since all the chains are collapsed). As the simulation goes on, and the polymer rings begin to swell until they reach their characteristic length, the result of the kinetic energy will become closer to the expected quantum value. Moreover, the term $3/2T$ appearing in (5.26) helps also in reducing the variance since it is a large value, compared to the other term in the same equation, with zero variance because of it is a constant.

The situation is not the same if the thermodynamic estimator were used. If all the polymer rings start from a collapsed situation and the kinetic energy is estimated using equation (5.18), the result would be the classical energy times the number of beads used. For a typical simulation temperature this would be as much as a hundred times the classical value of the energy. Obviously, the starting value of the energy is farther from the expected value in the case of the thermodynamic estimator, than in the case of the virial estimator.

Furthermore, once the polymer rings have reached their characteristic length for the temperature simulated, the thermodynamic estimator still yields larger variances than the virial estimator.

The scalar product in (5.26) can be rewritten as

$$\sum_{a=1}^{3}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}\left(r_i^\alpha - r_i^M\right)^a \cdot \frac{\partial V(r_i^\alpha)}{\partial \left(r_i^\alpha\right)^a}, \tag{5.27}$$

where a labels each one of the three Cartesian coordinates. On the other hand, we already know that the potential in a many particle system is calculated between equally labelled beads of different atoms, therefore, the potential appearing in (5.27) represents indeed

$$V(r_i^\alpha) = \sum_{j=1}^{N} V(r_{ij}^\alpha). \tag{5.28}$$

And now applying the chain rule to the potential we obtain

$$\frac{\partial V(r_i^\alpha)}{\partial \left(r_i^\alpha\right)^a} = \sum_{j=1}^{N} \frac{\partial V(r_{ij}^\alpha)}{\partial \left(r_i^\alpha\right)^a} = \sum_{j=1}^{N} \frac{\partial V(r_{ij}^\alpha)}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial \left(r_i^\alpha\right)^a}, \tag{5.29}$$

with

$$r_{ij} = \sqrt{\sum_{i=1}^{3}\left(r_i^\alpha - r_j^\alpha\right)^{a^2}} \tag{5.30}$$

$$\frac{\partial r_{ij}}{\partial \left(r_i^\alpha\right)^a} = \frac{2\left(r_i^\alpha - r_j^\alpha\right)^a}{2r_{ij}^\alpha} = \frac{\left(r_{ij}^\alpha\right)^a}{r_{ij}^\alpha}. \tag{5.31}$$

Substitution in the expression for the virial estimator yields

$$K^{\mathrm{V}} = \sum_{i=1}^{N}\sum_{\alpha=1}^{M}\sum_{a}^{3}\left(r_i^\alpha - r_i^M\right)^a \sum_{j=1}^{N} \frac{\left(r_i^\alpha - r_j^\alpha\right)^a}{r_{ij}^\alpha} \frac{dV(r_{ij})}{dr_{ij}^\alpha} \tag{5.32}$$

which is the virial estimator of the kinetic energy for a primitive action written in a more suitable form for coding.

5.1.10 Virial and thermodynamic estimators using a Li–Broughton action

Equations (5.18) and (5.32) give the thermodynamic and virial estimators of the kinetic energy for a primitive action, in the form that will be used for coding, respectively.

The same job we did for the primitive action is going to be carried out for the Li–Broughton action, that is we are going to give the thermodynamic and virial estimators for that action. However, the derivation of suitable forms for coding is more elaborated, therefore, we just give the useful formulæ and in Appendix D the full derivation is presented.

Let us start from the expression with Li–Broughton's for the virial estimator of the energy

$$\begin{aligned} E &= \frac{3}{2}T + \frac{1}{NM}\left\langle \sum_{i=1}^{N}\sum_{\alpha=1}^{M}\left\{ V(\mathbf{r}_{i\alpha}) + \frac{\hbar^2\beta^2}{8mM^2}[\nabla_{i\alpha}V(\mathbf{r}_{i\alpha})]^2 \right\} \right\rangle \\ &+ \frac{1}{2NM}\left\langle \sum_{i=1}^{N}\sum_{\alpha=1}^{M}(\mathbf{r}_{i\alpha} - \mathbf{R}_i)\cdot\nabla_{i\alpha}\left\{ V(\mathbf{r}_{i\alpha}) + \frac{\hbar^2\beta^2}{24mM^2}[\nabla_{i\alpha}V(\mathbf{r}_{i\alpha})]^2 \right\} \right\rangle , \end{aligned} \tag{5.33}$$

with $\mathbf{R}_i$ the i–th atom centre of mass. The first term on the right hand side amounts the classical kinetic energy. The second term is the potential energy, while the third one is the quantum correction to the kinetic energy. The previous equation might be expanded into

$$\begin{aligned} E &= \frac{3}{2}T + \frac{1}{NM}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}V(\mathbf{r}_{i\alpha}) \\ &+ \frac{1}{N}\frac{\hbar^2\beta^2}{8mM^3}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}[\nabla_{i\alpha}V(\mathbf{r}_{i\alpha})]^2 \\ &+ \frac{1}{2NM}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}(\mathbf{r}_{i\alpha} - \mathbf{R}_i)\cdot\nabla_{i\alpha}V(\mathbf{r}_{i\alpha}) \\ &+ \frac{1}{N}\frac{\hbar^2\beta^2}{48mM^3}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}(\mathbf{r}_{i\alpha} - \mathbf{R}_i)\cdot\nabla_{i\alpha}[\nabla_{i\alpha}V(\mathbf{r}_{i\alpha})]^2 \end{aligned} \tag{5.34}$$

From the previous study of the action, equation (3.49), we know that $\sum_{i=1}^{N}\sum_{\alpha=1}^{M}[\nabla_{i\alpha}V(\mathbf{r}_{i\alpha})]^2$ can be translated into a scalar product of forces, given by

$$\frac{\beta^2\hbar^2}{8M^3m}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}\mathbf{F}_{i\alpha}\cdot\mathbf{F}_{i\alpha}, \tag{5.35}$$

where $F_{i\alpha}$ is defined in equation (3.50). This is a useful expression for the kinetic energy for programming purposes when dealing with three dimensional many particle systems.

We also have shown an alternative way of representing the fourth term of (5.34) which is more suitable for coding purposes. Thus, the only term that needs a special treatment is the fifth one. Let us define the tensors

$$T(i)_a^b = \sum_{j\neq i}\left[\left(\frac{\delta_a^b}{|\mathbf{r}_{ij}|} - \frac{r_{ij}^b r_{ij\,a}}{|\mathbf{r}_{ij}|^3}\right)\frac{dV(\mathbf{r}_{ij})}{d|\mathbf{r}_{ij}|} + \frac{r_{ij}^b r_{ij\,a}}{|\mathbf{r}_{ij}|^2}\frac{d^2V(\mathbf{r}_{ij})}{d|\mathbf{r}_{ij}|^2}\right] \tag{5.36}$$

and

$$F(i)_b = \sum_{k\neq i}\frac{r_{ik\,b}}{r_{ik}}\frac{dV(\mathbf{r}_{ij})}{d|\mathbf{r}_{ij}|}. \tag{5.37}$$

Provided with those definitions, the thermodynamic estimator of the kinetic energy with the Li–Broughton action can be written as (Weht et al., 1998)

$$\begin{aligned} < K_{\mathrm{T}}^{\mathrm{LB}} > &= \frac{3MT}{2N} - \frac{mMT^2}{2\hbar^2N}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}(\mathbf{r}_{i\,\alpha+1}-\mathbf{r}_{i\,\alpha})^2 \\ &+ \frac{\hbar^2}{24M^3T^2mN}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}\mathbf{F}_{i\alpha}\cdot\mathbf{F}_{i\alpha}, \end{aligned} \tag{5.38}$$

and the potential energy

$$\begin{aligned} < V_{\mathrm{T}}^{\mathrm{LB}} > &= \frac{1}{NM}\sum_{\alpha=1}^{M}\sum_{i<j}V(\mathbf{r}_{ij}^{\alpha}) \\ &+ \frac{\hbar^2}{12M^3T^2mN}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}\mathbf{F}_{i\alpha}\cdot\mathbf{F}_{i\alpha}. \end{aligned} \tag{5.39}$$

Therefore the total thermodynamic energy estimator using a Li–Broughton action is:

$$\begin{aligned} < E_{\mathrm{T}}^{\mathrm{LB}} > &= \frac{3MT}{2N} - \frac{mMT^2}{2\hbar^2 N} \sum_{i=1}^{N} \sum_{\alpha=1}^{M} (\mathbf{r}_{i\,\alpha+1} - \mathbf{r}_{i\,\alpha})^2 + \frac{1}{NM} \sum_{\alpha=1}^{M} \sum_{i<j} V(\mathbf{r}_{ij}^{\alpha}) \\ &+ \frac{\hbar^2}{8M^3T^2mN} \sum_{i=1}^{N} \sum_{\alpha=1}^{M} \mathbf{F}_{i\alpha} \cdot \mathbf{F}_{i\alpha}. \end{aligned} \quad (5.40)$$

On the other hand, the virial estimator of the total energy using the Li–Broughton action is:

$$\begin{aligned} < E_{\mathrm{V}}^{\mathrm{LB}} > &= \frac{3}{2}T + \frac{1}{2MN} \sum_{i=1}^{N} \sum_{\alpha=1}^{M} (r_{i\alpha}^{a} - R_{i}^{a}) F(i,\alpha)_a \\ &+ \frac{1}{NM} \sum_{\alpha=1}^{M} \sum_{i<j} V(\mathbf{r}_{ij}^{\alpha}) + \frac{\hbar^2}{8M^3T^2mN} \sum_{i=1}^{N} \sum_{\alpha=1}^{M} \mathbf{F}_{i\alpha} \cdot \mathbf{F}_{i\alpha} \\ &+ \frac{\hbar^2}{24M^3NmT^2} \sum_{i=1}^{N} \sum_{\alpha=1}^{M} (r_{i\alpha}^{a} - R_{i}^{a})\, T(i,\alpha)_a^b\, F(i,\alpha)_a. \end{aligned} \quad (5.41)$$

It is easily seen how equations (5.40) and (5.41) become the thermodynamic and the virial energy estimators for a primitive action, respectively, if only the first three terms of each equations are kept.

5.1.11 Tail corrections to the potential energy

The code we have constructed simulates the quantum liquid inside a three–dimensional box of length L. Periodic boundary conditions have been implemented (Frenkel and Smit, 1996). Provided that, potential interaction ranging farther away from $L/2$ are neglected. In addition, interactions closer than $L/2$ use the nearest periodic image to be calculated.

Due to the neglected interactions there is a systematic error in the potential energy, since the potential is not rigorously zero for $r > L/2$.

However, that error can be estimated, provided the radial distribution function ($g(r)$) becomes approximately 1 for any $r > L/2$. Once the error is estimated, it can be added to the result of the potential energy in order to get a tail corrected result.

The tail correction to the potential energy is

$$\frac{< V >}{N} = \frac{1}{2}\rho \int drg(r)V(r), \tag{5.42}$$

with ρ the density. Once the radial distribution function gets to $L/2$, then $g(r)$ should be approximately 1, therefore,

$$\left(\frac{< V >}{N}\right)_T = \frac{1}{2}\rho \int_{L/2}^{\infty} drV(r) = \frac{1}{2}\rho 4\pi \int_{L/2}^{\infty} r^2 drV(r). \tag{5.43}$$

Only the attractive part is taken into account when integrating (5.43), which in a Lennard–Jones potential amounts σ^6/r^6. Substitution and integration yields,

$$\left(\frac{< V >}{N}\right)_T = -\frac{64}{3L^3}\pi\sigma^6\epsilon\rho. \tag{5.44}$$

This result is only valid if a Lennard–Jones potential is used.

Also notice that equation (5.43) yields bounded results only if the potential part decays more rapidly than r^{-3}, otherwise the tail corrections cannot be calculated that way.

The values of the potential energy reported in our tables for real systems simulations have that correction included.

5.2 Momentum distribution

The momentum distribution is an observable of major importance. Experimental momentum distribution measurements are done via neutron scattering. In order to extract results the impulse approximation is assumed. In the impulse approximation the interaction of an atom with its neighbours is neglected, and the atom is supposed to recoil freely. However, this is not rigorously true, and the resulting curves are broadened, hiding important fine details (Sears, 1969), (Woods and Sears, 1977), (Sosnick et al., 1989).

The importance of the momentum distribution spans beyond superfluidity. In classical theory it is given by a Maxwellian with width equal to $k_{\rm B}T$, but in quantum mechanics it is no longer so. When lowering temperature, quantum effects appear well before the lambda transition, and a single classical Maxwellian does not adjust anymore to the momentum distribution. Therefore, the momentum distribution gives account of the departure from the classical regime (Brualla et al., 2002). Furthermore, the value of the atomic kinetic energy can be inferred from the momentum distribution, allowing cross–checking with the kinetic energy estimators.

It is, so, of vital importance accurate measurements of the momentum distribution, and due to the large experimental errors inherent to neutron scattering, cross–checking with simulations acquires even more relevance.

In this section we show how to calculate the momentum distribution, and related quantities, such as the single–particle density matrix, or the Compton profile, from the PIMC point of view. Up until now there were in literature two different methods available for calculating the momentum distribution. Due to the importance of the momentum distribution, we created a novel method for its calculation, that surmounts some liabilities present in existing methods. A detailed description of our algorithm, the *trail method*, is given herein.

5.2.1 The single–particle density matrix

Our PIMC algorithm is performed in position space. However, in quantum mechanics positions and momenta have an equal status since they are conjugate variables. The density matrix in momentum space is

$$\hat{\rho}(p,p';\beta) = \sum_{\alpha} \hat{\phi}_\alpha(p) e^{-\beta E_\alpha} \hat{\phi}^*_\alpha(p'), \tag{5.45}$$

with $\hat{\phi}_\alpha(p)$ the eigenfunctions of the Hamiltonian in momentum space. The Fourier transform of these eigenfunctions are the eigenfunctions in position space,

$$\hat{\phi}_\alpha(p) = \frac{1}{(2\pi\hbar)^2} \int dx e^{ipx/\hbar} \phi_\alpha(x). \tag{5.46}$$

It is possible to find out the momentum distribution calculating the single–particle density matrix, and then, Fourier transform it in order to get the momentum distribution.

We define the single–particle density matrix by

$$n(\mathbf{r}) = Z^{-1} \int d\mathbf{r}_1 d\mathbf{r}_2 \cdots d\mathbf{r}_P < \mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_P | \exp(-\beta \mathcal{H}) | \mathbf{r}_1 + \mathbf{r}, \mathbf{r}_2, \ldots, \mathbf{r}_P >, \tag{5.47}$$

being Z the partition function. From (5.47) we notice that the off–diagonal part of the density matrix is required. Remember that in energy estimations we needed the trace, that is, only diagonal terms of the density matrix, but the situation here is different. In the polymer isomorphism language off–diagonal terms imply open chains.

Open chain algorithm

A direct interpretation of equation (5.47) is to perform a simulation with $N-1$ closed polymers rings and 1 open. In order to do so, we choose atom $i = 1$ as the one to keep open and we introduce an additional variable $\mathbf{r}'_{1\,1}$. During the simulation $\mathbf{r}'_{1\,1}$ is moved and Metropolis tested in the same way as the other beads. At each Monte Carlo step the distance $\mathbf{r}_{1\,1} - \mathbf{r}'_{1\,1}$ is added to a histogram. The final histogram of occurrences of $\mathbf{r}_{1\,1} - \mathbf{r}'_{1\,1}$ is proportional to $n(\mathbf{r}_{1\,1} - \mathbf{r}'_{1\,1})$.

That straightforward approach presents, notwithstanding, several drawbacks. Since one ring remains open throughout the whole simulation, diagonal properties cannot be calculated during the same run. Furthermore, it is not very efficient; only one distance out of the $3NM$ path variables contributes to the single particle density matrix. Moreover, the end-to-end distance spends more time at large distances than at small ones, which means that the value of $n(\mathbf{r})$ at small $\mathbf{r}$ is undersampled. However, this sampling problem can be solved using importance sampling (Ceperley and Pollock, 1992).

The histogram obtained is proportional to the single–particle density matrix, with a normalisation externally imposed.

There is one more *caveat* about this approach: it does not only implies a specific simulation but also a specific code, since one path variable, the open chain, receives a different treatment respect to the others.

McMillan method

Another approach is similar to the McMillan method used at zero temperature. A simulation with all closed polymers is occasionally stopped and each bead displaced, making a path momentarily open. The single–particle density matrix $n(r)$ is the average ratio of the displaced to the undisplaced density matrix. In this way, an estimator for the single–particle density matrix is

$$n(\mathbf{r}_{1\,1}, \mathbf{r}'_{1\,1}) = \left\langle \exp\left[\frac{(\mathbf{r}_{1\,1} - \mathbf{r}_{1\,2})^2}{4\lambda\tau} - U(R_1 \leftarrow \mathbf{r}'_{1\,1}, R_2) + U(R_1, R_2)\right]\right\rangle , \tag{5.48}$$

where U is the potential part of the action and R_1 represents all the coordinates of the beads belonging to atom 1.

A specific simulation is no longer necessary, and the normalisation is obtained by the same simulation, however, this method is only suitable for nearly classical systems and for r smaller than the thermal wavelength.

Trail method

The *trail method* (Brualla et al., 2002) surmounts the problems of previous approaches. It is inspired in the McMillan method and, therefore, it gives the right normalisation. It does not require neither a specific simulation, nor a specific code. Contrariwise to McMillan's recipe, our method efficiently samples the whole range of r without any importance sampling.

We are not to procrastinate anymore the introduction of the new algorithm. When a path is momentarily open, all but one bead, that remains fixed at one end of the polymer, are moved. The movement proposed is a proportional displacement of the form

$$\mathbf{r}'_\alpha \leftarrow \mathbf{r}_\alpha + \frac{\alpha\boldsymbol{\gamma}}{M}, \tag{5.49}$$

where $\boldsymbol{\gamma}$ is a three–dimensional vector given by

$$\boldsymbol{\gamma} = \mu\sqrt{\frac{2\lambda}{T}}\left(\boldsymbol{\delta} - \frac{1}{2}\right), \tag{5.50}$$

and $\boldsymbol{\delta}$ is a 3–D uniformly distributed random vector. The factor $(2\lambda/T)^{1/2}$ is the mean thermal wavelength. We have found that with $\mu = 3$, the one–body density matrix is efficiently sampled.

Notice that the displacement of a given bead is proportional to its relative position to the fixed end of the chain.

An asset of the *trail method* is its simplicity yet its mightiness. It yields right results with smaller variance than the other methods, and also with lower CPU time. In addition it is very easy to code. In order to clarify the algorithm and to make it available to anyone desiring to use it, we have included a commented FORTRAN 77 subroutine in Appendix C.

5.2.2 Momentum distribution

The momentum distribution is defined as the Fourier transform of the single–particle density matrix,

$$n(\mathbf{k}) = \frac{1}{(2\pi)^3\rho}\int d\mathbf{r}\,\exp(-i\mathbf{k}\cdot\mathbf{r})n(\mathbf{r}). \tag{5.51}$$

The momentum distribution is normalised as $\int d^3\mathbf{k}n(\mathbf{k}) = 1$. The kinetic energy K is given by the first moment of the momentum distribution

$$K = \frac{\lambda}{2}\int d^3\mathbf{k}\,k^2\,n(\mathbf{k}) = -\frac{\lambda}{\nabla}^2 n(\mathbf{r})\Big|_{r=0}. \tag{5.52}$$

In the classical limit the distance between neighbouring beads is much less than the inter–atom spacing. To a first approximation, quantum effects affect only the width of the classical single–particle density matrix (and equivalently the momentum distribution) keeping its shape, but changing its width. That is, the momentum distribution remains Maxwellian, but with an effective temperature that gives approximate account of the departure of the kinetic energy from its classical value $3T/2$.

The classical momentum distribution is given by

$$n_{\mathrm{Cl}}(k) = \frac{1}{(2\pi)^3\rho}C\exp\left(-\frac{\lambda}{T}k^2\right), \tag{5.53}$$

with T the temperature. Since the momentum distribution must be normalised to one, the normalisation constant C takes the value

$$C = \frac{8\rho(\pi\lambda)^{3/2}}{T^{3/2}}. \tag{5.54}$$

We know that the zero moment of the momentum distribution must be 1

$$4\pi A \frac{1}{(2\pi)^3\rho} \int_0^\infty k^2 \exp\left(-\frac{\lambda}{T}k^2\right) dk = 1, \tag{5.55}$$

with A the normalisation constant and $\lambda = \hbar^2/2m$. Integration gives

$$A = \frac{8\rho(\pi\lambda)^{3/2}}{T^{3/2}}. \tag{5.56}$$

The second moment of the momentum distribution is the kinetic energy

$$K = A\frac{\lambda}{2\pi^2\rho} \int_0^\infty k^4 \exp\left(-\frac{\lambda}{T}k^2\right) dk. \tag{5.57}$$

The classical momentum distribution is given by

$$n_G(k) = A \exp\left(-\frac{\lambda}{T}k^2\right) \tag{5.58}$$

with T the temperature. A first approximation to the quantum result is to substitute the T by an effective temperature.

The Fourier anti–transformation of the classical momentum distribution yields the single particle density matrix.

$$n_G(r) = 4\sqrt{2}\rho\pi^{3/2}\frac{\lambda}{T} \exp\left(-\frac{T}{4\lambda}r^2\right). \tag{5.59}$$

5.2.3 Compton profile

Within the framework of information exchange with experimentalists, the Compton profile becomes relevant, being the observable measured in neutron scattering experiments.

At sufficiently high momentum transfer, the impulse approximation holds because the neutron scatters from a single atom. Assuming that, the scattering function is given by

$$J_{\mathrm{IA}}(Y) = \frac{q}{m} S_{\mathrm{IA}}(q,\omega), \tag{5.60}$$

with

$$Y = \frac{m}{q}\left(\omega - \frac{q^2}{2m}\right). \tag{5.61}$$

Provided that, we can define the Compton profile in terms of the momentum distribution as (Mazzanti et al., 1996),

$$J_{\mathrm{IA}}(Y) = \frac{1}{4\pi^2\rho}\int_{|Y|}^{\infty} kn(k)dk. \tag{5.62}$$

In a PIMC simulation, the calculation of the Compton profile is done once the code has finished its run. From the simulation we obtain the single–particle density matrix, from that the momentum distribution, and then integrating in order to get the Compton profile.

5.3 Radial distribution function

The radial distribution function is proportional to the probability of finding two particles separated a distance r. A formal definition is,

$$g(r) = \frac{V}{N^2}\langle\sum_i\sum_{i\neq j}\delta(\mathbf{r}-\mathbf{r}_{ij})\rangle, \tag{5.63}$$

where N is the number of atoms and the indexes i, j run over atoms.

In the practice the *delta* function is translated into a binning and a histogram is generated with the distances falling within the binning. Notice there is no bead label. What we do is to choose always the same bead, for example, bead 1 of all the polymer chains, and then we calculate $g(r)$ as if we were dealing with a classical liquid where every atom is represented by just one point–particle.

More statistics can be obtained using all the beads instead of just one bead per particle. The error bars of the $g(r)$ are very small compared to those of the energy for the same number of iterations.

5.4 Structure factor

The structure factor is another important distribution function. It will serve us for deciding whether we are simulating a liquid or a solid. This is of key importance in solid–liquid phase transitions, as we shall notice in a forthcoming chapter §9.

It can be defined in terms of the pair distribution function as

$$S(k) = 1 + \rho \mathcal{F} h(r), \tag{5.64}$$

with $\mathcal{F}h(k)$ the Fourier transform of the total correlation function $h(r) = g(r) - 1$. In an homogeneous system the angular part of (5.64) can be carried out exactly, and then,

$$S(k) = 1 + 4\pi\rho \int_0^\infty r^2 \frac{\sin kr}{kr} (g(r) - 1) dr. \tag{5.65}$$

From equation (5.65) is possible to calculate the structure factor out of the $g(r)$. Notwithstanding, this approach poses a problem: the pair distribution function is accurate for a short range of r due to finite size effects of the simulation box. Thus, its Fourier transform will be accurate only for large k. For short k the results obtained using (5.65) are unaccurated since for distances greater than $L/2$ we have approximated $g(r)$ to one.

A second approach, useful mainly for small values of k, is to estimate the structure factor within the simulation using a subroutine coded for that purpose. In order to do so, we use the microscopic definition of the structure factor

$$S(k) = \left\langle \sum_i e^{i\mathbf{k}\cdot\mathbf{r}_i} \sum_j e^{-i\mathbf{k}\cdot\mathbf{r}_j} \right\rangle, \tag{5.66}$$

with discretised $\mathbf{k}$ values according to the usual rule

$$\mathbf{k} = \frac{2\pi}{L}(n_x, n_y, n_z). \tag{5.67}$$

6. THE RICHARDSON EXTRAPOLATION

How many beads are required is perhaps the first question one wonders just before starting a PIMC simulation. There are theoretical considerations about the number of beads, stating that both of the following requirements have to be fulfilled (Gillan, 1990). First, the root mean squared length of the springs must be smaller than the relevant length scale of the external potential. This translates into the following relation:

$$M \gg \frac{\beta\hbar^2}{m\sigma^2}, \tag{6.1}$$

with M the number of beads, and σ the length scale of the potential. Second, τ must be much smaller than the typical vibration period of the molecule.

Though valid, these criteria just say that the number of beads must be larger than some quantities.

A more practical answer to the initial question is that M must be large enough so that increasing it further makes no significant changes to the calculated averages. The meaning of ‘significant’ depends on the degree of accuracy we require. Proceeding that way implies a large number of simulations, and once the plateau has been reached, where the changes of averages fall within error bars, one has to repeat some more simulations with even larger values of M.

This approach poses one substantial problem. If the temperature we are simulating is low enough that we are close to the limit of our computing capabilities, then the extra simulations required to ensure we have arrived to the plateau are not feasible. In a situation like this the Richardson extrapolation comes in very handy.

Should the simulated temperature be high enough, allowing us to perform the extra simulations required in order to make certain the results, the Richardson extrapolation saves us from doing so, and therefore the overall time spent in a given problem reduces significantly.

The simplest way to use the Richardson extrapolation consists in doing a simulation with a given number of beads M and another with $2M$. Provided with the energy figures for those two choices of beads we find the extrapolated result for $4M$. Then we perform the simulation with $4M$ beads and we compare the simulated value of the energy with the extrapolated one. Should they coincide within error bars, we terminate the simulation process and we assume that the right number of beads is $4M$. If the simulated and extrapolated values of the energy are not compatible the algorithm keeps iterating until agreement is reached.

We have also developed a more elaborated way of applying the Richardson extrapolation that clearer the moment when convergence has been reached. A further advantage of the Richardson extrapolation is the self–consistency data check it provides.

Richardson extrapolation can be used along with the trapezoidal integration rule. That combination is known as Romberg integration (Guardiola et al., 1995). It has also been used for solving ordinary differential equations, yielding the Bulirsch–Stoer method (Press et al., 1999). However, as far as we know, there are no applications of the Richardson extrapolation to quantum Monte Carlo problems in literature.

In this chapter we shall present a simple derivation of the Richardson extrapolation and we also shall give a program for calculating it using the more refined process cited above. We shall also briefly discuss the possibility of implementing the Richardson extrapolation inside the PIMC code itself, and the problems entailed with it. Its usefulness will be seen in the chapter devoted method testing.

The strength of the Richardson extrapolation has been overlooked many times. It is a powerful method, as shown in the words of Press et al. (1999) "Richardson extrapolation is a method for turning straw into gold! (Lead into gold for alchemist readers.)" Notwithstanding that sentence, it must be said this is absolutely true only if we are dealing with error–free numbers. In the presence of the typical error bars associated with PIMC results, the strength of the Richardson extrapolation can be diminished.

6.1 Theoretical considerations

There is a general derivation of the Richardson extrapolation to a fourth order accuracy available in the book of Guardiola et al. (1995). That reference shows that the Richardson extrapolation obtained using two values of the energy, given by two simulations (one using M beads and the other one using $2M$ beads) is:

$$E_\infty = \frac{4E_{2M} - E_M}{3}, \tag{6.2}$$

where E_∞ is the asymptotic value of the energy.

Iterating the procedure described in (Guardiola et al., 1995) we found the Richardson extrapolation using three points

$$E_\infty = \frac{64E_{4M} - 20E_{2M} + E_M}{45}. \tag{6.3}$$

In the previous formulæ it has been used the fact that the next simulation always doubles the number of beads. This fact simplifies the algebra involved and the final expression obtained. However, it is not always feasible to double in number of beads, due to the CPU time required, yet it is possible to keep increasing the number of beads. We shall present an analogous derivation to the Richardson extrapolation formula, more adapted to our circumstances.

We shall find the extrapolation to infinity for a fourth order correction knowing two values of the energy for any number of beads. The derivation for a second order is completely analogous. We define E_1 and E_2 as the energies found using $M1$ and M_2 beads respectively.

It is possible to define E_1 and E_2 as

$$E_1 = E_0 + \frac{\alpha}{M_1^4}, \tag{6.4}$$

and

$$E_2 = E_0 + \frac{\alpha}{M_2^4}, \tag{6.5}$$

where α is any real number. It is possible to find α in terms of the energies and the number of beads as:

$$E_2 - E_1 = \alpha \left(\frac{1}{M_2^4} - \frac{1}{M_2^4} \right). \tag{6.6}$$

$$\alpha = \frac{E_2 - E_1}{\frac{1}{M_2^4} - \frac{1}{M_1^4}}. \tag{6.7}$$

With $M_2 > M_1$ and taking into account (6.6) and (6.7)

$$\begin{aligned} E_0 &= E_2 - \alpha \frac{1}{M_2^4} \\ &= E_2 - \frac{E_2 - E_1}{\frac{1}{M_2^4} - \frac{1}{M_1^4}} \frac{1}{M_2^4} \\ &= E_2 - \frac{(E_2 - E_1)}{\left(\frac{1}{M_2^4} - \frac{1}{M_2^4}\right)} \frac{M_1^4}{M_1^4 M_2^4} \\ &= E_2 + \frac{(E_2 - E_1)}{1 - \left(\frac{M_1}{M_2}\right)^4} \left(\frac{M_1}{M_2}\right)^4 . \end{aligned} \tag{6.8}$$

Therefore, the extrapolation to infinity is

$$E_\infty = E_2 + \left\{ \frac{(M_1/M_2)^4}{1 - (M_1/M_2)^4} \right\} (E_2 - E_1). \tag{6.9}$$

It is also desirable to know the extrapolated value not to infinity but to a given number of beads, in order to perform a simulation for that number of beads and then to compare. The Richardson extrapolation to a number of beads M of two given energies E_1 and E_2 calculated using M_1 and M_2 beads, and assuming a second order behaviour is given by

$$E_M = E_{M_1} + (E_{M_2} - E_{M_1}) \left[\frac{1 - (M_1/M)^2}{1 - (M_1/M_2)^2} \right]. \tag{6.10}$$

Depending on the action we shall be using, primitive or Li–Broughton, we shall apply either the second order or the fourth order extrapolation.

6.2 Caveat

We have already mentioned that extrapolated numbers should have small error bars, otherwise we would be extrapolating errors. This represents a capital problem, since accurate numbers are difficult to obtain with PIMC simulations. Simulations of ^{4}He at 5 K could require a month, or so, of a Pentium 800 MHz processor, in order to obtain two significant decimal figures.

So far we have given the formulæ for the Richardson extrapolation to infinity, without further discussion about the meaning of ‘infinity’. The energy of a system depends on the number of beads in a quadratic form once we are getting close to the number of beads required for the simulated temperature. Therefore, the Richardson extrapolation will yield accurate results to infinity once the quadratic behaviour has been set up. Extrapolations performed in regions with lower number of beads will approach to the ‘exact’ result, but they will not be the right answer.

The overall dependence of the energy on the number of beads, in the full range from 1 to M beads, is more similar to a straight line than to a parabola, for the primitive approximation. Therefore, using results obtained with a number of beads much lower than the required ones would induce us to errors in the extrapolation. Those numbers convey little information about the quantum properties of the system. Since the common practice is to double the number of beads in every new simulation, the values of the energy are more separated for low M in a plot of E as a function of $1/M$, thus, the shape and the slope of the extrapolation curve is mainly determined by them.

6.3 Recursive extrapolation

Using equation (6.10) we have coded an algorithm that extrapolates the values found in previous extrapolations in an iterative way. The advantage of doing so, is that one can easily realise which is the sufficient number of beads. The results of the code are presented in a matrix. The first column contains the values of the energy (or any other averaged quantity). The second column has the extrapolations obtained with (6.10). The third column has the extrapolations of the extrapolated values of the second column, and so on. Convergence is reached once we find a row with all the numbers compatible.

The code, written in Mathematica, also gives a matrix with the associated error to each position of the extrapolation matrix. The errors matrix is obtained with error propagation theory applied to (6.10).

```
(* Richardson Matrix with errors                *)
(* lim = number of points, i.e. simulations      *)
(* x = potential behaviour of the extrapolation. *)
(* x = 1/4. = quadratic = primitive action       *)
(* x = 1/16.= quartic   = li broughton action    *)

(* Total energy Helium 5K with errors.           *)
(* Primitive action. Virial est.                 *)
Clear[lim,x,y,adjust,q,qerr];
lim = 8;
x = N[1/4.0,5];
q = Table[0,{i,1,lim},{j,1,lim}];
qerr = Table[0,{i,1,lim},{j,1,lim}];
q[[1,1]]=-35.48;
q[[2,1]]=-21.94;
q[[3,1]]=-12.92;
q[[4,1]]=-8.250;
q[[5,1]]=-5.16;
q[[6,1]]=-3.33;
q[[7,1]]=-2.47;
q[[8,1]]=-2.2;
qerr[[1,1]]=0.04;
qerr[[2,1]]=0.04;
qerr[[3,1]]=0.02;
qerr[[4,1]]=0.007;
qerr[[5,1]]=0.01;
qerr[[6,1]]=0.02;
qerr[[7,1]]=0.03;
qerr[[8,1]]=0.1;
Do[
Do[
q[[n,m]]=N[(q[[n,m-1]]-x^(m-1)
         q[[n-1,m-1]])/(1-x^(m-1)),5],
{m,2,n}
],
{n,2,lim}
];
```

```
Do[
Do[
qerr[[n,m]]=
N[Sqrt[qerr[[n,m-1]]^2 + x^(2m-2)*
       qerr[[n-1,m-1]]^2]/(1-x^(m-1)),5],
{m,2,n}
],
{n,2,lim}
];
Print[MatrixForm[q]]
Print[MatrixForm[qerr]]
```

In order to give the complete example, we reproduce the output of the code used upon a set of simulations of ^{4}He at 5 K. Matrix element $(1,1)$ has the value of the energy for 1 bead. We double the number of beads until reaching 128 beads. The matrices shown are the extrapolation matrix and the error matrix respectively.

```
-35.48  0       0       0       0       0       0       0
-21.94-17.427   0       0       0       0       0       0
-12.92 -9.9133 -9.4124  0       0       0       0       0
 -8.25 -6.6933 -6.4787 -6.4321  0       0       0       0
 -5.16 -4.13   -3.9591 -3.9191 -3.9093  0       0       0
 -3.33 -2.72   -2.626  -2.6048 -2.5997 -2.5984  0       0
 -2.47 -2.1833 -2.1476 -2.14   -2.1381 -2.1377 -2.1376  0
 -2.2  -2.11   -2.1051 -2.1044 -2.1043 -2.1043 -2.1043 -2.1043
```

```
  0.04  0       0       0       0       0       0       0
  0.04  0.05498 0       0       0       0       0       0
  0.02  0.02981 0.03201 0       0       0       0       0
  0.007 0.01147 0.01240 0.01260 0       0       0       0
  0.01  0.01354 0.01446 0.01469 0.01474 0       0       0
  0.02  0.02687 0.02868 0.02914 0.02925 0.02928 0       0
  0.03  0.04055 0.04329 0.04398 0.04415 0.04420 0.04421 0
  0.1   0.13371 0.14265 0.14491 0.14548 0.14562 0.14566 0.14567
```

We notice that 128 beads are required for reaching convergence. The value of the total energy with 128 beads is -2.2(1) K. However, if the

Richardson extrapolation is used the same result is obtained with a fraction of the computational effort. The extrapolated energy obtained using simulations up to 64 beads is -2.14(5)K, which is compatible with the value found with a 128 beads running. The simulation performed with 128 beads took more time than all the other simulations together, and the variance is an order of magnitude larger.

6.4 Internal Richardson extrapolation

Another approach we have tried to the Richardson extrapolation, was to implement an extrapolation inside the code. That means, while the code is running we perform extrapolations using reweighted values of the energy. For example, when we run a simulation with eight beads we estimate the energy for that polymer length at every Monte Carlo step. The internal Richardson extrapolation routine we have coded estimates the value of the energy for the four beads polymer chain taking one every two beads of the eight-beads chain and reweighting. And then does the same again in order to estimate the value of the energy for a chain made up of two beads. In that way when we are performing a simulation with M beads, we can have two rough estimates of the energy of the same system using $M/2$ and $M/4$ beads.

We coded that routine with the hope of reducing the total CPU time invested in a system, since the reweighted values require much less computational effort than real simulations. However, the reweighted results are significantly noisier than their PIMC–estimated counterparts. This level of noise introduced into the Richardson extrapolation becomes it completely useless for giving an accurate result when the algorithm is run in the asymptotic zone. Results on the internal Richardson extrapolation will be shown in § 7.1.3.

7. TESTING THE METHODS

In this chapter we shall give the results obtained with the different methods showed so far. We shall see comparisons between primitive and Li–Broughton actions, as well as results derived from some of the different samplings explained so far. We shall also present comparative results related to different momentum distribution methods.

Following the same stream developed in the chapters 3 and 5, we shall give the results for the harmonic oscillator first, and then results on real systems. Most of the tests performed on real systems have been done on ^{4}He at 5 K and Ne at 25 K and 35 K.

For either, the harmonic oscillator, as well as the real systems, we present results of the Richardson extrapolation.

7.1 Harmonic oscillator

7.1.1 Actions

In chapter 3 we derived several actions. We found the expressions applied to the harmonic oscillator for the primitive approximation (3.14), the Li–Broughton correction (3.31), the variational theorem applied to the primitive approximation (3.37), and finally the variational improvement applied on a Li–Broughton action (3.39).

We shall first test how these various effective potentials behave in a PIMC code for estimating the energy of the quantum harmonic oscillator. This model problem will help us to choose which of those potentials are the best candidates for a real problem.

All the simulations of the harmonic oscillator have been performed at a temperature of $T = 0.2$ K, which is sufficiently low for appearing quantum effects. From (2.49) we calculate the energy using $\omega = \hbar = k_{\mathrm{B}} = \lambda = 1$. The exact value of the energy at that temperature is

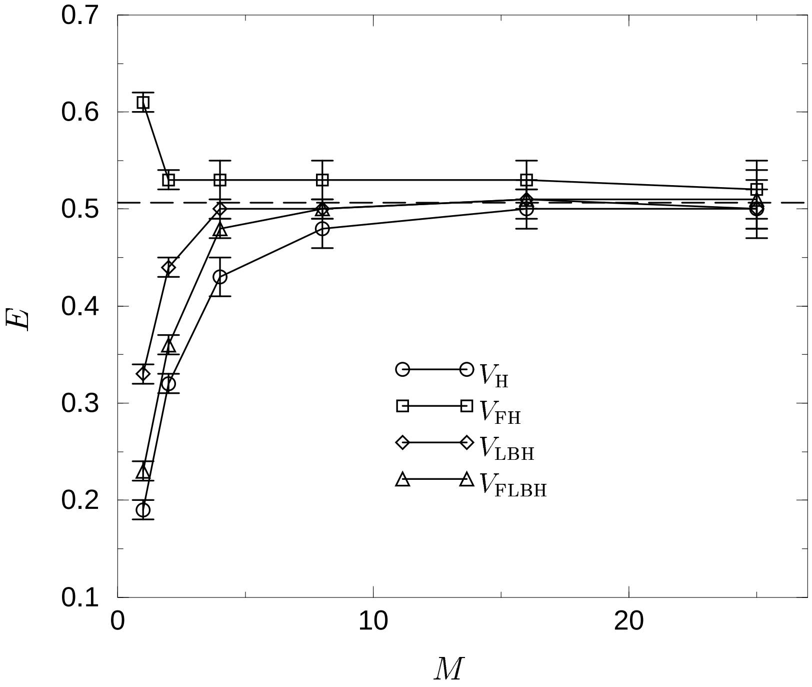


Fig. 7.1: Estimation of the harmonic oscillator energy at $T = 0.2$ K, using different actions, as a function of the number of beads. Exact value of the energy (0.50678 K) dashed line.

0.50678 K.

In order to test the convergence of the different actions studied we measure the total average energy of the quantum harmonic oscillator. We have only used the thermodynamic energy estimator for these tests. The results shown in table 7.1.1 have been obtained running all the simulations an equal amount of CPU time. This choice about the CPU time has been done in order to observe the behaviour, for a given action, of the error bars as a function of the number of beads, as well as to compare the variance, for a given number of beads, amongst different actions. Figure 7.1 shows a plot of the results given in table 7.1.1.

We observe that the variance of all the actions tested behaves similarly with the number of beads. Therefore, the variance is not an issue when choosing an action.

M	V_{H}	V_{FH}	V_{LBH}	V_{FLBH}
1	0.19(1)	0.61(1)	0.33(1)	0.23(1)
2	0.32(1)	0.53(1)	0.44(1)	0.36(1)
4	0.43(2)	0.53(2)	0.50(1)	0.48(1)
8	0.48(2)	0.53(2)	0.50(1)	0.50(1)
16	0.50(2)	0.53(2)	0.51(1)	0.51(2)
25	0.50(3)	0.52(3)	0.50(2)	0.51(3)

Tab. 7.1: Comparison of different actions for the harmonic oscillator. Average value of the energy (thermodynamic estimator) at $T = 0.2$ K as a function of the number of beads M.

We stated, when explaining the Feynman variational theorem, that it gives an upper bound for the value of the energy. We see that fact reflected in figure 7.1. Feynman's estimator converges quicker to the vicinity of the exact value than the other estimators, however this rapid convergence rate drastically slows down when the number of beads is increased. The variational theorem guarantees the most accurate value for a single bead evaluation, that is, not taking into account the Trotter formula. Raedt and Raedt (1983) proved that the primitive approximation is accurate to the β^2 order. The variational theorem adds a β–dependent term, therefore, for a number of integrals greater than one *i.e.* $M > 1$, Feynman's correction slows down the convergence of the Trotter formula.

In the light of these facts we decided not to use any action involving the variational theorem. Now, we focus our attention on the primitive approximation to the action and the Li–Broughton correction. Table 7.2 shows the average values of the energy of the harmonic oscillator as a function of the number of beads, for both, the primitive approximation and the Li–Broughton actions. We let every simulation run until five significant figures for the energy were obtained. The outcomes obtained are in agreement with those from Li and Broughton (1987).

At a first glance we notice that for obtaining the analytical value of the energy with the primitive approximation, 512 beads are required. Li–Broughton's action achieves the same value with just 32 beads. Figure 7.2 is a plot of the data presented in table 7.2 along with two fits. The

M	E_{PR}	E_{LB}
1	0.18519	0.32666
2	0.30755	0.44702
4	0.43162	0.50053
8	0.48424	0.50630
16	0.50085	0.50675
32	0.50528	0.50678
64	0.50641	0.50678
128	0.50669	0.50678
256	0.50676	0.50678
512	0.50678	0.50678

Tab. 7.2: Energy of the harmonic oscillator as a function of the number of beads for the primitive and Li–Broughton actions

inset in the upper right corner is a zoom of the region near the plateau.

Once the number of beads employed is sufficient to capture the quantum effects of the problem, the primitive approximation behaves quadratically with the inverse of the number beads, as we expected. The same happens with the Li–Broughton correction to the action: it goes as a fourth power of $1/M$, as we also expected. The data obtained with the primitive approximation fits to the parabola $E = 0.5061(6) - 1.20(3)(1/M)^2$ excluding the values yielded with one and two beads. The values of the energy got using a Li–Broughton action fit to the curve $E = 0.5064(4) - 0.95(2)(1/M)^4$ once the result for one bead has been shredded. In both cases the difference between the independent term and the analytical value is less than 0.2%.

7.1.2 Richardson extrapolation in the harmonic oscillator

In § 6.1 we introduced the second–order two–point Richardson extrapolation formula (6.2) assuming that the number of beads M_2 used for the second simulation doubles the number of beads of the first one M_1. Based on an equivalent assumption , that is $M_3 = 2M_2 = 4M_1$, we also introduced (6.3) which is the second–order three–point Richardson extrapolation formula. Later on, we derived the general fourth–order

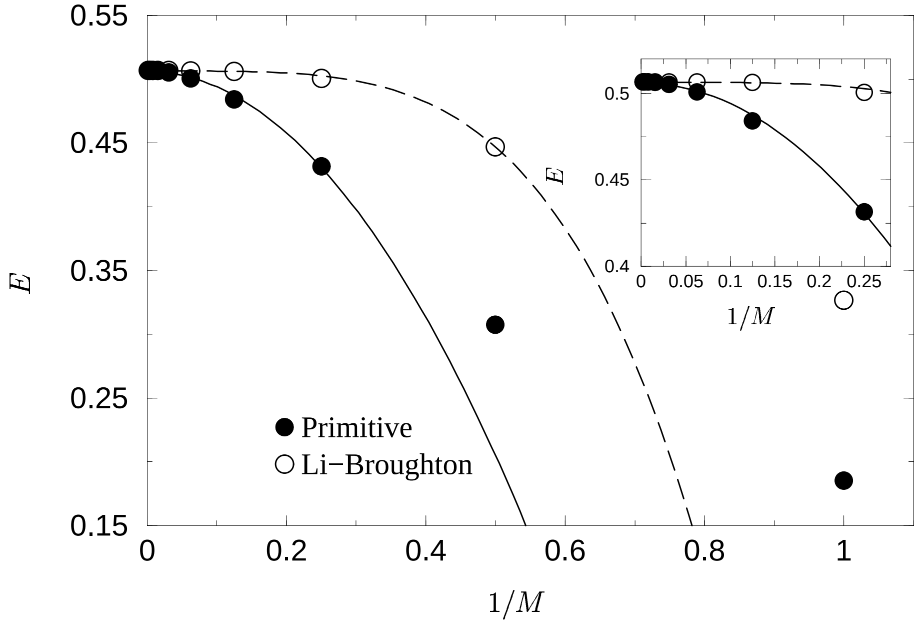


Fig. 7.2: Comparison between primitive and Li–Broughton actions in the harmonic oscillator at $T = 0.2$ K

M	E_{PR}	$R_2(\mathcal{O}^2)$	$R_3(\mathcal{O}^2)$	E_{LB}	$R_2(\mathcal{O}^4)$
1	0.18519			0.32666	
2	0.30755	0.34834		0.44702	0.45504
4	0.43162	0.47298	0.48129	0.50053	0.50410
8	0.48424	0.50178	0.50370	0.50630	0.50668
16	0.50085	0.50639	0.50670	0.50675	0.50678
32	0.50528	0.50676	0.50678	0.50678	0.50678
64	0.50641	0.50678	0.50678	0.50678	0.50678
128	0.50669	0.50678	0.50678	0.50678	0.50678
256	0.50676	0.50678	0.50678	0.50678	0.50678
512	0.50678	0.50678	0.50678	0.50678	0.50678

Tab. 7.3: Richardson extrapolation applied to the harmonic oscillator using primitive and Li–Broughton actions

two–point Richardson extrapolation formula (6.9). For this case we can also use the condition $2M_2 = 4M_1$ and obtain

$$E_\infty = \frac{16E_{2M} - E_M}{15}. \tag{7.1}$$

We shall refer to (6.2), (6.3), and (7.1) as $R_2(\mathcal{O}^2)$, $R_3(\mathcal{O}^2)$, and $R_2(\mathcal{O}^4)$ respectively.

Using the formulæ cited above on table 7.2 we have calculated table 7.3. Just performing three simulations with 4, 8 and 16 beads, along with a Li–Broughton action, and lately applying (7.1) we match the analytical result. Considering that a primitive action by itself requires 512 beads to yield the same precision, these upshots are astonishing. However we must remember that most of the strength of the Richardson extrapolation is rooted in the fact that the numbers involved have a high degree of accuracy. We shall see when dealing with real problems that five significant figures of accuracy are not possible to obtain using primitive or Li–Broughton actions. In such cases the strength of the Richardson extrapolation might become seriously handicapped.

M	E_{PR}	$R_3^i(\mathcal{O}^2)$
8	0.484(1)	0.50(4)
12	0.496(1)	0.50(4)
24	0.504(1)	0.51(6)
32	0.505(1)	0.48(6)
48	0.506(1)	0.51(6)
96	0.507(1)	0.52(5)
128	0.507(1)	0.46(6)

Tab. 7.4: Internal Richardson extrapolation

7.1.3 Internal Richardson extrapolation

Based on the code explained in § 6.4 we have calculated table 7.4, which shows the values of the energy as a function of M for the harmonic oscillator at $T = 0.2$ K. E_{PR} stands the primitive approximation outcomes, while $R_3^i(\mathcal{O}^2)$ holds for the internal three–point Richardson extrapolation.

In 7.3 we plot the data of table 7.4. There are no error bars for the energies estimated with the primitive action since the errors fall within the symbol size.

Since we were aware that the key point of the Richardson extrapolation was to employ numbers with a low level of noise, we wanted to observe the influence of the reweighted results on the extrapolation. In order to do so we let each simulation on table 7.4 run until three significant figures in the energy were obtained. Notice that the values of $R_3^i(\mathcal{O}^2)$ begin to fluctuate at 32 beads, just when the primitive action has reached the expected exact result for the required precision.

The denouement indicates us not to use an internal Richardson extrapolation. However, the Richardson extrapolation by itself proves to be very useful as shown in table 7.3.

7.2 Real systems

Real strengths and weaknesses have to be observed in real systems, which all in all, are our ultimate goal.

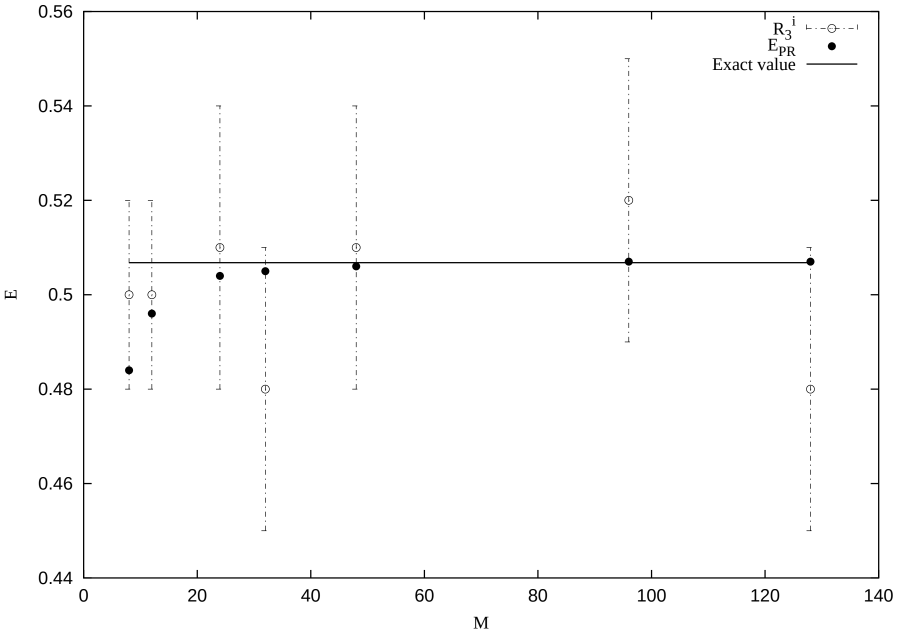


Fig. 7.3: Internal Richardson extrapolation of the energy estimated for the harmonic oscillator at $T = 0.2$ K

We shall begin studying near–classical systems, where quantum effects do not have a large influence. As we gain confidence on the codes, we shall lower the temperature and move onto systems which show a clear quantum behaviour. Nevertheless, we work in all cases in conditions where Bosonisation is not necessary.

We have chosen to group the information of this section on the theoretical bodies we have been using so far, that is: sampling, action, Richardson extrapolation, radial distribution function, and momentum distribution. We did that, rather than grouping the information in terms of physical systems (helium, neon), since, at the moment, we are more interested in the behaviour of the codes rather than discovering new physics. As long as it has been possible, we have simulated systems with well known physical characteristics, from both, the experimental, as well as the simulationist's point of view.

We have studied ^{4}He and Ne at several temperatures and densities. In order to obtain results that can be compared with the available literature, we have used the Lennard–Jones potentials of Ne and ^{4}He, and also the Aziz and Slaman (1989), and the Aziz et al. (1997) potentials.

In the following bulleted lists we summarise the physical conditions and key features of every system studied. We shall give a code name to every system, in that way citing that reference will avoid to keep mentioning every single aspect of the simulation. If any simulation condition deviates from the lists below, it will be indicated. The lists also indicate the value of the tail correction. These corrections are already included in all the tables of this chapter.

Liquid neon at 35 K

- *Code name*: **Ne35**
- *Temperature*: 35.209 K
- *Density*: 3.346×10^{-2}Å^{-3}
- *Number of atoms*: 64
- *Potential*: Lennard–Jones: $\sigma = 2.789$ Å, $\epsilon = 36.814$ K
- *Tail correction*: -20.313 K

- *Reported experimental results*: The kinetic energy at $T = 35.3\pm0.4$ K and at the same density of the simulation, measured via neutron scattering, was reported to be 66.4 ± 3.3 K (Peek et al., 1992).

Liquid neon at 25 K

- *Code name*: **Ne25**

- *Temperature*: 25.8 K

- *Density*: $3.63{\times}10^{-2}$Å^{-3}

- *Number of atoms*: 108

- *Potential*: Aziz HFD-B Ne–Ne (Aziz and Slaman, 1989)

- *Tail correction*: -9.610 K

- *Reported experimental results*: We found two reported values of the kinetic energy for the temperature and density specified, one from Glyde et al. (1995) yielding 52.9 ± 2.5 K, and the other by D. A. Peek et al. (1992), who gives a compatible value of 52.8 ± 3.7 K. However, there is an earlier result by Sears that yields 48.8 K for the same system.

Liquid helium at 5 K

- *Code name*: **He5LJ** for the Lennard–Jones potential and **He5A** for the Aziz potential

- *Temperature*: 5 K

- *Density*: $2.1858{\times}10^{-2}$Å^{-3}

- *Number of atoms*: 64

- *Potential*: Lennard–Jones: $\sigma = 2.556$ Å, $\epsilon = 10.21651$ K

- *Potential*: Aziz HFD-B(HE) (Aziz II) (Aziz et al., 1997)

- *Tail correction Lennard–Jones potential*: -1.4253 K

M	V	K_{T}	E_{T}
1	-187.34(5)	52.8135	-134.53(5)
2	-184.91(6)	57.43(6)	-127.48(6)
4	-183.77(6)	60.2(2)	-123.6(2)
8	-183.49(5)	60.5(5)	-123.0(5)

Tab. 7.5: Ne35. Primitive action, bead per bead sampling, thermodynamic energy estimator

- *Tail correction Aziz potential*: -1.3094 K
- *Tail correction Aziz potential (108 atoms)*: -0.7683 K

7.2.1 Sampling

We shall focus our attention on comparing the bead per bead sampling against bisection. We first studied the system Ne35 using a primitive action, a bead per bead sampling, and a thermodynamic energy estimator. All the simulations have the same number of iterations. The results are shown in table 7.5.

We observe that, within error bars, the asymptotic value has been reached using just 8 beads. As we have already announced, we also observe that the variance grows with the number of beads, in contrast with the variance of the potential energy.

In table 7.6 we study the same system, but instead of using a bead per bead sampling, we used bisection. The minimum number of beads required for a bisection code is eight, enabling us to use a level two of bisection algorithm.

Outcomes from table 7.6 are compatible with those obtained on table 7.5, therefore, the results obtained using either bisection or bead per bead sampling are equivalent, as it has to be. We have chosen Ne at 35 K for this study since with just 8 beads the asymptotic value of the energy has been reached, therefore, any change in the value of the energy, or its variance, when increasing the number of beads further than 8, will reflect the influence of the simulation tools we have chosen, and not any change of the physics of the system.

M	l	It	V	K_{T}	E_{T}
8	2	6	-183.2(2)	60.4(3)	-122.8(5)
16	2	6	-183.8(3)	59.5(9)	-124(1)
16	3	6	-183.6(3)	61.1(7)	-122.5(7)
32	2	10	-183.7(2)	61(4)	-122(1)
32	3	10	-183.2(3)	60.2(5)	-123.2(6)
32	4	10	-183.6(4)	59.6(9)	-124(1)

Tab. 7.6: Ne35. Primitive action, bisection sampling, thermodynamic estimator

In table 7.6 we also present a dependence study of the estimated energy on the bisection level. The column entitled M indicates the number of beads used, and the column headed by l refers to the level of the bisection algorithm used. Column 'It' indicates the number of blocks the code has iterated; all blocks had the same number of Monte Carlo steps for all simulations. For 8 beads the only possible choice, excluding the trivial one, was $l = 2$. The variances of the total energy obtained with an eight bead polymer ring for either the bead per bead or the bisection sampling are the same, however the bisection code lasted 1/10 of the CPU time spent on the bead per bead sampling

When 16 beads are employed we can either choose between 2 or 3 levels. Since the number of iterations for the simulations performed with 8 and 16 beads is the same, it is obvious that simulations with 16 beads have a larger variance than the simulation using 8 beads. However, the differences in the variance of the kinetic energy between the results of $l = 2$ and $l = 3$ at 16 beads, can only account for the choice of the level, since all the other conditions are the same. For this case it is better to use a $l = 3$ sampling than any other choice.

Our results for the 16 beads simulation of Ne35 can be compared with those published by Singer and Smith (1988). They studied the same system using a primitive action and 16 beads also. The results showed in their publication is $V = -183(2)$ K, $K = 60.74$ K, and $E = 123(3)$ K, which are in full agreement with our results for the same number of beads (table 7.2.1).

We continued oversampling the system, increasing the number of

M	l	It	V	K_{T}	E_{T}
8	2	10	-183.9(2)	60.7(3)	-123.2(4)
16	2	10	-183.4(2)	59.4(8)	-124.0(9)
16	3	10	-183.4(2)	61.2(5)	-122.2(6)
32	2	10	-183.6(2)	59(2)	-124(2)
32	3	10	-183.4(2)	60.8(6)	-122.6(7)
32	4	10	-183.4(2)	60.9(8)	-123.0(9)

Tab. 7.7: Ne35. Primitive action, unbiased bisection sampling, thermodynamic estimator

beads to 32. In order to keep the variance within reasonable boundaries we almost doubled the simulation time. In spite of that, the variance increased, as we expected. With 32 beads, 2, 3, and 4 levels can be used. The lowest value of the variance was still obtained, in the 32 beads sample, for $l = 3$. The variance obtained with $l = 2$ was an order of magnitude larger than the one given by a level three bisection.

In § 4.3.5 we discussed about a small bias of the bisection algorithm. We coded a bisection algorithm that avoided the bias, at the price of much longer CPU time, and we did study the the Ne35 system again, producing the results shown in table 7.2.1.

The figures shown in that table are compatible with those ones of table 7.6. All the simulations had ten blocks of data, however the CPU time spent for each block was larger on an unbiased bisection than the regular bisection. We also performed a study of the dependence of the variance on the level chosen, arriving to the same conclusion: whenever is possible, the best choice is a $l = 3$ bisection.

In order to be sure of the equivalence between a bead per bead sampling, and a bisection code, we simulated another system. Outcomes for neon at 25 K are shown on Table 7.8. There are two rows in that table that give account for the results obtained with 4 beads. The first row is the result obtained using a bead per bead sampling, and the second one using bisection. The results for the thermodynamic and the virial estimators of the energy are compatible within error bars. The CPU time devoted to bisection was 1/10 of the time spent on the bead per bead simulation. The improvement on sampling speed of bisection over bead per bead, grows with the length of the segment of polymer ring bisected.

7.2.2 Action

The comparisons carried out between the primitive approximation and the Li–Broughton action applied to the harmonic oscillator, showed a clear improvement of the latter respect to the former. We observed that in order to obtain five significant figures in the energy, the primitive action required 512 beads, instead of just 32, which was the case of the Li–Broughton correction. In the light of these numbers we thought it was possible to lower the temperature, and to get into regimes unreachable for the primitive action. The primitive action requires about 128 beads to simulate ^{4}He at 5 K. Our current sampling methods do not allow us to simulate more than 128 beads and 108 atoms efficiently. In a situation like that, we are bound to a better action should we require even lower temperatures. If the behaviour we observed in the harmonic oscillator would have been unaltered, the results obtained with 128 beads and a primitive action, can be reproduced with just 16 beads along with a Li–Broughton correction. This would allow us to keep decreasing the temperature. However, as we shall notice through the following studies, Li–Broughton's performance becomes seriously handicapped in the presence of a real system.

The comparative study between the primitive approximation and the Li–Broughton action has been done in terms of temperature. We have studied neon at 25 K and then helium at 5 K. Table 7.8 shows the values of the energy for the system Ne25. The column entitled 'Spl' indicates the sampling method, being 'bb' a bead per bead sampling and 'bc' bisection. At the moment we shall focus on the $E_{\rm V}$ column, which states the values of the total energy for the virial estimator. Later on, in § 7.2.3 we shall return to this table and make a comparative analysis between the thermodynamic and virial energy estimators.

The same structure of table is kept in table 7.9. It shows the results of the simulations performed on neon at 25 K, but this time using a Li–Broughton action instead of a primitive approximation. Once again, we shall focus on column $E_{\rm V}$.

We observe from table 7.8 that the asymptotic value of the total

M	Spl	V	K_{T}	K_{V}	E_{T}	E_{V}
1	bb	-213.4(2)	38.7000	38.7000	-174.7(1)	-174.7(1)
2	bb	-210.2(2)	44.3(2)	44.46(3)	-165.8(3)	-165.9(2)
4	bb	-208.6(3)	47.6(2)	47.7(1)	-161.0(4)	-160.9(3)
4	bc	-208.3(2)	47.6(2)	47.76(5)	-160.7(3)	-160.5(2)
8	bc	-207.3(2)	48.8(3)	49.07(7)	-158.5(4)	-158.2(2)
16	bc	-207.6(4)	49.8(5)	49.5(1)	-157.8(6)	-158.1(4)
32	bc	-207.2(4)	49(2)	49.6(1)	-157(2)	-157.5(4)
64	bb	-207.1(3)	50.2(5)	50.0(1)	-156.9(6)	-157.2(3)

Tab. 7.8: Ne25. Primitive action, thermodynamic and virial estimators

M	Spl	V_{T}	K_{T}	E_{T}	E_{V}
1	bc	-222.3(2)	56.0446	-166.2(2)	-166.2(2)
2	bc	-212.9(2)	55.4(2)	-157.4(2)	-157.2(2)
4	bc	-209.3(3)	51.9(2)	-157.3(4)	-157.1(3)
8	bc	-207.8(2)	50.3(3)	-157.4(3)	-157.4(3)
16	bc	-207.7(5)	50(1)	-157(1)	-157.7(6)
64	bc	-207.0(5)	50(4)	-157(4)	-157.3(5)

Tab. 7.9: Ne25. Li–Broughton action, thermodynamic and virial estimators

M	l	V	K_{T}	E_{T}	R_{T}	$R_{\mathrm{T}}^{\mathrm{M}}$
1	1	-42.51(5)	7.6624	-34.84(5)		
2	1	-31.95(8)	9.87(2)	-22.08(7)	-17.8(1)	-18.89(8)
4	1	-24.60(4)	11.61(2)	-12.99(3)	-9.96(5)	-10.72(5)
8	2	-22.36(3)	13.92(5)	-8.44(6)	-6.92(9)	-7.3(1)
16	3	-21.26(3)	16.03(6)	-5.23(7)	-4.1(1)	-4.4(1)
32	3	-20.77(3)	17.4(1)	-3.3(2)	-2.7(3)	-2.8(3)
64	3	-20.56(4)	17.6(3)	-2.9(4)	-2.8(6)	-2.8(6)

Tab. 7.10: He5LJ. Primitive action, bisection sampling, thermodynamic estimator

energy is $-157.2(3)$ K. That figure is obtained using 64 beads. However, using just 2 beads with a Li–Broughton action we retrieve the same value. Further increase of the number of beads keeps yielding compatible figures of the asymptotic energy, confirming that the result obtained with 64 beads and a primitive action is the asymptotic one indeed. Our kinetic energy data is compatible with those reported by D. A. Peek et al. (1992).

Albeit Li–Broughton's results action are very encouraging so far, we must not be fooled by them. Now we shall stress the system up to 5 K.

Firstly, we study ^{4}He at 5 K using a Lennard–Jones potential. Simulations performed with the primitive approximation yielded table 7.10. The asymptotic value of the total energy is $-2.9(4)$ K, and it is given by 64 bead polymer rings. We shall confirm this result based upon the Richardson extrapolation in § 7.2.4. All the simulations related to helium have been performed using 64 atoms, unless otherwise indicated.

Table 7.11 summarises the results obtained for the same system using a Li–Broughton action instead. The improvement given by the Li–Broughton action is not as astonishing as it was for a higher temperature. The number of beads required to retrieve the same asymptotic value is 32, instead of 64.

In order to confirm our data validity, we compare it with that obtained by Singer and Smith (1988) and Ceperley and Pollock (1986) in table 7.12. The results belonging to the row 'LB' have been calculated by Singer and Smith (1988) using a Li–Broughton action. On the other

M	l	V_{T}	K_{T}	E_{T}	R_{T}
4	1	-24.99(2)	16.18(2)	-8.81(1)	
8	2	-23.109(5)	17.96(2)	-5.15(2)	-4.90(3)
16	3	-21.881(5)	18.52(3)	-3.36(3)	-3.22(4)
32	3	-21.12(2)	18.4(1)	-2.7(1)	-2.7(2)

Tab. 7.11: He5LJ. Li–Broughton action, bisection sampling, thermodynamic estimator

M	Mth	V	K_{spr}	K	ΔK	E
30	LB	-21.4(4)	211.3	18.6	1.175	-2.76(4)
40	CP	-20.89(2)		18.9(3)		-2.1(3)
80	CP	-20.69(3)		18.1(6)		-2.6(6)

Tab. 7.12: Singer and Smith (1988) and Ceperley and Pollock (1986) results. Lennard–Jones potential. $T = 5.108255$ K

hand, figures appearing in the rows 'CP' have been estimated by Ceperley and Pollock (1986) using the pair action. The asymptotic results reported by both groups of authors are compatible with ours.

For studying a more realistic system we repeated the simulations on helium at 5 K now using an Aziz potential (Aziz et al., 1997).

For assuring the validity of the results obtained using 64 atoms, we have performed a simulation of the system He5A using 108 atoms instead of 64. The estimated value of the energy for 8 beads and a Li–Broughton correction is $-0.47(8)$, which is compatible with the one given for the same number of beads in table 7.13.

The asymptotic value of the energy for the system He5A is reached with 128 beads using a primitive action, and with 32 beads with Li–Broughton correction, as it is shown in table 7.13. Even though Li–Broughton requires a lower number of beads than the primitive approximation, the way Li–Broughton reaches the asymptotic value is not monotonous. This situation poses several problems. First of all, an oscillating approximation to the asymptotic value forces us to carry out further simulations once the asymptotic value has been reached in order

M	E_{LB}	E_{PP}
2	-10.36(3)	
4	-4.71(3)	
8	-0.49(3)	-9.31(7)
16	-1.41(4)	-5.7(2)
32	-2.4(2)	-4.9(3)
64	-2.7(9)	-3.1(7)
128		-2.5(4)

Tab. 7.13: He5A. Primitive and Li–Broughton actions. Total energy given by the thermodynamic estimator

to ensure it is not a coincidence produced by one oscillation. Another problem derived from the oscillating behaviour is that the Richardson extrapolation becomes useless.

We observed that the Li–Broughton correction showed this oscillatory behaviour when hard core potentials are combined with low temperatures. However the oscillations are around the asymptotic value, and for any given number of beads the deviation from it is smaller than the deviation from the asymptotic value observed with the primitive approximation. Furthermore, the asymptotic energy is obtained using a significantly lower number of beads with the Li–Broughton correction, instead of the primitive approximation. From table 7.13 we notice that the total energy for the He5A system is $-2.4(2)$ K. That figure is achieved using 32 beads with a Li–Broughton action. The primitive approximation requires 128 beads in order to yield an equivalent result.

Even though the Li–Broughton action had a fourth order behaviour for the harmonic oscillator, we did not observe the same behaviour for real systems at temperatures lower than 15 K. Notwithstanding there is an interesting regime of temperatures, above 15 K, where quantum effects are not negligible, and where the Li–Broughton correction provides a significant betterment over the primitive approximation. To be more precise, we found that with just 2 beads and a Li–Broughton action, we matched the results obtained with the primitive approximation and 64 beads, in the case of neon at 25 K (Fig. 7.4). We shall discuss about that in § 7.2.4.

There are authors that give a different expression for the fourth order expansion (Jang et al., 2001). The implementation of this expansion perhaps would surmount some of the problems of the Li–Broughton correction at very low temperatures, however we have not done that since the cited paper has been recently published.

7.2.3 Energy estimators

Table 7.8 shows a comparison between the energy yielded by the thermodynamic estimator and the virial estimator. All the simulations shown in this table have the same number of Monte Carlo steps. The variance of the kinetic energy given by the virial estimator, K_{V}, is about an order of magnitude lower than the error given by the thermodynamic estimator of the kinetic energy K_{T}.

As the number of beads increases the error of the thermodynamic estimator becomes larger, mainly due to the kinetic energy. On the other hand, the error due to the potential energy keeps constant.

Using the virial estimator instead of the thermodynamic represents a significant lowering of the CPU times devoted to energy averaging, once the system has been thermalised, for the reason given on § 5.1.9. Since the variance of the thermodynamic estimator is systematically larger than the one of the virial estimator, even very long runs with the thermodynamic estimator would yield a larger variance. Albeit the mean value of the energy would be closer to the exact value as the number of Monte Carlo steps becomes larger.

7.2.4 Richardson extrapolation

The strength of the Richardson extrapolation is deeply rooted on the quality of the numbers involved and the existence of a power law. We have observed that the behaviour of the primitive approximation is quadratic and the Li–Broughton correction is quartic, at least for the harmonic oscillator.

We shall study the usefulness of the Richardson extrapolation in terms of the temperature and the action. We start with liquid neon at 25 K. The values for the total energy obtained with the primitive approximation given in table 7.8, along with the results for the same system using a Li–Broughton correction given in table 7.9, are plotted in figure 7.4. The values of the plot do not encompass the tail correction to the potential energy for that system, however, this fact does not affect the quality of the extrapolations or the fits, since it is just a global shift in the energy axis.

The filled symbols of 7.4 are results obtained with the primitive approximation, while the clear symbols represent energies obtained with the Li–Broughton correction. The circles are the estimated values of the energy obtained with the code, and the triangles are the results in the diagonal of the the Richardson extrapolation matrix explained in § 6.3. The dashed line is a quadratic fit of the energy values obtained with the primitive approximation and the solid line is a quartic fit of Li–Broughton's energies. The inset is a detail of the asymptotic region.

Regarding the primitive approximation, we observe that the Richardson extrapolation systematically improves the energy towards the asymptotic value. With just 32 beads and a primitive approximation the right energy is obtained. Since Li–Broughton already gave the asymptotic result with just 2 beads, the Richardson extrapolation little can do. Both fits were performed with the data that belongs to the asymptotic region. For the primitive action that means discarding the energies below 4 beads, and for Li–Broughton below 2 beads.

A more demanding system is ^{4}He at 5 K. In figure 7.5 we plotted the total energy E_{T}, given by the thermodynamic estimator, found on table 7.10. The energies plotted do not include the tail correction.

The Richardson extrapolation yields the right result with 32 beads. The asymptotic energy is reached with 64 beads. The energy fits to a parabola only for simulations with more than 8 beads. We did not show extrapolation values of Li–Broughton's energies since they do not have a fourth order behaviour at this low temperature.

Table 7.10 also includes the explicit values of two different Richardson extrapolations. Column $R_{\mathrm{T}}^{\mathrm{M}}$ is the second–order two–point extrapolation to a finite value of beads, given by equation (6.10). On the other hand, column R_{T} has the data of the second–order two–point Richardson extrapolation to infinity, given by (6.2). At the end both methods converge to the same value, however, the results obtained with (6.10) reach the expected value a little bit faster.

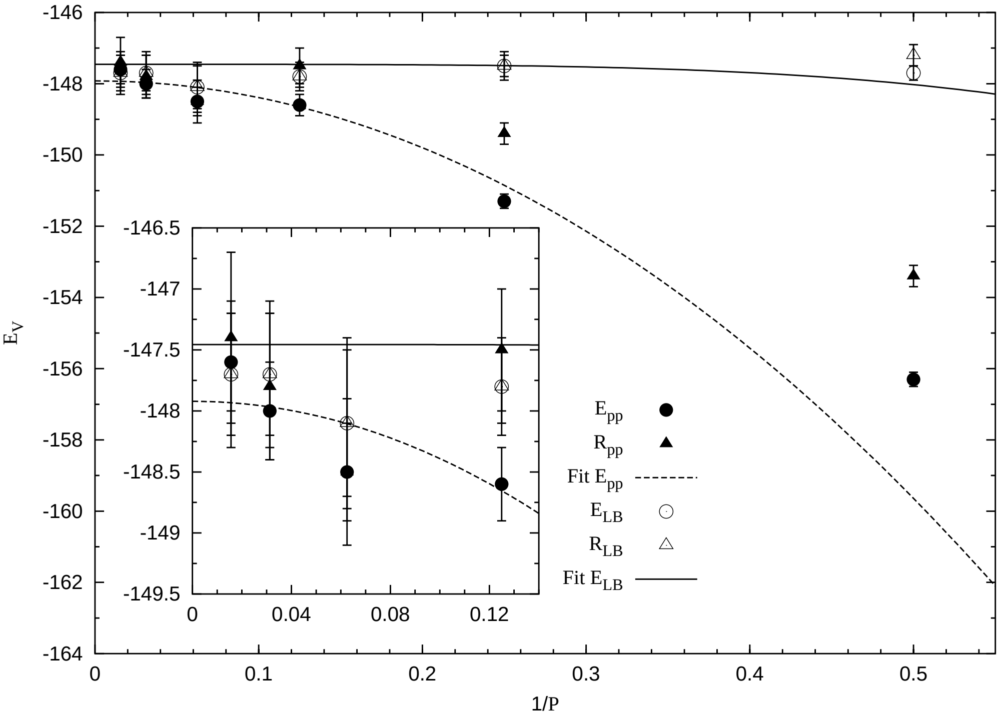


Fig. 7.4: Ne25. Total energy of neon using primitive and Li–Broughton action. R_{pp} and R_{LB} diagonals of the Richardson extrapolation matrices. Virial energy estimator.

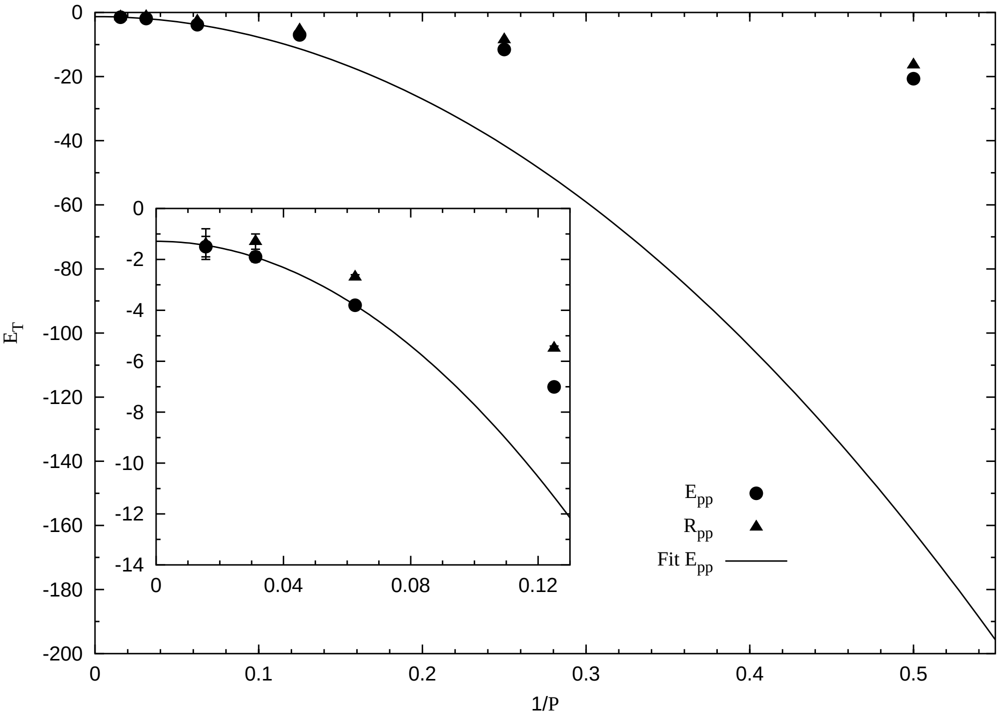


Fig. 7.5: He5LJ. Total energy of helium using primitive action. $R_{\rm pp}$ diagonal of the Richardson extrapolation matrix. Thermodynamic estimator. Quadratic fit of the energy: solid line

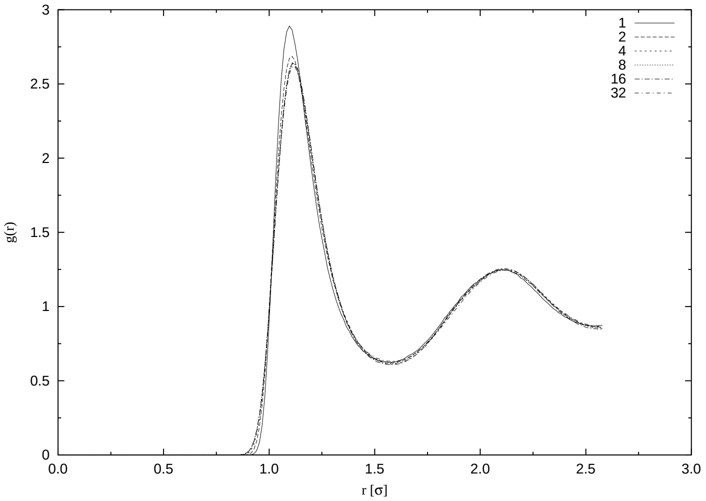


Fig. 7.6: Ne25. Pair distribution function. Primitive action. Behaviour of $g(r)$ in terms of the number of beads

7.2.5 Radial distribution function

For the radial distribution function just a few Monte Carlo steps once the system has been thermalised suffice to yield an accurate result of the $g(r)$. We do not include error bars in our $g(r)$ plots since the error is smaller than the line width.

Rather than discussing about the problems related to the calculation of the radial distribution function, that essentially do not exist, we will show the differences between a classical and a quantum $g(r)$.

Figure 7.6 shows plots of the pair distribution functions obtained with simulations of neon at 25 K using different number of beads, and a primitive action. The plots range from 1 bead, the classical simulation, up to 32 beads. For more than 4 beads all the plots are the same, in agreement with the results obtained for the energy dependence on the number of beads for that system. The main difference between a classical and a quantum $g(r)$ is observed at the first peak.

An equivalent figure is given by figure 7.7, being the action chosen the only difference. However, since for that figure we used a Li–Broughton action, the plot of the $g(r)$ related to 1 bead does not longer represents

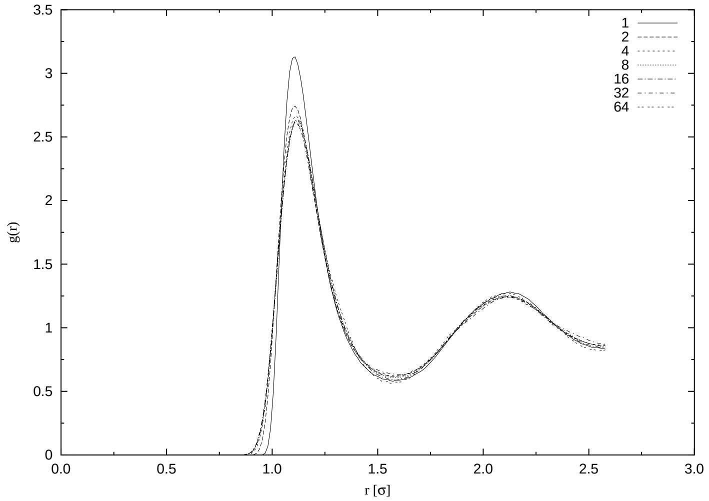


Fig. 7.7: Ne25. Pair distribution function. Li–Broughton action. Behaviour of $g(r)$ in terms of the number of beads

a classical pair distribution function. The Li–Broughton action always encompasses quantum corrections. In order to observe the differences between a classical pair distribution action and a first order quantum correction we plotted together the $g(r)$ for 1 bead using a primitive approximation and a Li–Broughton action in figure 7.8.

The pair distribution functions obtained with either action for asymptotic values of beads coincide. We showed § 7.2.2 that 8 beads was sufficient for both actions to yield the right asymptotic energy. In agreement with that, the pair distribution functions calculated using 8 beads for both actions, are the same.

The quantum effects are more explicit in helium at 5 K. For that system 32 beads are required in order to achieve the asymptotic value when a Li–Broughton action is used. This is also in agreement with the behaviour of the energy. The differences between a simulation with 1 bead and 32 beads are more dramatical. From figure 7.10 it is possible to infer that a classical study of ^{4}He at 5 K would yield a solid, instead of a liquid, as it is shown by the plot of the $g(r)$ using 32 beads. The quantum effects are also visible at the beginning of the pair distribution

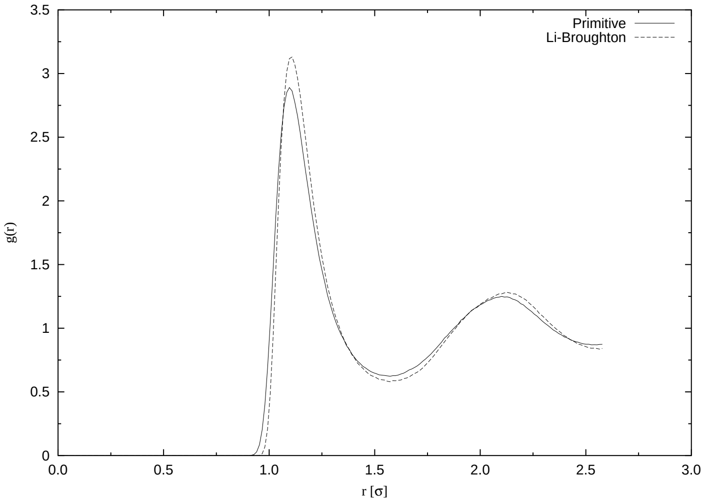


Fig. 7.8: Ne25. Pair distribution functions using 1 bead. Primitive and Li–Broughton actions

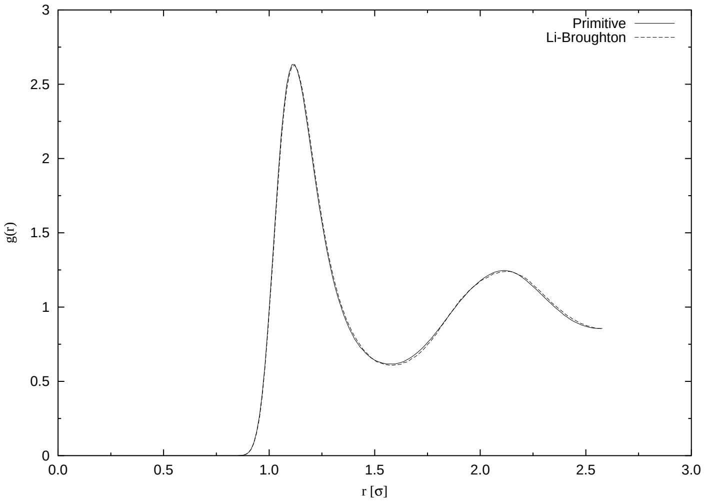


Fig. 7.9: Ne25. Pair distribution functions using 8 beads. Primitive and Li–Broughton actions

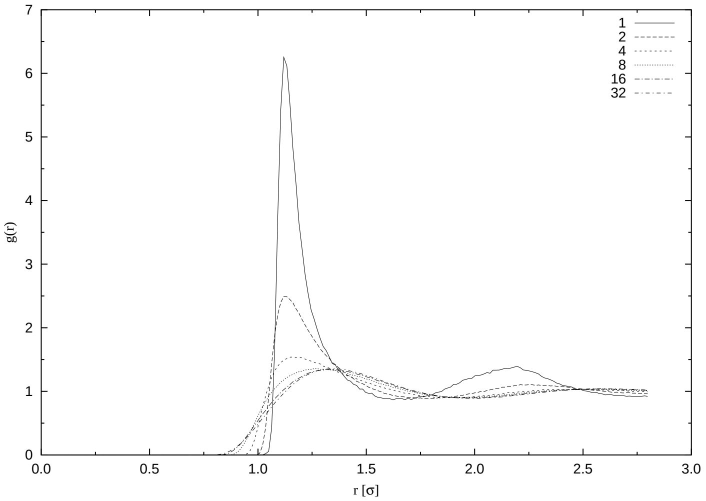


Fig. 7.10: Pair distribution function for different number of beads. Li–Broughton action. Helium 5 K. Lennard–Jones potential.

function. The slope of the first peak of the $g(r)$ is almost infinity and the function becomes different from zero just past one σ. This is the behaviour of a classical solid very compressed. On the hand, the $g(r)$ associated to 32 beads has a gentler slope and becomes different from zero at 0.8σ. It is also possible to observe from that figure the delocalisation effects appearing while going from the classical to the quantum regime, expressed in a lower height of the first peak for the quantum $g(r)$, in comparison to the classical one. Also the first peak position of the quantum $g(r)$ is slightly shifted to the right, with respect to the classical plot. Both aspects point out to a major delocalisation in the quantum regime, as it could be expected. In fact, the quantum kinetic energy is significantly larger than the classical estimate $3/2T$.

7.2.6 Momentum distribution

We shall not study in depth the behaviour of the momentum distribution in terms of the temperature at this moment. Chapter 8 is wholly devoted to that task.

Instead, we shall see the differences between the *trail* method, and

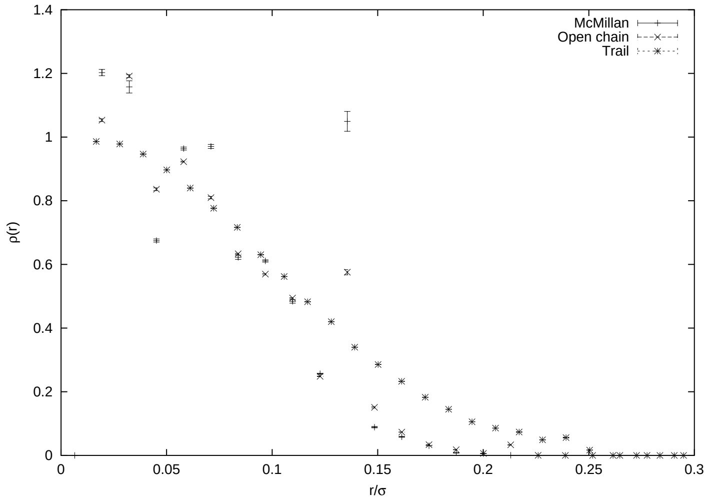


Fig. 7.11: Ne35. Single particle density matrix. Comparison of the three different methods

the other two methods available. Figure 7.11 shows a plot of the single particle density matrix calculated using the trail method, McMillan, and the open chain method. McMillan's and trail's results were calculated during the same simulation. For the open chain we ran an equivalent simulation, including the open polymer, at the same time. We did stop both simulations once the function yielded by the trail method had converged. The plots from the other two methods were still far away from the right function. If we would let the open chain method to grab more statistics, it would arrive to the same result we are giving with the trail method. The same does not happen with McMillan; it will only yield the right behaviour for small r. The differences between the method we propose and the other methods are remarkable.

The full strength of the trail method, and the righteousness of the functions given will be demonstrated in chapter 8.

The Fourier transform of the single particle density matrix obtained with the trail method, yields the momentum distribution. It is depicted in figure 7.12.

In the same way we did for the radial distribution function, we

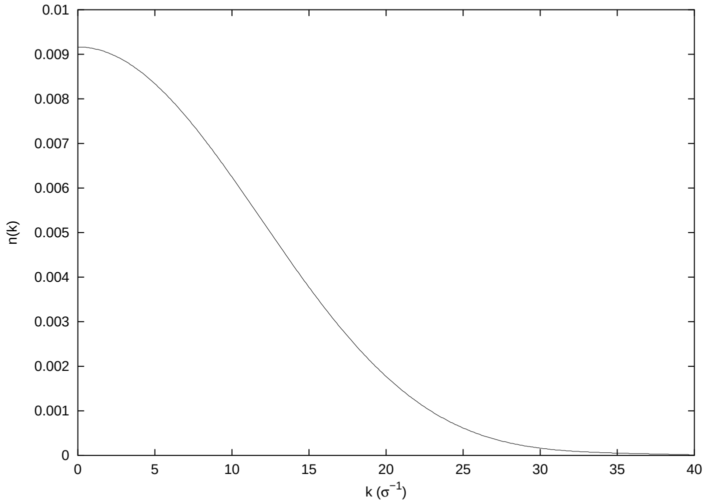


Fig. 7.12: Ne35. Momentum distribution obtained with the trail method

wanted to observe the inclusion of quantum effects as we increased the number of beads, and lowered the temperature. The first system studied was neon at 25 K. Figure 7.13 shows the plots of the momentum distributions for different number of beads. In a first approximation, quantum effects do not change the classical Gaussian behaviour of the momentum distribution, but its width. That is, a near classical momentum distribution is a Gaussian with a width associated to a effective temperature, as it has been explained in § 5.2.2.

In a plot with squared abscissas and a logarithmic scale on the $n(k)$, a Gaussian must be a straight line. Figure 7.14 shows that plot. We observe that the tail of the result obtained with 4 beads is the only part that deviates from a Gaussian. As we showed before, more than 2 beads are required to capture nature of the system when using a Li–Broughton action. In momentum space, large k imply small r, and it is precisely at small r where the first quantum effects can be seen. We also observed that in the plot of $g(r)$ for helium at 5 K, where the quantum behaviour was more evident at small r.

More quantum effects can be seen using helium at 5 K. Figure 7.15 shows the plots of different simulations performed using the Li–

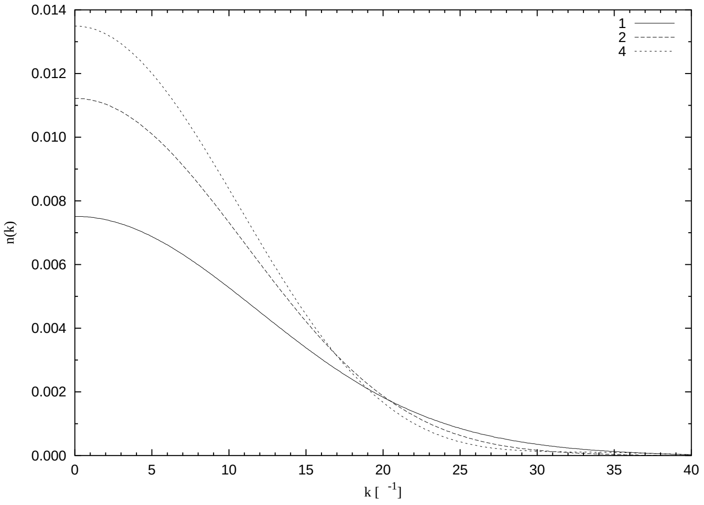


Fig. 7.13: Ne25. Momentum distribution for different number of beads using the Li–Broughton correction

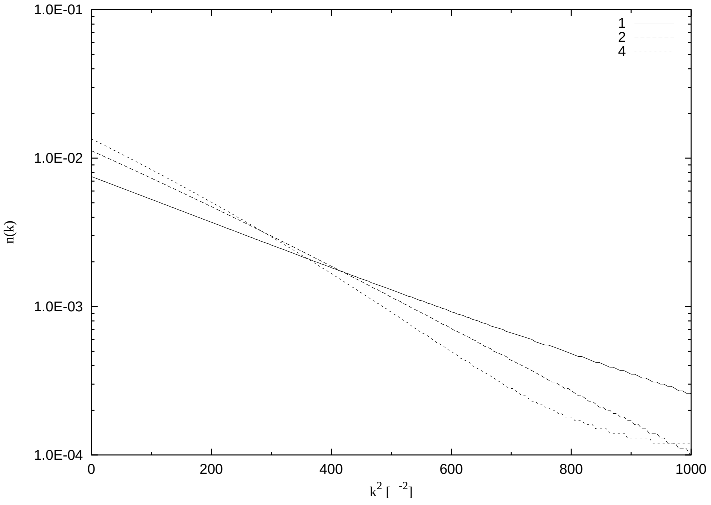


Fig. 7.14: Ne 25. How the momentum distribution deviates from a Gaussian as the number of beads increases. Li–Broughton action

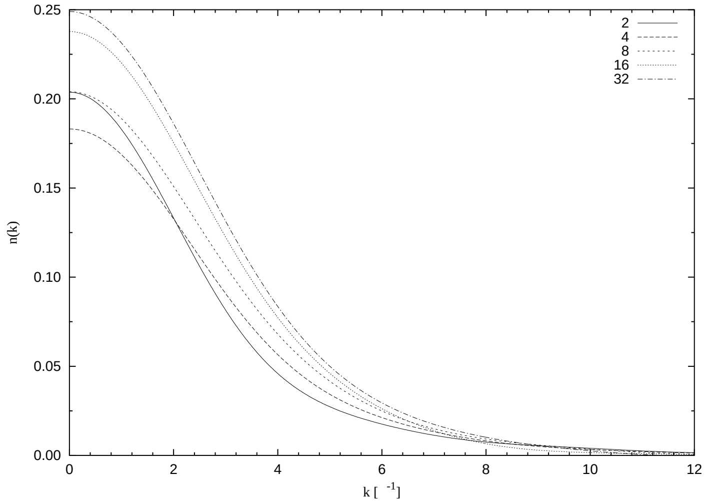


Fig. 7.15: He5LJ. Momentum distributions for different number of beads using Li–Broughton action

Broughton action on a range of beads. In agreement with the results previously exposed for the energy, we observe that 32 beads are required to reach asymptotic behaviour. We also notice that the fluctuation induced by the oscillatory behaviour of the Li–Broughton correction is reflected on the plots associated to 4 and 8 beads.

Figure 7.16 shows that the deviation from a Gaussian is much more accentuated at this low temperature. The momentum distribution for ^{4}He at 5 K cannot be fitted neither to a single Gaussian, nor to a couple of them.

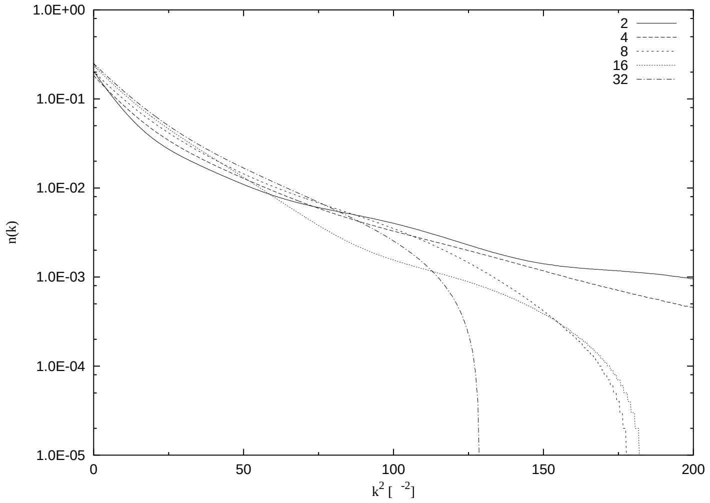


Fig. 7.16: He5LJ. How the momentum distribution deviates from a Gaussian as the number of beads increases. Li–Broughton action

8. MOMENTUM DISTRIBUTION OF QUANTUM LIQUIDS AT FINITE TEMPERATURE

We present results of a path integral Monte Carlo (PIMC) calculation of the momentum distribution of Ne and normal ^{4}He at low temperatures. In the range of temperatures analysed, exchanges can be disregarded and both systems are considered Boltzmann quantum liquids. Their quantum character is well reflected in their momentum distributions which show clear departures from the classical limit. The PIMC momentum distributions are sampled using a new and more efficient method that generalises the standard McMillan approach. Kinetic energies of both systems, as a function of temperature and at a fixed density, are also reported.

8.1 Introduction

The momentum distribution of liquids at low temperatures, where quantum effects are still relevant, is a challenging problem of fundamental interest (Silver and Sokol, 1989), (Glyde, 1994). It provides essential information on the correlations present in the system, which do not show up explicitly in other quantities. In the past years, accurate deep inelastic neutron–scattering experiments in the quantum regime have allowed the study of several aspects of the momentum distributions $n(k)$. However, a clean extraction of $n(k)$ is not possible since instrumental broadening smoothes out fine details of the dynamic structure function. Moreover, if the momentum transfer is not high enough for the impulse approximation to be valid it is necessary to take into account final state effects (Mazzanti et al., 1996). In spite of those problems, it is possible to extract with a reasonable accuracy the kinetic energy from the response at high momentum transfer. That experimental measure usually relies on the assumption of a Gaussian form for $n(k)$, a feature that only a theoretical microscopic analysis can really verify.

In the present chapter, new (Gillan, 1990) PIMC results on the momentum distributions of Ne and normal ^{4}He as a function of temperature (T) are presented. At low T, and in the Boltzmann quantum regime, we have studied the relevance of quantum effects in both $n(k)$ and the single–particle kinetic energies. Special emphasis is made in the departure of $n(k)$ from the Gaussian shape usually assumed in neutron scattering analysis.

The momentum distribution is the Fourier transform of the single–particle density matrix

$$n(r) = Z^{-1} \int d\mathbf{r}_1 d\mathbf{r}_2 \ldots d\mathbf{r}_N \; \rho(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N; \mathbf{r}_1 + \mathbf{r}, \mathbf{r}_2, \ldots, \mathbf{r}_N; \beta) \; . \tag{8.1}$$

In terms of the polymer isomorphism, the calculation of $n(r)$ corresponds to a simulation in which one of the chains is open, beginning at $\mathbf{r}_1$ and ending at $\mathbf{r}_1 + \mathbf{r}$. For the simulations performed in this study, we have used the *trail* method introduced in § 5.2.1.

8.2 Results

The kinetic energies and momentum distributions of liquid Ne and normal liquid ^{4}He have been studied by means of the PIMC method. At a fixed density $\rho_{\mathrm{Ne}} = 0.0363$ Å^{-3}, the properties of Ne have been calculated at temperatures 25.8, 35, 45, and 55 K using as interatomic potential the accurate model of Aziz and Slaman (1989). In the case of ^{4}He, the fixed density is $\rho_{\mathrm{He}} = 0.0218$ Å^{-3} and the temperatures analyzed are 10, 15, and 20 K. The interatomic potential for ^{4}He is the HFD-B(HE) Aziz potential (Aziz et al., 1997) which has proved its high accuracy in microscopic calculations of the equation of state at zero temperature (Boronat and Casulleras, 1994).

PIMC results for the total and kinetic energies of both systems are reported in Table 8.2. In each case, a number of beads large enough to remove discretisation errors has been used. In both systems, and inside the range of temperatures analyzed, the kinetic energy decreases linearly with the temperature. However, as expected due to the low temperature

	T(K)	K(K)	E(K)	$K - K_{\rm cl}$(K)	$(K - K_{\rm cl})/K_{\rm cl}$(%)
Ne	55	90.16(3)	-94.6(2)	7.7	14.0
	45	75.85(2)	-116.4(2)	8.3	18.4
	35	61.76(2)	-138.1(2)	9.3	26.6
	25.8	50.0(1)	-157.2(3)	11.3	43.8
^{4}He	20	39.24(2)	18.60(3)	9.2	46.0
	15	32.61(6)	11.41(7)	10.1	67.3
	10	25.80(6)	4.01(7)	10.8	108.0

Tab. 8.1: Kinetic (K) and total (E) energies of Ne and ^{4}He as a function of the temperature. The densities are 0.0363 and 0.0218 Å^{-3} for Ne and ^{4}He, respectively. $K_{\rm cl}$ is the classical value for the kinetic energy.

of both liquids, the value of the slope does not coincide with the classical value 1.5. For Ne the slope is 1.38 and for ^{4}He is still slightly smaller, 1.34. In Table 1 we report the differences between the kinetic energies and the classical predictions, both in absolute and relative values. The difference increases when the temperature goes down, but even Ne at $T = 55$ K shows a non–negligible correction. On the other hand, the presented results for ^{4}He are in an overall agreement with the PIMC and experimental results of Ceperley et al. (1996). The reported result for Ne at $T = 25.8$ K compares well with the experimental determination of Azuah et al. (1997) ($K^{\rm expt} = 52.9 \pm 2.5$ K).

The PIMC momentum distributions of Ne and ^{4}He are shown in Fig. 8.1 as a function of the temperature. All of them are normalized in the form

$$\frac{1}{(2\pi)^3\rho} \int d\mathbf{k}\; n(k) = 1 \; . \tag{8.2}$$

Both systems behave in a similar way when the temperature increases: $n(k)$ becomes broader due to the increase in the population of higher k states. The shape of the momentum distributions reported in Fig. 8.1 shows a Gaussian resemblance. However, it is not possible to fit the PIMC data with a single normalized Gaussian, even in the *more classical* case of Ne at 55 K.

In Fig. 8.2, the PIMC $n(k)$ is compared with the Gaussian momen-

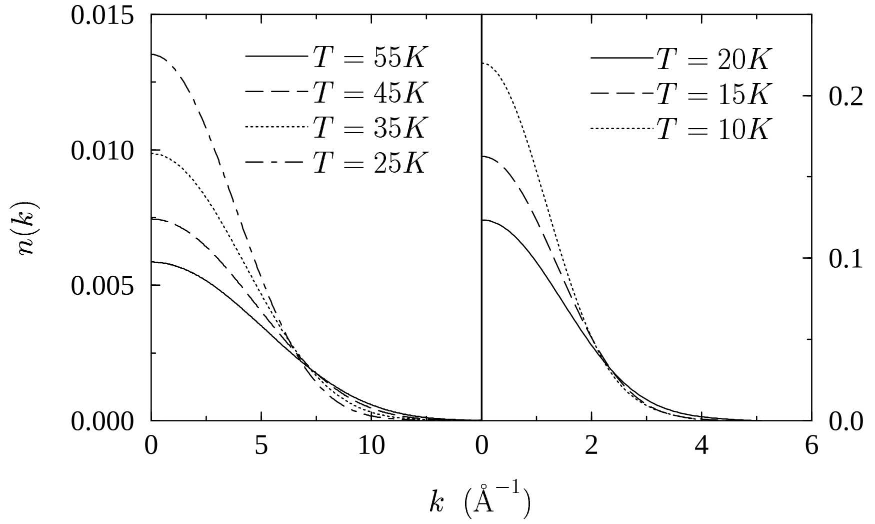


Fig. 8.1: PIMC momentum distributions for Ne (left) and ^{4}He (right) as a function of temperature.

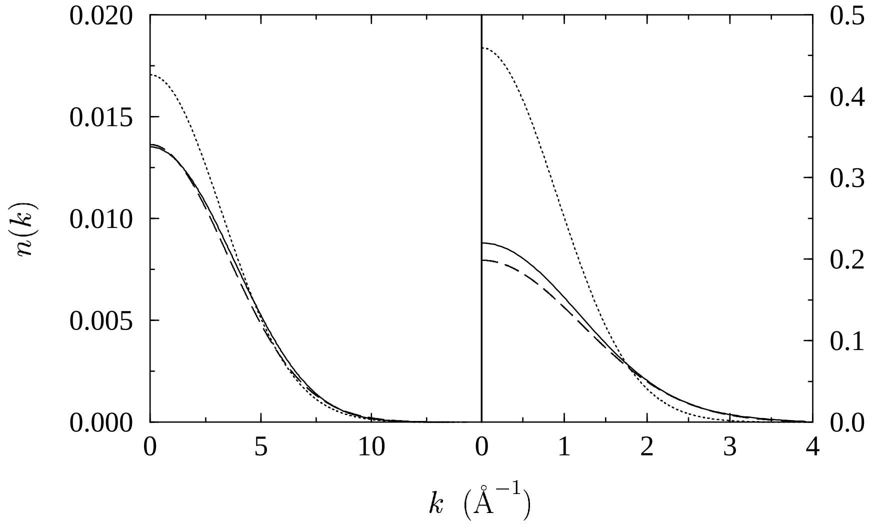


Fig. 8.2: Comparison between the PIMC $n(k)$ (solid line), classical $n(k)$ (dotted line), and classical $n_{\mathrm{cl}}(k)$ with effective temperature (dashed line). Left, Ne at 25.8 K; right, ^{4}He at 10 K.

tum distribution

$$n_{\text{cl}}(k) = 8\rho \left(\frac{\pi\lambda}{T} \right)^{3/2} e^{-\frac{\lambda}{T}k^2} \ , \tag{8.3}$$

at the lowest temperatures of the present analysis: Ne at 25.8 K and ^{4}He at 10 K. As expected, the differences between $n(k)$ and $n_{\text{cl}}(k)$ are important, specially in the case of ^{4}He due to both the lower temperature and its higher quantum nature implied by its smaller mass. For small quantum effects, the increase of the kinetic energy respect to the classical value can be expressed in terms of an effective temperature $T_{\text{ef}} > T$ such that $3T_{\text{ef}}/2 = K_{\text{PIMC}}$. In Fig. 8.2, results for this corrected Gaussian model are also shown. As one can see the resulting momentum distributions reproduce much better the PIMC data than $n_{\text{cl}}(k)$ but there are still significant differences which are again more pronounced for ^{4}He.

In order to see how the deviation from a Gaussian of the PIMC results are more pronounced in the 10 K regime of ^{4}He than in the case of Ne at 25 K we plotted the same results of Fig. 8.2 in a semi–log scale with the abscissæ squared (Fig. 8.3). In such a scale, a Gaussian becomes a straight line. We notice how most of the deviation comes from the tail, as we expected.

In the impulse approximation, the Compton profile can be derived from the momentum distribution § 5.2.3. In figure 8.4 we compare the Compton profiles of Ne (45 K) and ^{4}He (10 K) obtained with the simulation with those that are obtained classically using (Celli et al., 1998)

$$J(y) = (2\pi\sigma^2)^{-1/2} \exp\left(-\frac{y^2}{2\sigma^2} \right) , \tag{8.4}$$

with σ the length parameter of the Lennard–Jones potential.

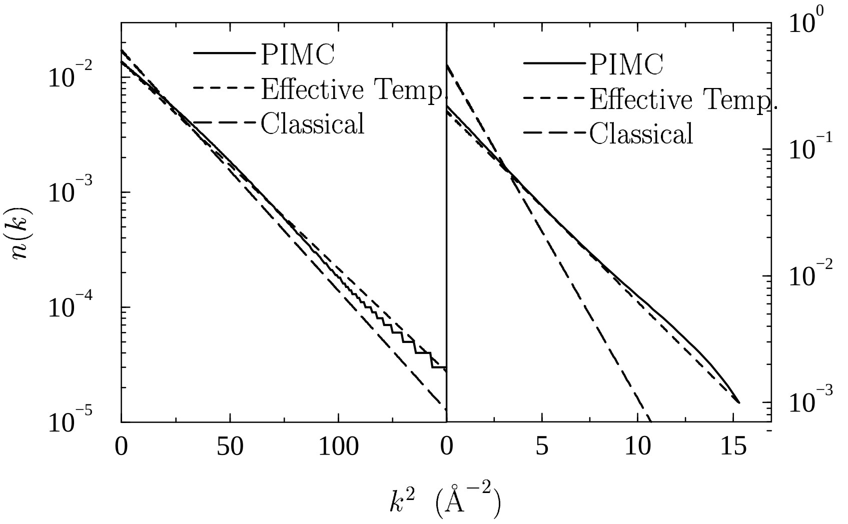


Fig. 8.3: Comparison between the PIMC $n(k)$, classical $n(k)$, and classical $n_{\rm cl}(k)$ with effective temperature in semi–log scale. Left, Ne at 25.8 K; right, ^{4}He at 10 K.

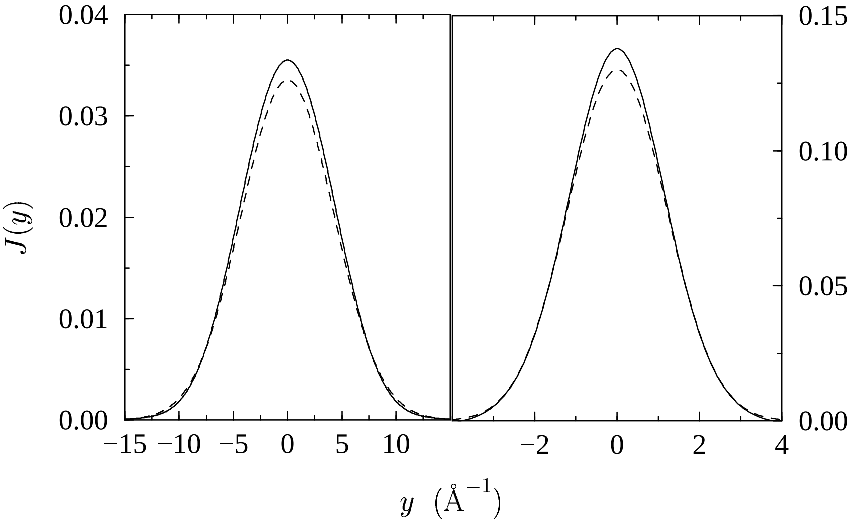


Fig. 8.4: Compton profiles. Left: Ne at 45 K. Right: ^{4}He at 10 K. In both plots solid lines refer to PIMC results, and plot of Eq. (8.4) in dashed lines.

9. SOLID–LIQUID PHASE TRANSITION OF NEON AT 35 K

While thermodynamic properties of solid or liquid neon are well known (Peek et al., 1992), (D. A. Peek et al., 1992), the behaviour of some properties, such as the energy or the density, during the solid–liquid phase transition present some uncertainties (Crawford, 1977). For example, experimental data for the phase diagram, which determines solid and liquid boundaries, present sizeable differences (Fugate and Swenson, 1973), (Lippold, 1969).

Taking advantage of the codes we have developed, we shall study the solid–liquid phase transition of neon along the 35 K isotherm, going from a solid up to a liquid density. The temperature regime chosen is perfectly suitable for our simulation capabilities.

An important feature of PIMC, useful for this kind of problems, is its ability to choose the right phase, solid or liquid, for the given thermodynamic conditions. In other words, we set the density and the temperature of a simulation and we start with a solid configuration; if the thermodynamic conditions are those of a liquid, after a few steps the solid structure has been lost; on the other hand, should the conditions be those of a solid, no matter how many iterations we perform, the system remains solid. There are no spurious or non–physical functions or parameters that have to be introduced *ad hoc* in order to drive the system to the right phase. This is not the case with other simulation techniques, such as diffusion Monte Carlo, where the phase is imposed by the importance sampling (Guardiola, 1998), (Boronat, 2002).

In our work we assumed the atoms obey Boltzmann (distinguishable particle) statistics. Therefore, bead exchanges, due to Bose statistics, have not to be considered. The temperature chosen is high enough so that Bose or Fermi statistics corrections are small, although the system is strongly quantum mechanical (Pollock and Ceperley, 1984). There exists a large range of temperatures where the system can only be accurately described with the inclusion of quantum effects, yet the influence of particle's statistics plays a minor role (Runge and Chester, 1988). Furthermore, in the solid phase, the influence of statistics is almost negligible for all temperatures.

9.1 Simulation conditions

All the simulations have been performed at 35 K. Each simulation has been done at a different molar volume, ranging from 13 cm^3/mole, which is clearly solid for that temperature, up to 17.5 cm^3/mole, corresponding to a liquid. Those molar volumes are, in terms of density, 4.633×10^{-2}Å^{-3} and 3.442×10^{-2}Å^{-3}, respectively. We have used the Aziz HFD–B Ne–Ne pair potential (Aziz and Slaman, 1989). The characteristic length unit has been taken from the Lennard–Jones potential given by (B.3). Since neon solidifies as a fcc crystal, all the simulations have departed from a fcc solid configuration of 108 atoms, with no Gaussian movements around sites.

We showed in § 7.2.1 that 16 beads per polymer ring, at 35 K and 18 cm^3/mole, were enough to accurately simulate neon using the primitive approximation to the action. Therefore, since the conditions of these simulations were similar to those ones, we used throughout all the runnings of this chapter that number of beads. Nevertheless, we performed test runs, in the solid, and in the liquid phase, using 32 beads. The values of the potential, kinetic and total energies obtained are shown in table 9.1. All the energies per particle are expressed in Kelvin. They are compatible with those shown, for the same molar volume, on table 9.2, that has been done using 16 beads. Furthermore, the plots of the structure factor, the radial distribution function, and the momentum distribution, done using 32 beads, have no significant differences with their 16 beads counterparts. From now on, all the results shown will correspond to 16 beads simulations.

The primitive approximation has been the action chosen. Regarding to sampling, we have used a $l = 3$ bisection algorithm. The energies have been calculated using the virial estimator. Since we are interested in distinguishing a solid from a liquid, the structure factor $S(k)$ is of great

$V(\text{cm}^3/\text{mole})$	V	K	E
14.50	-237.9(4)	63.74(6)	-174.2(5)
16.25	-203.4(3)	62.50(6)	-140.9(3)

Tab. 9.1: Potential (V), kinetic (K) and total (E) energies of neon at 35 K as a function of the molar volume. Simulations performed using 32 beads

importance. The theory related to that function is explained on § 5.4. Calculation of the structure from integration of the radial distribution function is prone to error, since 108 atoms are not enough for capturing the long range behaviour of $g(r)$. Therefore, we coded a subroutine that performs the calculation of $S(k)$ during the running, using the positions of the particles and the vertices of a hypercube. The plots of the structure factor shown are the result of that subroutine.

9.2 Results

A single average run takes around fifteen days of Pentium 800 MHz CPU time. The simulations that took most of the time were those close to the phase transition, averaging out a month of CPU time each one.

The energies estimated are shown on table 9.2. The tail correction to the potential energy using 108 atoms and the Aziz HFD–B Ne–Ne pair potential is -9.610 K and has been added to all the potential energies shown in tables and plots within this chapter.

In order to compare our results with previous literature, we found that D. A. Peek et al. (1992) gives a kinetic energy per particle of 60 K for liquid neon at 35 K and 17.4 cm^3/mole. Linear interpolation of our results yield 60.88(6) K for that molar volume. We have not found any other bibliographic source giving results that could be compared with ours. However, since we have used the same code for simulations of neon at 25.8 K, and for that more demanding temperature the values estimated proved to be compatible with others found in literature, we are quite confident of the figures we are reporting.

Through the phase transition we found a discontinuity of 1.4 K in the kinetic energy. It amounts to 2% of the kinetic energy after the solid–liquid transition. The existence of the discontinuity can be clearly

V(cm^3/mole)	V	K	E
13.00	-257.5(2)	68.10(3)	-189.4(2)
13.25	-255.3(3)	67.08(3)	-188.3(3)
13.50	-252.8(2)	66.23(2)	-186.5(2)
13.75	-249.5(3)	65.45(4)	-184.0(3)
14.00	-246.2(2)	64.76(3)	-181.5(2)
14.25	-242.0(2)	64.24(2)	-177.7(2)
14.50	-238.3(2)	63.67(3)	-174.7(2)
14.75	-233.7(4)	63.31(6)	-170.4(6)
15.00	-216.7(3)	64.67(4)	-152.0(3)
15.25	-214.2(2)	64.15(3)	-150.0(2)
15.50	-211.4(3)	63.71(3)	-147.7(3)
15.75	-209.0(2)	63.23(2)	-145.8(2)
16.00	-206.2(1)	62.84(2)	-143.4(2)
16.25	-203.5(1)	62.49(2)	-141.0(1)
16.75	-198.29(8)	61.81(2)	-136.5(1)
17.00	-195.7(2)	61.53(3)	-134.2(2)
17.50	-190.69(7)	60.99(2)	-129.70(7)

Tab. 9.2: Potential (V), kinetic (K) and total (E) energies of neon at 35 K as a function of the molar volume. Upper part of the table corresponds to the solid state, lower part to the liquid phase

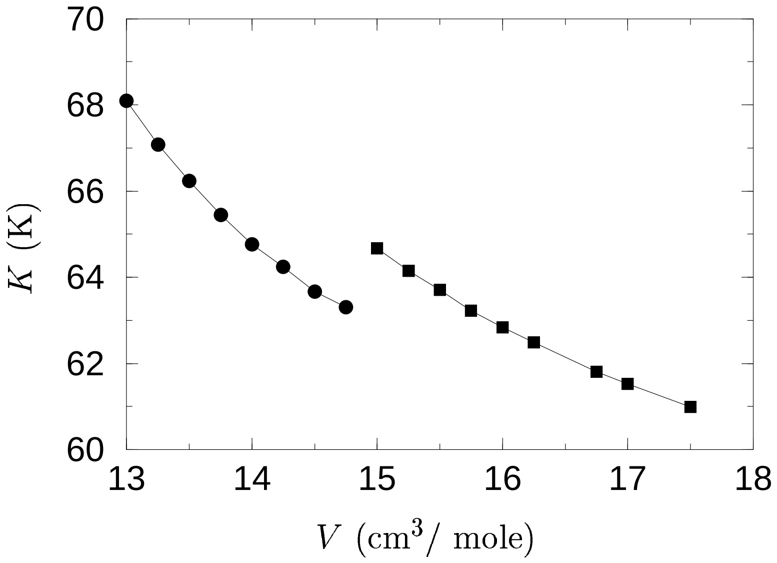


Fig. 9.1: Kinetic energy discontinuity in the solid–liquid phase transition of neon

observed since the error bars of the kinetic energy are less than 0.1% of the estimated values.

Figures 9.1, 9.2, and 9.3 show the plots of the kinetic, potential and total energies, respectively, as a function of the molar volume. The errors are smaller than symbols size. Discontinuities are also observed in the potential, and hence the total energies.

Similar discontinuities have been observed by Ceperley (1995a) and Celli et al. (1998) for ^{4}He.

In order to be sure of the phase transition nature of the discontinuity, we plotted the structure factor of the systems simulated just before (14.75 cm^3/mole) and after transition (15.00 cm^3/mole) on figure 9.4. The vertical dashed lines are the positions of the crystal fcc lattice for 108 atoms and 14.75 cm^3/mole. The open circles are the $S(k)$ at 14.75 cm^3/mole. The peaks of that function are placed exactly on the positions occupied by the crystal, indicating that the solid structure has been preserved. On the other hand, the plot of the $S(k)$ at 15.00 cm^3/mole has the shape of a liquid, and the solid structure has been lost. Notice

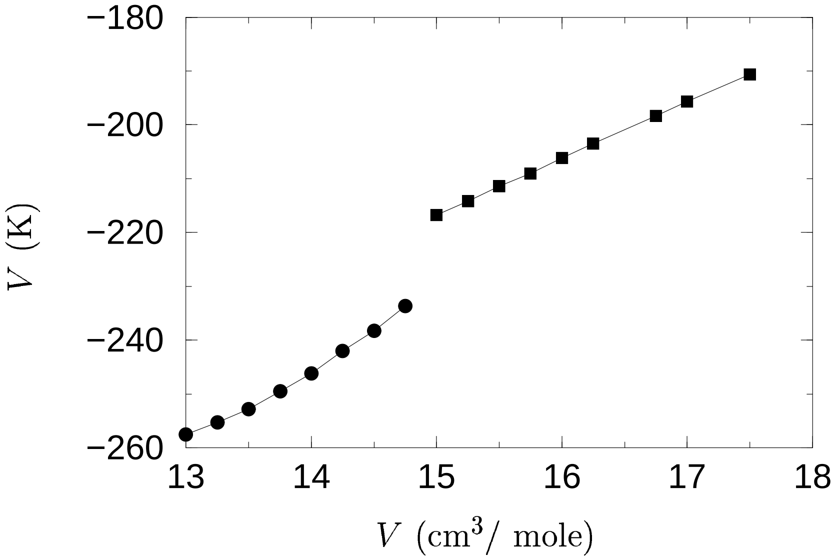


Fig. 9.2: Potential energy discontinuity in the solid–liquid phase transition of neon

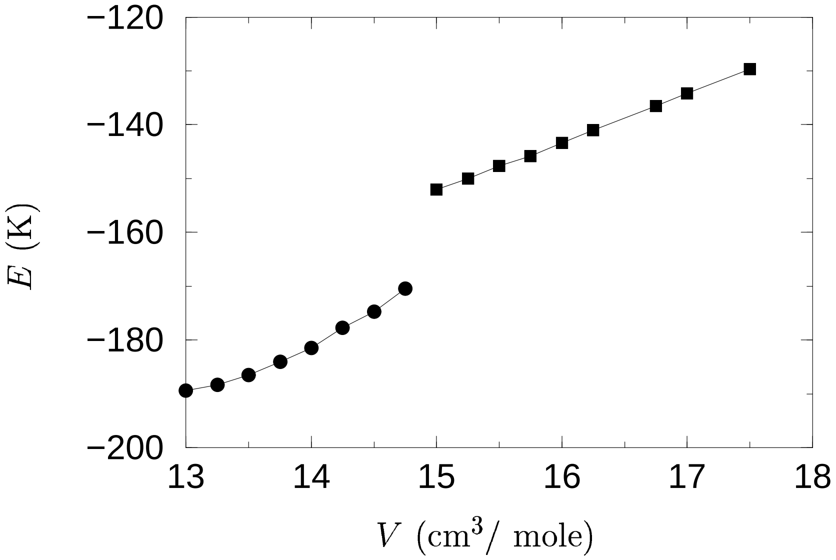


Fig. 9.3: Total energy discontinuity in the solid–liquid phase transition of neon

that these plots are represented with different scales, being the peak of the liquid much more lower than the one of the solid, as we would expect.

Even though the radial distribution function of these two systems are similar, the presence of a third small peak between the two main peaks of the solid $g(r)$ can be perceived (Fig. 9.5). Nevertheless, the definitive proof that characterises the solid and the liquid phase is provided by the structure factor, as it has been seen.

Our results indicate that the solid–liquid phase transition of neon at 35 K occurs between 14.75 cm^3/mole and 15.00 cm^3/mole molar volume. In terms of densities this is between 4.0834×10^{-2}Å^{-3} and 4.0153×10^{-2}Å^{-3}. We have pinpointed that band in the phase diagram given by Crawford (1977), and shown on figure 9.6. Our results are in agreement with the upper curve of the phase diagram measured by Lippold (1969). The lower curve are results from Fugate and Swenson (1973). The discrepancies between these two curves are a proof of the difficulties entailed with this kind of experiments, and also indicate the necessity of more experiments and simulations.

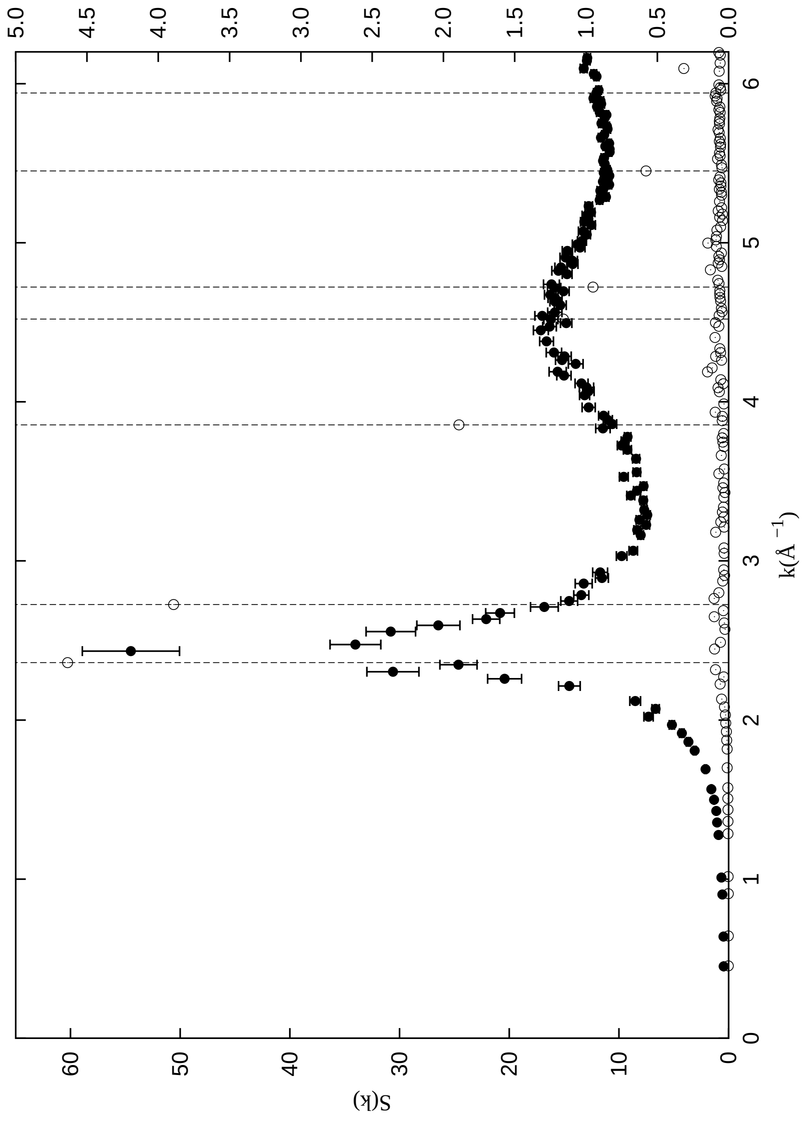


Fig. 9.4: Structure factor of neon at 14.75 cm^3/mole (open circles, scale on left–hand side) and 15.00 cm^3/mole (solid circles, scale on right–hand side). Vertical dashed lines: positions of the structure factor peaks for a fcc crystal of 108 atoms and molar volume 14.75 cm^3/mole

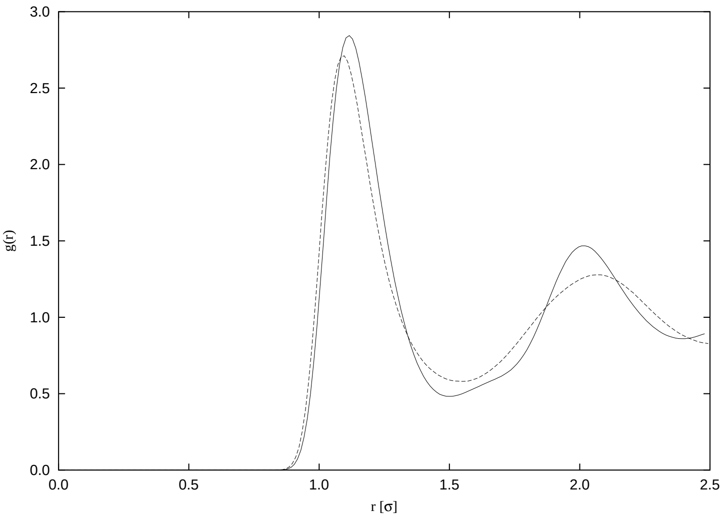


Fig. 9.5: Radial distribution function of neon at 14.75 cm^3/mole (solid line), and 15.00 cm^3/mole (dashed line)

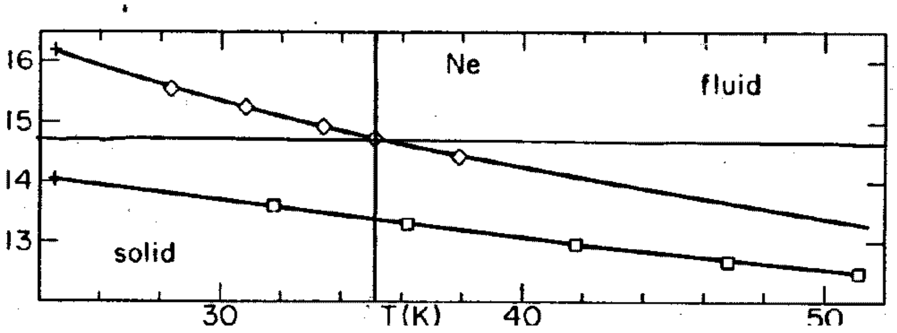


Fig. 9.6: Phase diagram for neon. Reproduced from Crawford (1977). Upper curve: solid–liquid transition by Lippold (1969). Lower curve: Fugate and Swenson (1973). Cross hair: point in the phase diagram estimated with our PIMC simulations

10. CONCLUSIONS

Throughout the many pages we have written so far, some partial conclusions have been already given. Notwithstanding, we summarise them from a global perspective, as well as pointing out where this ongoing project should drive us in future research.

The research we have performed can be divided into two major areas. One is concerned with the methods and the theory of PIMC simulations, which has involved coding and testing; and the other is the application of the tools developed onto current research of quantum liquids at finite temperature. Provided these two dissimilar fields, we shall address our closing remarks keeping them in mind.

The action has been our first issue of study. Using the harmonic oscillator we discarded several combinations of variational improved actions, ending up with two actions to be tested. Li–Broughton's correction proves to be extremely useful whenever it is used in the right regime. For liquid neon at a density of 3.63×10^{-2}Å^{-3}, the adequate regime are the temperatures above 15 K or so. In this favourable regime it produces accurate results with an astonishing low number of beads. However, for lower temperatures, where quantum effects are more noticeable, Li–Broughton's action converges in a non monotonously way, yet with a lower number of beads than the primitive approximation.

Future research might be to code another kind of asymptotic expansion, like the one cited in § 3.3.3. However, it is our belief that these kind of expansions would never allow us to go into systems where particle exchanges are significant. Pollock and Ceperley have performed studies on Bosonic systems using their pair action formulation. We are inclined towards this approach.

Regarding sampling, we are very satisfied with the behaviour of bisection. Even though we did not carry out a full study of staging, we might infer from the theory upon which it is based that its performance must be similar bisection. Bisection represents a major improvement on a bead per bead sampling, and at the moment we shall content with that.

Another asset is the virial estimator of the energy. Its variance has been even an order of magnitude lower than the one of the thermodynamic estimator. This fact can be translated into lower CPU times.

The trail method is part of the techniques we have developed. It gives accurate results for the whole range of r when calculating the single particle density matrix without the necessity of performing a specific simulation for that property. Furthermore, it yields the results faster than the open chain and the McMillan methods.

We have also introduced the Richardson extrapolation for PIMC results. Up until now, authors have not made use of this extrapolation. Richardson extrapolation is useful in three different ways. First, it indicates when a sufficient number of beads has been reached. This feature avoids the necessity of performing simulations with higher number of beads, which are time consuming, and many times not possible due to computing limitations. Second, it allows to give an estimate, with a reasonable degree of accuracy, of the energy of systems that are a little beyond the computing capabilities reached with the chosen action. Finally, it serves as a check of data self–consistency, and to verify that the action is behaving to the order theoretically expected.

Our future research on the methods area will be directed towards accessing lower temperatures. For doing so, we will have to find an action suitable for that task. Furthermore, we need to include bead exchanges within the code. However, with the tools developed so far, we are able to study interesting and challenging systems, as we have demonstrated in the two preceding chapters.

The consistency of our codes has been satisfactorily tested with the momentum distribution helium and neon study, as well as with the solid–liquid neon phase transition.

In the former case we reproduced already known results, and we observed the departure from the classical regime descending down to 10 K. Low temperature single–particle properties of liquid Ne and normal liquid ^{4}He have been studied using the PIMC method. The quantum character of the PIMC simulations allows an accurate calculation of properties like the kinetic energy or the momentum distribution. These quantities are specially sensitive to the quantum nature of the system. In the range of temperatures analyzed here the kinetic energies of both liquids increase linearly with the temperature but with a slope smaller than the classical value. Momentum distributions do not have the classical Gaussian shape and the differences with a Gaussian become larger when the temperature decreases. The definition of Gaussian distributions with an effective temperature reduces significantly the differences between the classical predictions and the PIMC results. Nevertheless, the agreement with the exact results is not very satisfactory, mainly in the case of ^{4}He, pointing again to the non–Gaussian character of $n(k)$.

In the latter, our results were in agreement with previous experiments, and we have seen that PIMC simulations nicely yield the right phase of a system. We have depicted a point of the phase diagram of neon. That point falls within the curve given by Lippold (1969).

No doubt our forthcoming research will also encompass further studies of phase transitions in the regime that is currently accessible to our codes. Even though significant CPU time is required for completing the phase diagram curve of neon, we think this challenging problem is worth the effort.

A. APPENDIX

Derivation of the three dimensional, many particle expression for Li–Broughton's correction

In §3.3.2 we just gave the useful results of Li–Broughton's correction for real systems. That result, in the present form, does not appear in literature, instead the correction in terms of the gradient is found. The expression with the gradient is useless in terms of computer coding. In the light of these facts it is helpful to give the full derivation of the formulæ here presented.

We shall start with equation (3.26) expanding the commutators therein

$$\begin{aligned} C' = [[V,T],V] &= (VT - TV)V - V(VT - TV) \qquad (A.1) \\ &= -(TV)V - V(VT) + 2VTV. \end{aligned}$$

It has been demonstrated that (A.1) gives the right correction to fourth order of the action Raedt and Raedt (1983). Therefore, finding the form of (A.1) will give us the additive term to the primitive action required for the correction.

In preparation for solving (A.1) we need to know the form of the kinetic and potential energy operators, which are

$$T = -\frac{\hbar^2}{2m}\frac{1}{M}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}\nabla^2_{i\alpha}, \qquad (A.2)$$

and

$$V = \frac{1}{M}\sum_{\alpha=1}^{M}\sum_{i<j}V\left(|r_{i\alpha} - r_{j\alpha}|\right). \qquad (A.3)$$

The products TV and TVV in (A.1) yield

$$TV = \left(-\frac{\hbar^2}{2m}\frac{1}{M}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}\nabla_{i\alpha}^2\right)\left(\frac{1}{M}\sum_{i<j}V\left(|r_{i\alpha}-r_{j\alpha}|\right)\right), \tag{A.4}$$

and

$$\begin{aligned} TVV &= \left(-\frac{\hbar^2}{2m}\frac{1}{M}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}\nabla_{i\alpha}^2\right) \\ &\times \left[\left(\frac{1}{M}\sum_{j}V\left(r_{i\alpha}-r_{j\alpha}\right)\right)\left(\frac{1}{M}\sum_{k}V\left(r_{i\alpha}-r_{k\alpha}\right)\right)\right]. \end{aligned} \tag{A.5}$$

Substitution of the previous results into (A.1) give us

$$\begin{aligned} C' &= -\frac{2\lambda}{M^3}\sum_{i<j}V|r_{i\alpha}-r_{j\alpha}|\sum_{\alpha=1}^{M}\sum_{i=1}^{N}\nabla_{i\alpha}^2\sum_{i<k}V|r_{i\alpha}-r_{k\alpha}| \\ &+ \frac{\lambda}{M^3}\left(\sum_{\alpha=1}^{M}\sum_{i=1}^{N}\nabla_{i\alpha}^2\sum_{i<j}V|r_{i\alpha}-r_{j\alpha}|\right)\sum_{i<k}V|r_{i\alpha}-r_{k\alpha}| \\ &+ \frac{\lambda}{M^3}\sum_{i<j}V|r_{i\alpha}-r_{j\alpha}|\left(\sum_{i<k}V|r_{i\alpha}-r_{k\alpha}|\sum_{\alpha=1}^{M}\sum_{i=1}^{N}\nabla_{i\alpha}^2\right), \end{aligned} \tag{A.6}$$

where $\lambda = \hbar^2/2m$ as usual.

Before proceeding let us recall some results from the theory of distributions. Let ψ be a support function, then

$$\begin{aligned} \psi\nabla^2\phi &= \nabla(\psi\nabla\phi) - \nabla\psi\nabla\phi \\ &= -\nabla(\psi\nabla\psi\phi) + \nabla^2\psi\phi. \end{aligned} \tag{A.7}$$

The first terms of each line of (A.7) go to zero as they are divergences of support functions that are to be integrated. Therefore,

$$\psi\nabla^2 = \nabla^2\psi. \tag{A.8}$$

Similarly,

$$\psi\nabla\phi = \nabla(\psi\phi) - (\nabla\psi)\phi, \tag{A.9}$$

which leads to

$$\psi\nabla = -\nabla\psi. \tag{A.10}$$

Using (A.8) and (A.10) in (A.6) we get

$$\begin{aligned}
C' &= -\frac{2\lambda}{M^3}\sum_{i<j} V|r_{i\alpha} - r_{j\alpha}| \sum_{\alpha=1}^{M}\sum_{i=1}^{N}\nabla_{i\alpha}^2 \sum_{i<k} V|r_{i\alpha} - r_{k\alpha}| \\
&+ \frac{\lambda}{M^3}\left(\sum_{\alpha=1}^{M}\sum_{i=1}^{N}\nabla_{i\alpha}^2 \sum_{i<j} V|r_{i\alpha} - r_{j\alpha}|\right)\sum_{i<k} V|r_{i\alpha} - r_{k\alpha}| \\
&+ \frac{\lambda}{M^3}\sum_{\alpha=1}^{M}\sum_{i=1}^{N}\nabla_{i\alpha}^2\left(\sum_{i<j} V|r_{i\alpha} - r_{j\alpha}|\sum_{i<k} V|r_{i\alpha} - r_{k\alpha}|\right) \\
&= -\frac{2\lambda}{M^3}\sum_{i<j} V|r_{i\alpha} - r_{j\alpha}| \sum_{\alpha=1}^{M}\sum_{i=1}^{N}\nabla_{i\alpha}^2 \sum_{i<k} V|r_{i\alpha} - r_{k\alpha}| \\
&+ \frac{\lambda}{M^3}\left(\sum_{\alpha=1}^{M}\sum_{i=1}^{N}\nabla_{i\alpha}^2 \sum_{i<j} V|r_{i\alpha} - r_{j\alpha}|\right)\sum_{i<k} V|r_{i\alpha} - r_{k\alpha}| \\
&+ \frac{\lambda}{M^3}\sum_{\alpha=1}^{M}\sum_{i=1}^{N}\nabla_{i\alpha}^2\left(\sum_{i<j} V|r_{i\alpha} - r_{j\alpha}|\right)\sum_{i<k} V|r_{i\alpha} - r_{k\alpha}| \\
&+ \frac{\lambda}{M^3}\sum_{i<j} V|r_{i\alpha} - r_{j\alpha}| \sum_{\alpha=1}^{M}\sum_{i=1}^{N}\nabla_{i\alpha}^2 \sum_{i<k} V|r_{i\alpha} - r_{k\alpha}| \\
&= \frac{\lambda}{M^3}\Bigg[2\sum_{\alpha=1}^{M}\sum_{i=1}^{N}\nabla_{i\alpha}^2\left(\sum_{i<j} V|r_{i\alpha} - r_{j\alpha}|\right)\sum_{i<k} V|r_{i\alpha} - r_{k\alpha}| \\
&- \sum_{i<j} V|r_{i\alpha} - r_{j\alpha}| \sum_{\alpha=1}^{M}\sum_{i=1}^{N}\nabla_{i\alpha}^2 \sum_{i<k} V|r_{i\alpha} - r_{k\alpha}|\Bigg] \\
&= -\frac{\lambda}{M^3}\sum_{i<j} V|r_{i\alpha} - r_{j\alpha}| \sum_{\alpha=1}^{M}\sum_{i=1}^{N}\nabla_{i\alpha}^2 \sum_{i<k} V|r_{i\alpha} - r_{k\alpha}|. \qquad \text{(A.11)}
\end{aligned}$$

Finally, we have

$$C' = -\frac{\lambda}{M^3}\sum_{\alpha=1}^{M}\left[\left(\sum_{i=1}^{N}\nabla_{i\alpha}\sum_{i<j}V|r_{i\alpha}-r_{j\alpha}|\right)\left(\sum_{l=1}^{N}\nabla_{l\alpha}\sum_{l<k}V|r_{l\alpha}-r_{k\alpha}|\right)\right]. \tag{A.12}$$

The new term goes as $e^{-\frac{\beta^3}{24}[[V,T],V]}$.

Thus Li–Broughton's corrections yields

$$V_{\mathrm{LB}} = \frac{\hbar^2}{24m}\frac{\beta^2}{M^3}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}\mathbf{F}_{i\alpha}\cdot\mathbf{F}_{i\alpha} \tag{A.13}$$

with

$$\begin{aligned} F^a_{i\alpha} &= \sum_{k\neq i}\nabla^a_{i\alpha}V\left(|r_{i\alpha}-r_{k\alpha}|\right) \\ &= \sum_{k\neq i}\frac{dV\left(|r_{i\alpha}-r_{k\alpha}|\right)}{d\left(|r_{i\alpha}-r_{k\alpha}|\right)}\frac{\left(r_{i\alpha}-r_{k\alpha}\right)^a}{|r_{i\alpha}-r_{k\alpha}|}, \end{aligned} \tag{A.14}$$

where a refers to the spatial coordinate index i.e., x, y, z.

B. APPENDIX

Bisection code

We include here a full FORTRAN 77 code that performs a general bisection algorithm of level l. Moreover, the code is presented in a form that can be easily converted into a complete PIMC code. Since bisection exactly samples the kinetic part of the action, it becomes very entangled with the action itself. Therefore, we also explicitly include the code for a primitive approximation to the action. We have chosen not to use in this code the Li–Broughton action in order to focus the reader in the bisection algorithm.

The subroutines with an empty parameter line are listed within the code just to indicate where to put them. The interested reader should write these subroutines explicitly.

As general notation rule, all the names or variables appearing within the text will be typeset in `verbatim` when they are referring to names or variables in the code. There are several general considerations that must be taken into account in order to use this code directly:

- This code is intended for simulating liquids or solids within a cubic box with periodic boundary conditions. The length of the box is L (`L`) and the space coordinates are defined from $-L/2$ to $L/2$ (`L2`). The size of the box is calculated inside of the code in terms of the density required (`rhor`) and the typical length of the system given in terms of σ (`sgm`).
- The calculations are carried out in reduced units,

$$T^* = \frac{k_\mathrm{B}T}{\epsilon}, \tag{B.1}$$

$$\rho^* = \rho\sigma^3, \tag{B.2}$$

where $k_{\mathrm{B}} = 1.381 \times 10^{-23}$ J/K, and σ and ϵ are given by the Lennard–Jones potential

$$V_{\mathrm{LJ}} = \epsilon \left(\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^{6} \right). \tag{B.3}$$

In the case of neon $\epsilon = 5.084 \times 10^{-22}$ J and $\sigma = 2.789 \times 10^{-10}$ m. For ^{4}He $\epsilon = 10.21651$ K and $\sigma = 2.556 \times 10^{-10}$ m.

- The physical parameters defining the system are all given in SI units. They are: temperature (`Tr`), Lennard–Jones' σ (`sgm`), density ρ (`rhor`), and the atomic weight of the element in uma (`atw_uma`). In the code given below these values are set for neon at 25.8 K.

- Vector `p(k,j,i)` contains `k`–th Cartesian coordinate of the `j`–th bead belonging to the `i`–atom. The maximum number of atoms allowed in the simulation is dimensioned by `nax`, and the maximum number of beads is dimensioned by `nbx`. The number of atoms used in the simulation is given by `na`, and the number of beads per atom used is given by `nb`.

- A full pass throughout the whole code is defined as a Monte Carlo step. The number of Monte Carlo steps desired is given by `nsp`. The number of blocks of Monte Carlo steps is given by `nbl`.

- The desired level of bisection is set with the variable `nlv`. The level that at any given moment is being carried out is indicated by `lev`.

- The function `POT(r)` is the potential, and the parameter `r` is the distance between equally labelled beads of different atoms.

- The following subroutines are not explicitly written, however, they are expected to perform the following tasks:

  - `PREPARESIM()`: to initialise the simulation. The most important thing to be done is to fill in the vector `p(k,j,i)`.
  - `MOVECM()`: in order to guarantee a speedy convergence all the atoms must be moved as a whole every once in a while.

The movement of every atom is proposed displacing all its constituting beads the same distance, in that way inter–bead distance within a polymer ring are preserved. This centre of mass movement is performed using a standard Metropolis algorithm. A jump parameter that yields roughly a fifty per cent acceptance of centre of mass movements must be given at the beginning of the simulation.

- `PROPERTIES()`: inside this subroutine should be placed all the subroutines that calculate properties, such as the energy or the single particle density matrix.
- `AVERAGES()`: the values calculated with `PROPERTIES()` must be averaged out using the number of times the code has passed through `PROPERTIES()`.
- `EMPTYCOUNTERS()`: the values calculated using `PROPERTIES()` will be stored in variables and arrays that must be equaled to zero after averaging with `AVERAGES()` in order not to accumulate the same value many times.
- `INITCOUNTERS()`: just assign zero to all variables.
- `ran1(idum)`: this is the random number generator given in *Numerical Recipes* Press et al. (1999).

```
IMPLICIT REAL*8(a-h,o-z)
REAL*8 L,L2,Lv,Kn,Ko
PARAMETER (hbar_eVs = 6.582122020d-16)
PARAMETER (slite_ms = 299792458.d0)
PARAMETER (uma_eV   = 931.4943228d6)
PARAMETER (kB_JK    = 1.3806503d-23)
PARAMETER (K_eV     = 8.617385d-5)
PARAMETER (nax=270,nbx=128)
DIMENSION p(3,nbx,nax)
DIMENSION pnew(3,0:257)
PARAMETER (naux3 = nbx * 3)
DIMENSION k(0:256),Uprt(10),irvec(10)
DIMENSION pn(3,0:257),po(3,0:257),AV(10,2)
DIMENSION ranv(naux3)
PARAMETER (PI25 = 3.14159265358979323846264338327950d0)
CALL INITCOUNTERS()
na     = 108
nb     = 32
nbl    = 500
nsp    = 100
nlv    = 3
idum   = 747
sgm    = 2.789d-10
atw_uma= 20.1796d0
rhor   = 3.63d28
Tr     = 25.8d0
rlamda = (hbar_eVs * hbar_eVs * slite_ms * slite_ms)/
&        (2.d0 * uma_eV * atw_uma * K_eV * sgm * sgm)
L      = ((DBLE(na) / rhor)**(1.d0/3.d0)) / sgm
L2     = L * 0.5d0
Lv     = 1.d0 / L
tauv   = DBLE(nb) * Tr
tau    = 1.d0 / tauv
pi     = PI25 + SIN(PI25)
pi2    = 2.d0 * pi
nrg    = 2 ** nlv
Utot   = 0.d0
ipass  = 0
```

Initialising variables. `tau` is the imaginary time step.

```
CALL PREPARESIM()
DO ibl = 1, nbl
CALL EMPTYCOUNTERS()
DO isp = 1, nsp
CALL MOVECM()
DO i = 1, na
```

Iterations over blocks, Monte Carlo steps, and atoms. Atoms are sampled sequentially.

```
k0 = INT((ran1(idum) * nb) + 1.d0)
IF (k0 .EQ. (nb + 1)) k0 = nb
k(0) = k0
k(nrg) = k0 + nrg
IF (k(nrg) .GT. nb) k(nrg) = k(nrg) - nb
```

Beads are chosen at random preserving cyclic properties of the polymer ring.

```
DO lev = nlv, 1, -1
  lrg = 2 ** lev
  lrg2 = 2 ** (lev - 1)
  nin = 2 ** (nlv - lev)
  lac=0
  ifrg = 1
  alpha = SQRT( DBLE( 2 ** (lev - 1) ) * rlamda * tau )
  lran = 3 * nin
  IF ( MOD(lran,2) .EQ. 1 ) lran = lran + 1
```

Preparing auxiliary level variables. `lev` stands for current level.

```
  DO iran = 1, lran, 2
    r1 = ran1(idum)
    r2 = ran1(idum)
    abm = SQRT(-2.d0 * LOG(r1))
    bbm = pi2 * r2
    ranv(iran)   = abm * COS(bbm)
    ranv(iran+1) = abm * SIN(bbm)
  END DO
```

Box–Muller generator of normal distributed numbers for exactly sampling the kinetic part of the action.

```
  iran = 1
  DO lin = 1, nin
    IF (ifrg .EQ. 1) THEN
      lac = lac + lrg2
      ifrg = 0
    ELSE
      lac = lac + lrg
    END IF
    k(lac) = k0 + lac
    IF (k(lac) .GT. nb) k(lac) = k(lac) - nb

    xnext = p(1,k(lac+lrg2),i) - p(1,k(lac),i)
    ynext = p(2,k(lac+lrg2),i) - p(2,k(lac),i)
    znext = p(3,k(lac+lrg2),i) - p(3,k(lac),i)

    xprev = p(1,k(lac),i) - p(1,k(lac-lrg2),i)
    yprev = p(2,k(lac),i) - p(2,k(lac-lrg2),i)
    zprev = p(3,k(lac),i) - p(3,k(lac-lrg2),i)
```

Mapping between $(\mathbf{r}_0, \ldots, \mathbf{r}_{2^l})$ and the chain segment obtained starting with the bead `k0`.

```
    xnext = xnext - L * ANINT(xnext * Lv)
    ynext = ynext - L * ANINT(ynext * Lv)
    znext = znext - L * ANINT(znext * Lv)

    xprev = xprev - L * ANINT(xprev * Lv)
    yprev = yprev - L * ANINT(yprev * Lv)
    zprev = zprev - L * ANINT(zprev * Lv)

    pxnext = p(1,k(lac),i) + xnext
    pynext = p(2,k(lac),i) + ynext
    pznext = p(3,k(lac),i) + znext

    pxprev = p(1,k(lac),i) - xprev
    pyprev = p(2,k(lac),i) - yprev
    pzprev = p(3,k(lac),i) - zprev
```

Application of the generalised version of equations (4.33) and (4.34).

```
    pn(1,lac) = 0.5d0 * (pxnext + pxprev)
&             + ranv(iran) * alpha
    iran = iran + 1
    pn(2,lac) = 0.5d0 * (pynext + pyprev)
&             + ranv(iran) * alpha
    iran = iran + 1
    pn(3,lac) = 0.5d0 * (pznext + pzprev)
&             + ranv(iran) * alpha
    iran = iran + 1

    pn(1,lac) = pn(1,lac) - L * ANINT(pn(1,lac) * Lv)
    pn(2,lac) = pn(2,lac) - L * ANINT(pn(2,lac) * Lv)
    pn(3,lac) = pn(3,lac) - L * ANINT(pn(3,lac) * Lv)

    po(1,lac) = p(1,k(lac),i)
    po(2,lac) = p(2,k(lac),i)
    po(3,lac) = p(3,k(lac),i)
  END DO  !-------------------------------END DO lin = 1
```

Application of the generalised version of equations (4.36) and (4.35).

```
  IF (lev .NE. nlv) THEN
    namvdold = 2 ** (nlv - lev) - 1
    lac=0
    DO ja = 1, namvdold
      lac = lac + (2 * lrg2)

      pnew(1,lac) = p(1,k(lac),i)
      pnew(2,lac) = p(2,k(lac),i)
      pnew(3,lac) = p(3,k(lac),i)

      p(1,k(lac),i) = po(1,lac)
      p(2,k(lac),i) = po(2,lac)
      p(3,k(lac),i) = po(3,lac)
    END DO
  END IF
```

Saving old positions in case all the levels are not completed.

```
  namvd = 2 ** (nlv - lev + 1) - 1
  AVov = 0.d0
  DO ia = 1, na-1
    lac=0
    DO ja = 1, namvd
      lac = lac + lrg2
      j = k(lac)
      DO iaa=ia+1,na
        CALL DISTANCE(ia,j,iaa,L,Lv,r,p)
        IF (r .LE. L2) AVov = AVov + POT(r)
      END DO
    END DO
  END DO
  AV(lev,1) = DBLE( 2 ** (lev-1) ) * tau * AVov
```

Calculating the potential part of the action using the original (old) positions

```
IF (lev .NE. nlv) THEN
  lac=0
  DO ja = 1, namvdold
    lac = lac + (2 * lrg2)
    p(1,k(lac),i) = pnew(1,lac)
    p(2,k(lac),i) = pnew(2,lac)
    p(3,k(lac),i) = pnew(3,lac)
  END DO
END IF
```

Preparing positions for new action

```
lac=0
ifrg = 1
DO lin = 1, nin
  IF (ifrg .EQ. 1) THEN
    lac = lac + lrg2
    ifrg = 0
  ELSE
     lac = lac + lrg
  END IF
  p(1,k(lac),i) = pn(1,lac)
  p(2,k(lac),i) = pn(2,lac)
  p(3,k(lac),i) = pn(3,lac)
END DO
```

Calculating the new action

```
AVnv = 0.d0
DO ia = 1, na-1
  lac=0
  DO ja = 1, namvd
    lac = lac + lrg2
    j = k(lac)
    DO iaa=ia+1,na
      CALL DISTANCE(ia,j,iaa,L,Lv,r,p)
      IF (r .LE. L2) AVnv = AVnv + POT(r)
    END DO
  END DO
END DO
AV(lev,2) = DBLE( 2 ** (lev-1) ) * tau * AVnv
```

Discounting actions from levels above current level using generalised version of equation (4.40)

```
ipass = ipass + 1
Uprt(lev) = AV(lev,2) - AV(lev,1)
icrtlev = 1
IF (lev .EQ. nlv) THEN
  nact = nlv
ELSE
  nact = lev + 1
END IF
DO ilev = lev, nact
  IF (icrtlev .EQ. 1) THEN
    icrtlev = 0
    Utot = -Uprt(ilev)
  ELSE
    Utot = Utot + Uprt(ilev)
  END IF
END DO
```

```
  IF (Utot .GE. 0.d0) THEN
    CONTINUE
  ELSE
    IF (ran1(idum) .LT. EXP(Utot)) THEN
      CONTINUE
    ELSE
      irvec(lev) = irvec(lev) + 1
      lac=0
      DO ja = 1, namvd
        lac = lac + lrg2
        p(1,k(lac),i) = po(1,lac)
        p(2,k(lac),i) = po(2,lac)
        p(3,k(lac),i) = po(3,lac)
      END DO
    GOTO 100
    END IF
  END IF
```

Metropolis

```
END DO !----------------------------------END DO LEVEL
100   Utot = 0.d0
END DO !----------------------------------END DO ATOMS
CALL PROPERTIES()
END DO !----------------------------------END DO MCS
CALL WRITERESULTS()
END DO !----------------------------------END DO BLOCK
CALL AVERAGES()
STOP
END
```

Ending algorithm and calculating thermal properties

```
SUBROUTINE DISTANCE(i,j,ia,L,Lv,r,p)
IMPLICIT REAL*8(a-h,o-z)
REAL*8 L,Lv
PARAMETER (nax=270,nbx=128)
DIMENSION p(3,nbx,nax)

x = p(1,j,i) - p(1,j,ia)
y = p(2,j,i) - p(2,j,ia)
z = p(3,j,i) - p(3,j,ia)

x = x - L * ANINT(x * Lv)
y = y - L * ANINT(y * Lv)
z = z - L * ANINT(z * Lv)

r = SQRT(x*x + y*y + z*z)

RETURN
END
```

Distance between two equally labelled beads of different atoms

C. APPENDIX

Trail code

The following subroutine calculates the single–particle density matrix within a PIMC simulation using the *trail method* we introduced in §5.2.1. This subroutine is completely portable provided the conventions given in Appendix B are fulfilled.

The random number generator chosen is `ran1` supplied with the book *Numerical Recipes in FORTRAN* (Press et al., 1999).

The subroutine `DISTANCE`, also given herein, calculates the distance between equally labelled beads of different atoms. This is the distance required for the potential part of the action.

In opposition to the bisection method, where only the potential part of the action was required, for estimating the single–particle density matrix is necessary to calculate the kinetic part of the action, as if it were a bead per bead sampling. The subroutine `DISTANCEK` calculates the distance between two beads within the same polymer ring. With that distance it is possible to calculate the kinetic part of the action.

The last portion of code averages the data obtained and normalises the histogram. It has to be included in the main part of the code where these tasks are carried out. This is the reason why is located outside the main subroutine

```
SUBROUTINE TRAIL
& (idum,rlamda,Tr,tau,twl,na,nb,L,L2,L2SQ,Lv,
& ahist,mhist,delr,p,rlv2,TV)
IMPLICIT REAL*8 (a-h,o-z)
REAL*8 L2,L,Lv,L2SQ
PARAMETER (nax=270,nbx=128)
DIMENSION p(3,nbx,nax),psave(3,nbx)
PARAMETER (mdbinx = 256)
PARAMETER (iaux = nbx * 3)
DIMENSION mhist(0:mdbinx), ahist(0:mdbinx)
DATA psave/iaux*0.d0/
SAVE
ndgr = 200
ngrdcr = 0
delr = L / DBLE( 2 * ndgr )
rlamda = (hbar_eVs * hbar_eVs * slite_ms * slite_ms)/
&        (2.d0 * uma_eV * atw_uma * K_eV * sgm * sgm)
twl = SQRT( 2.d0 * rlamda / Tr )
```

Initialisation. The variable `twl` stands for the thermal–wavelength.

```
DO iter = 1, na
imd = INT(ran1(idum) * na) + 1
jmd = INT(ran1(idum) * nb) + 1
IF (imd .EQ. (na + 1)) imd = na
IF (jmd .EQ. (nb + 1)) jmd = nb
xran = ((3.d0 * twl * ran1(idum)) - (twl * 1.5d0))
yran = ((3.d0 * twl * ran1(idum)) - (twl * 1.5d0))
zran = ((3.d0 * twl * ran1(idum)) - (twl * 1.5d0))
```

Calculation of the three–dimensional random vector γ

```
AVov = 0.d0
jcur = jmd
DO jaux = nb+1, 1, -1
  DO ia = 1, na
    IF (ia .NE. imd) THEN
      CALL DISTANCE(ia,jcur,imd,L,Lv,r,p)
      IF (r .LE. L2) AVov = AVov + POT(r)
    END IF
  END DO
  jcur = jcur - 1
  IF (jcur .EQ. 0) jcur = nb
END DO
AVo = tau * AVov
```

Calculation of the potential part of the action prior proposing the trail displacement. The function `POT(r)` is the potential, that must be included within the code.

```
AKo = 0.d0
jcur = jmd
DO jaux = nb, 1, -1
  jp = jcur - 1
  IF (jp .EQ. 0) jp = nb
  CALL DISTANCEK(imd,jcur,jp,L,Lv,rsq,p)
  IF (rsq .LE. L2SQ) AKo = AKo + rsq
  jcur = jcur - 1
  IF (jcur .EQ. 0) jcur = nb
END DO
AKo = Tr * AKo * DBLE(nb) / (4.d0 * rlamda)
```

Calculation of the kinetic part of the action before moving the beads of the open chain.

```
jcur = jmd
DO jaux = nb+1, 1, -1
  IF (jaux .EQ. 1) THEN
    p(1,nb+1,imd) = psave(1,jmd)
    p(2,nb+1,imd) = psave(2,jmd)
    p(3,nb+1,imd) = psave(3,jmd)
  ELSE
    psave(1,jcur) = p(1,jcur,imd)
    psave(2,jcur) = p(2,jcur,imd)
    psave(3,jcur) = p(3,jcur,imd)
    p(1,jcur,imd) = p(1,jcur,imd)
&                 + DBLE(jaux-1) * xran / DBLE(nb)
    p(2,jcur,imd) = p(2,jcur,imd)
&                 + DBLE(jaux-1) * yran / DBLE(nb)
    p(3,jcur,imd) = p(3,jcur,imd)
&                 + DBLE(jaux-1) * zran / DBLE(nb)
    CALL TOBOX(imd,jcur,L,Lv,p)
    jcur = jcur - 1
    IF (jcur .EQ. 0) jcur = nb
  END IF
END DO
```

The trail displacement is proposed. Periodic boundary conditions are taken into account with the subroutine `TOBOX`

```
AKn = 0.d0
jcur = jmd
DO jaux = nb, 1, -1
  jp = jcur - 1
  IF (jp .EQ. 0) jp = nb
  IF (jaux .EQ. 1) jp = nb + 1
  CALL DISTANCEK(imd,jcur,jp,L,Lv,rsq,p)
  IF (rsq .LE. L2SQ) AKn = AKn + rsq
  jcur = jcur - 1
  IF (jcur .EQ. 0) jcur = nb
END DO
AKn = Tr * AKn * DBLE(nb) / (4.d0 * rlamda)
```

The kinetic part of the action associated with the displaced open chain is calculated.

```
AVnv = 0.d0
jcur = jmd
DO jaux = nb+1, 1, -1
  IF (jaux.EQ.1) THEN
    xo = psave(1,jmd)
    yo = psave(2,jmd)
    zo = psave(3,jmd)
    xf = p(1,jmd,imd)
    yf = p(2,jmd,imd)
    zf = p(3,jmd,imd)
    p(1,jmd,imd) = psave(1,jmd)
    p(2,jmd,imd) = psave(2,jmd)
    p(3,jmd,imd) = psave(3,jmd)
    jcur = jmd
  END IF
```

Preparing variables for the potential part of the action.

```
  DO ia = 1, na
    IF (ia .NE. imd) THEN
      CALL DISTANCE(ia,jcur,imd,L,Lv,r,p)
      IF (r .LE. L2) AVnv = AVnv + POT(rlv2,TV,r)
    END IF
  END DO
  jcur = jcur - 1
  IF (jcur .EQ. 0) jcur = nb
END DO
AVn = tau * AVnv
A1 = EXP(-AKn + AKo - AVn + AVo)
```

Calculation of the new potential part of the action, as well as the change in the action.

```
x11 = xo - xf
y11 = yo - yf
z11 = zo - zf
x11 = x11 - L * ANINT(x11 * Lv)
y11 = y11 - L * ANINT(y11 * Lv)
z11 = z11 - L * ANINT(z11 * Lv)
r = SQRT(x11*x11 + y11*y11 + z11*z11)
```

Calculation of the end–to–end distance.

```
IF (r.LE.L2) THEN
mdbin = INT(r / delr) + 1
IF (mdbin .LE. mdbinx) THEN
  ahist(mdbin) = ahist(mdbin) + A1
  mhist(mdbin) = mhist(mdbin) + 1
END IF
END IF
```

Histograms of distances and occurrences

```
jcur = jmd-1
DO jaux = nb, 1, -1
  IF (jcur .EQ. 0) jcur = nb
  p(1,jcur,imd) = psave(1,jcur)
  p(2,jcur,imd) = psave(2,jcur)
  p(3,jcur,imd) = psave(3,jcur)
  jcur = jcur - 1
END DO
p(1,nb+1,imd) = 0.d0
p(2,nb+1,imd) = 0.d0
p(3,nb+1,imd) = 0.d0
END DO
RETURN
END
```

Old positions are restored in order to continue with an all–closed polymers simulation.

```
SUBROUTINE DISTANCE(i,j,ia,L,Lv,r,p)
IMPLICIT REAL*8(a-h,o-z)
REAL*8 L,Lv
PARAMETER (nax=270,nbx=128)
DIMENSION p(3,nbx,nax)

x = p(1,j,i) - p(1,j,ia)
y = p(2,j,i) - p(2,j,ia)
z = p(3,j,i) - p(3,j,ia)

x = x - L * ANINT(x * Lv)
y = y - L * ANINT(y * Lv)
z = z - L * ANINT(z * Lv)

r = SQRT(x*x + y*y + z*z)

RETURN
END
```

Distance between two equally labelled beads of different atoms

```
SUBROUTINE DISTANCEK(i,j,jm,L,Lv,rsq,p)
IMPLICIT REAL*8(a-h,o-z)
REAL*8 L,Lv
PARAMETER (nax=270,nbx=128)
DIMENSION p(3,nbx,nax)

x = p(1,jm,i) - p(1,j,i)
y = p(2,jm,i) - p(2,j,i)
z = p(3,jm,i) - p(3,j,i)

x = x - L * ANINT(x * Lv)
y = y - L * ANINT(y * Lv)
z = z - L * ANINT(z * Lv)

rsq = x*x + y*y + z*z

RETURN
END
```

Squared distance between two neighbouring beads of the same atom

```
SUBROUTINE TOBOX(i,j,L,Lv,p)
IMPLICIT REAL*8(a-h,o-z)
PARAMETER (nax=270,nbx=128)
DIMENSION p(3,nbx,nax)
REAL*8 L,Lv

p(1,j,i) = p(1,j,i) - L * ANINT(p(1,j,i) * Lv)
p(2,j,i) = p(2,j,i) - L * ANINT(p(2,j,i) * Lv)
p(3,j,i) = p(3,j,i) - L * ANINT(p(3,j,i) * Lv)

RETURN
END
```

Keeps periodic boundary conditions

```
ccc TO BE INCLUDED IN THE MAIN CODE FOR AVERAGING PROPERTIES ccc
OPEN(63,FILE="prm.md",STATUS="unknown",ACCESS="append")
rdist = 0.5d0 * delr
DO mdbin = 0, mdbinx
IF (mhist(mdbin) .EQ. 0) THEN
  rmd(mdbin) = 0.d0
ELSE
  rmd(mdbin) = ahist(mdbin) / DBLE(mhist(mdbin))
END IF
WRITE(63,72) rdist,rmd(mdbin)
idbg2 = idbg2 + 1
72         FORMAT(F14.4,F14.4)
rdist = rdist + delr
END DO
CLOSE(63)
```

The file `prm.md` stores the single–particle density matrix for different configurations of the position space sampled in each block.

D. APPENDIX

Virial and thermodynamic energy estimators for the Li–Broughton action

A full derivation of the the thermodynamic and virial estimators of the energy for a Li–Broughton action is herein presented. Even though it is nothing else than an application of derivation rules, it is a little bit tricky. Furthermore, it serves as an example of how to work with similar problems that normally appear when dealing with squared gradients within codes.

Li–Broughton's virial estimator of the energy yields (Li and Broughton, 1987)

$$\begin{aligned} E &= \frac{3}{2}T + \frac{1}{NM}\left\langle \sum_{i=1}^{N}\sum_{\alpha=1}^{M}\left\{ V(\mathbf{r}_{i\alpha}) + \frac{\hbar^2\beta^2}{8mM^2}[\nabla_{i\alpha}V(\mathbf{r}_{i\alpha})]^2 \right\}\right\rangle \\ &+ \frac{1}{2NM}\left\langle \sum_{i=1}^{N}\sum_{\alpha=1}^{M}(\mathbf{r}_{i\alpha}-\mathbf{R}_i)\cdot\nabla_{i\alpha}\left\{ V(\mathbf{r}_{i\alpha}) + \frac{\hbar^2\beta^2}{24mM^2}[\nabla_{i\alpha}V(\mathbf{r}_{i\alpha})]^2 \right\}\right\rangle , \end{aligned} \tag{D.1}$$

with $\mathbf{R}_i$ the i–th atom centre of mass. The previous equation might be expanded into

$$\begin{aligned} E &= \frac{3}{2}kT + \frac{1}{NM}\sum_{i=1}^{N}\sum_{\alpha=1}^{M} V(\mathbf{r}_{i\alpha}) \\ &+ \frac{1}{N}\frac{\hbar^2\beta^2}{8mM^3}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}[\nabla_{i\alpha}V(\mathbf{r}_{i\alpha})]^2 \\ &+ \frac{1}{2NM}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}(\mathbf{r}_{i\alpha}-\mathbf{R}_i)\cdot\nabla_{i\alpha}V(\mathbf{r}_{i\alpha}) \\ &+ \frac{1}{N}\frac{\hbar^2\beta^2}{48mM^3}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}(\mathbf{r}_{i\alpha}-\mathbf{R}_i)\cdot\nabla_{i\alpha}[\nabla_{i\alpha}V(\mathbf{r}_{i\alpha})]^2 \end{aligned} \tag{D.2}$$

where the first term of the right hand side of the equation amounts the classical kinetic energy. The second one is the potential energy. The Li–Broughton correction to the kinetic energy is given by the third term. The fourth term represents the virial estimator of the kinetic energy, and finally, the fifth term gives the Li–Broughton correction to the virial estimator.

From previous work we know that $\sum_{i=1}^{N}\sum_{\alpha=1}^{M}[\nabla_{i\alpha}V(\mathbf{r}_{i\alpha})]^2$ can be translated into a scalar product of forces, given by

$$K = \frac{\beta^2\hbar^2}{8M^3m}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}\mathbf{F}_{i\alpha}\cdot\mathbf{F}_{i\alpha}. \tag{D.3}$$

where $F_{i\alpha}$ is defined in equation (3.50). We also have shown an alternative representation of the fourth term of (D.2) which is more suitable for coding purposes. Thus, the only term that needs a special treatment is the fifth one. In the following lines we shall rearrange it in a way more suitable for coding.

We start with the fifth term and we expand it in an explicit form. In doing so, we shall find useful expressions for representing the other terms in (D.2)

Equations (D.4) and (D.5) present useful results relative to derivatives of Cartesian coordinates operators that will be in the forthcoming derivation of the fifth term of (D.1). The Cartesian coordinate indexes are given by a, b, c.

$$\begin{aligned}
&\frac{\partial}{\partial r_i^a}\left\{\sum_{j\neq i}\frac{dV(\mathbf{r}_{ij})}{dr_i^b}\sum_{k\neq i}\frac{dV(\mathbf{r}_{ik})}{dr_i^b}\right\} \\
= \quad &\frac{\partial}{\partial r_i^a}\left\{\sum_{j\neq i}\frac{r_{ij}^b}{|\mathbf{r}_{ij}|}\frac{\partial V(\mathbf{r}_{ij})}{\partial|\mathbf{r}_{ij}|}\sum_{k\neq i}\frac{r_{ik}^b}{|\mathbf{r}_{ik}|}\frac{\partial V(\mathbf{r}_{ik})}{\partial|\mathbf{r}_{ik}|}\right\}. \qquad \text{(D.4)}
\end{aligned}$$

The derivative $\partial_{r_i^a}$ of the first term of the product yields

$$\begin{aligned}
\frac{\partial}{\partial r_i^a}\sum_{j\neq i}\frac{r_{ij}^b}{|\mathbf{r}_{ij}|}\frac{\partial V(\mathbf{r}_{ij})}{\partial|\mathbf{r}_{ij}|} &= \sum_{j\neq i}\frac{\partial}{\partial r_i^a}\left(\frac{r_{ij}^b}{|\mathbf{r}_{ij}|}\right)\frac{\partial V(\mathbf{r}_{ij})}{\partial|\mathbf{r}_{ij}|} \\
&+ \sum_{j\neq i}\frac{r_{ij}^b}{|\mathbf{r}_{ij}|}\frac{\partial}{\partial r_i^a}\frac{\partial V(\mathbf{r}_{ij})}{\partial|\mathbf{r}_{ij}|} \\
&= \left(\delta_{ab}|\mathbf{r}_{ij}| - r_{ij}^b\frac{r_{ij}^a}{|\mathbf{r}_{ij}|}\right)|\mathbf{r}_{ij}|^{-2}\frac{\partial}{\partial|\mathbf{r}_{ij}|}V(\mathbf{r}_{ij}) \\
&+ \frac{r_{ij}^b}{|\mathbf{r}_{ij}|}\frac{\partial^2}{\partial|\mathbf{r}_{ij}|\partial r_i^a}V(\mathbf{r}_{ij}). \qquad \text{(D.5)}
\end{aligned}$$

Provided with the previous results, we derive the fifth term in the following way

$$\sum_{j\neq i}\nabla_i^a V(\mathbf{r}_{ij}) = \sum_{j\neq i}\frac{d\mathbf{r}_{ij}}{dr_i^a}\frac{\partial}{\partial|\mathbf{r}_{ij}|}V(\mathbf{r}_{ij}) = \sum_{j\neq i}\frac{r_{ij}^a}{|\mathbf{r}_{ij}|}\frac{\partial}{\partial|\mathbf{r}_{ij}|}V(\mathbf{r}_{ij}), \qquad \text{(D.6)}$$

$$\sum_i^N\sum_\alpha^M(r_{i\alpha}^a - R_i^a)\nabla_i^a\cdot\mathbf{A}\left[\nabla_i^b\sum_{j\neq i}V(\mathbf{r}_{ij})\right]\left[\nabla_i^b\sum_{k\neq i}V(\mathbf{r}_{ik})\right] \qquad \text{(D.7)}$$

$$\begin{aligned}
&\frac{d}{dr_i^a}\left\{\left[\sum_{j\neq i}\frac{r_{ij}^b}{|\mathbf{r}_{ij}|}\frac{\partial}{\partial|\mathbf{r}_{ij}|}V(\mathbf{r}_{ij})\right]\left[\sum_{k\neq i}\frac{r_{ik}^b}{|\mathbf{r}_{ik}|}\frac{\partial}{\partial|\mathbf{r}_{ik}|}V(\mathbf{r}_{ik})\right]\right\} = \\
&\left[\sum_{j\neq i}\frac{d}{dr_i^a}\left(\frac{r_{ij}^b}{|\mathbf{r}_{ij}|}\frac{\partial}{\partial|\mathbf{r}_{ij}|}V(\mathbf{r}_{ij})\right)\right]\left[\sum_{k\neq i}\frac{r_{ik}^b}{|\mathbf{r}_{ik}|}\frac{\partial}{\partial|\mathbf{r}_{ik}|}V(\mathbf{r}_{ik})\right] + \\
&\left[\sum_{j\neq i}\frac{r_{ij}^b}{|\mathbf{r}_{ij}|}\frac{\partial}{\partial|\mathbf{r}_{ij}|}V(\mathbf{r}_{ij})\right]\left[\sum_{k\neq i}\frac{d}{dr_i^a}\left(\frac{r_{ik}^b}{|\mathbf{r}_{ik}|}\frac{\partial}{\partial|\mathbf{r}_{ik}|}V(\mathbf{r}_{ik})\right)\right].
\end{aligned} \tag{D.8}$$

In order to continue with equation (D.8) some previous calculations are required

$$\begin{aligned}
\frac{d}{dr_i^a}\left(\frac{r_{ij}^b}{|\mathbf{r}_{ij}|}\frac{dV(\mathbf{r}_{ij})}{d|rij|}\right) &= \frac{\delta_{ab}|\mathbf{r}_{ij}| - r_{ij}^b\frac{r_{ij}^a}{|\mathbf{r}_{ij}|}}{|\mathbf{r}_{ij}|^2}\frac{dV(\mathbf{r}_{ij})}{d|\mathbf{r}_{ij}|} \\
&+ \frac{r_{ij}^b}{|\mathbf{r}_{ij}|}\frac{r_{ij}^a}{|\mathbf{r}_{ij}|}\frac{d^2V(\mathbf{r}_{ij})}{d|\mathbf{r}_{ij}|^2} \\
&= \left(\frac{\delta_{ab}}{|\mathbf{r}_{ij}|} - \frac{r_{ij}^b r_{ij}^a}{|\mathbf{r}_{ij}|^3}\right)\frac{dV(\mathbf{r}_{ij})}{d|\mathbf{r}_{ij}|} \\
&+ \frac{r_{ij}^b r_{ij}^a}{|\mathbf{r}_{ij}|^2}\frac{d^2V(\mathbf{r}_{ij})}{d|\mathbf{r}_{ij}|^2}.
\end{aligned} \tag{D.9}$$

Now using (D.9) we shall express (D.8) as

$$2\sum_{j\neq i}\left(\frac{\delta_{ab}}{|\mathbf{r}_{ij}|} - \frac{r_{ij}^b r_{ij}^a}{|\mathbf{r}_{ij}|^3}\right)\frac{dV(\mathbf{r}_{ij})}{d|\mathbf{r}_{ij}|} + \frac{r_{ij}^b r_{ij}^a}{|\mathbf{r}_{ij}|^2}\frac{d^2V(\mathbf{r}_{ij})}{d|\mathbf{r}_{ij}|^2}\sum_{k\neq i}\frac{r_{ik}^3}{|\mathbf{r}_{ik}|}\frac{dV(\mathbf{r}_{ik})}{d|\mathbf{r}_{ik}}. \tag{D.10}$$

Let us define the following useful tensors for coding

$$T(i)_a^b = \sum_{j\neq i}\left[\left(\frac{\delta_a^b}{|\mathbf{r}_{ij}|} - \frac{r_{ij}^b r_{ij\,a}}{|\mathbf{r}_{ij}|^3}\right)\frac{dV(\mathbf{r}_{ij})}{d|\mathbf{r}_{ij}|} + \frac{r_{ij}^b r_{ij\,a}}{|\mathbf{r}_{ij}|^2}\frac{d^2V(\mathbf{r}_{ij})}{d|\mathbf{r}_{ij}|^2}\right] \tag{D.11}$$

and

$$F(i)_b = \sum_{k\neq i}\frac{r_{ik\,b}}{r_{ik}}\frac{dV(\mathbf{r}_{ij})}{d|\mathbf{r}_{ij}|}. \tag{D.12}$$

Therefore,

$$\frac{d}{dr_i^a}\left\{\left[\sum_{j\neq i}\frac{r_{ij}^b}{|\mathbf{r}_{ij}|}\frac{\partial}{\partial|\mathbf{r}_{ij}|}V(\mathbf{r}_{ij})\right]\left[\sum_{k\neq i}\frac{r_{ik}^b}{|\mathbf{r}_{ik}|}\frac{\partial}{\partial|\mathbf{r}_{ik}|}V(\mathbf{r}_{ik})\right]\right\} = 2T(i)_a^b F(i)_b. \tag{D.13}$$

Finally,

$$\begin{aligned}&\sum_i^N\sum_\alpha^M(r_{i\alpha}^a - R_i^a)\nabla_i^a\cdot\mathbf{A}\left[\nabla_i^b\sum_{j\neq i}V(\mathbf{r}_{ij})\right]\left[\nabla_i^b\sum_{k\neq i}V(\mathbf{r}_{ik})\right] = \\ &2\mathbf{A}\sum_i^N\sum_\alpha^M(r_{i\alpha}^a - R_i^a)T(i)_a^b F(i)_b\end{aligned} \tag{D.14}$$

Thus, the thermodynamic estimator of the kinetic energy with the Li–Broughton action can be written as

$$\begin{aligned}< K_{\mathrm{T}}^{\mathrm{LB}} > &= \frac{3MT}{2N} - \frac{mMT^2}{2\hbar^2N}\sum_{i=1}^N\sum_{\alpha=1}^M(\mathbf{r}_{i\,\alpha+1} - \mathbf{r}_{i\,\alpha})^2 \\ &+ \frac{\hbar^2}{24M^3T^2mN}\sum_{i=1}^N\sum_{\alpha=1}^M\mathbf{F}_{i\alpha}\cdot\mathbf{F}_{i\alpha},\end{aligned} \tag{D.15}$$

and the potential energy

$$\begin{aligned}< V_{\mathrm{T}}^{\mathrm{LB}} > &= \frac{1}{NM}\sum_{\alpha=1}^M\sum_{i<j}V(\mathbf{r}_{ij}^\alpha) \\ &+ \frac{\hbar^2}{12M^3T^2mN}\sum_{i=1}^N\sum_{\alpha=1}^M\mathbf{F}_{i\alpha}\cdot\mathbf{F}_{i\alpha}.\end{aligned} \tag{D.16}$$

Therefore the total thermodynamic energy estimator using a Li–Broughton action is:

$$\begin{aligned}< E_{\mathrm{T}}^{\mathrm{LB}} > &= \frac{3MT}{2N} - \frac{mMT^2}{2\hbar^2N}\sum_{i=1}^N\sum_{\alpha=1}^M(\mathbf{r}_{i\,\alpha+1} - \mathbf{r}_{i\,\alpha})^2 + \frac{1}{NM}\sum_{\alpha=1}^M\sum_{i<j}V(\mathbf{r}_{ij}^\alpha) \\ &+ \frac{\hbar^2}{8M^3T^2mN}\sum_{i=1}^N\sum_{\alpha=1}^M\mathbf{F}_{i\alpha}\cdot\mathbf{F}_{i\alpha}.\end{aligned} \tag{D.17}$$

On the other hand, the virial estimator of the total energy using the Li–Broughton action is:

$$\begin{aligned} < E_{\mathrm{V}}^{\mathrm{LB}} > &= \frac{3}{2}T + \frac{1}{2MN}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}(r_{i\alpha}^{a} - R_{i}^{a})F(i,\alpha)_{a} \\ &+ \frac{1}{NM}\sum_{\alpha=1}^{M}\sum_{i<j}V(\mathbf{r}_{ij}^{\alpha}) + \frac{\hbar^{2}}{8M^{3}T^{2}mN}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}\mathbf{F}_{i\alpha}\cdot\mathbf{F}_{i\alpha} \\ &+ \frac{\hbar^{2}}{24M^{3}NmT^{2}}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}(r_{i\alpha}^{a} - R_{i}^{a})\,T(i,\alpha)_{a}^{b}\,F(i,\alpha)_{a}. \end{aligned} \tag{D.18}$$

It is easily seen how equations (D.17) and (D.18) become the thermodynamic and the virial energy estimators for a primitive action, respectively, if only the first three terms of each equations are kept, yielding

$$\begin{aligned} < E_{\mathrm{T}}^{\mathrm{PA}} > &= \frac{3MT}{2N} - \frac{mMT^{2}}{2\hbar^{2}N}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}(\mathbf{r}_{i\,\alpha+1} - \mathbf{r}_{i\,\alpha})^{2} \\ &+ \frac{1}{NM}\sum_{\alpha=1}^{M}\sum_{i<j}V(\mathbf{r}_{ij}^{\alpha}), \end{aligned} \tag{D.19}$$

and

$$\begin{aligned} < E_{\mathrm{V}}^{\mathrm{PA}} > &= \frac{3}{2}T + \frac{1}{2MN}\sum_{i=1}^{N}\sum_{\alpha=1}^{M}(r_{i\alpha}^{a} - R_{i}^{a})F(i,\alpha)_{a} \\ &+ \frac{1}{NM}\sum_{\alpha=1}^{M}\sum_{i<j}V(\mathbf{r}_{ij}^{\alpha}), \end{aligned} \tag{D.20}$$

where the superscript PA means primitive approximation.

E. APPENDIX

Tricks of the trade

Seasoned simulationists surely will find this appendix plenty of already known information. Notwithstanding, anyone not fully endowed with PIMC simulations could still apply some of the hints here presented.

Many of the tricks about PIMC coding and the simulations involved have already been said throughout the thesis. However, there are still some tricks, we have not said yet, that have proved to be useful for us.

Whenever in this section we refer to something related to a particular choice of coding we did it in the light of the conventions set up in the preambles of Appendix B.

Basic PIMC algorithm

Throughout the thesis we have explained how to code a PIMC program, from the very simple approach of a primitive action along with a bead per bead sampling, until more elaborated recipes that include smart collective displacements of beads and an improved action. We have also included in Appendix B a full bisection code, that with some effort can be converted in an operational program. However, we have not given anywhere the algorithm of the simple recipe. Perhaps, most readers are already acquainted with a simple PIMC code, however, it can be useful for a person just coding his/her first PIMC program, and it helps to illustrate the ideas exposed so far.

The algorithm is written in a mixture of FORTRAN and pseudo–code. All the subroutines called herein have to be programmed. This algorithm is by no means an optimised approach to PIMC, yet it works and helps to ease the approximation to programming.

```
IMPLICIT NONE
CALL INITIALISE
c Initialise the system with a classical simulation
c
DO i = 1, iter
c iter=number of Monte Carlo steps
c
   CALL MOVECM
c  Move centre of mass of all atoms.
c
   DO iatom = 1, natom
c  natom = number of atoms of the system
c
      CALL PICKBEAD
c     Choose a bead at random, namely ibead.
c
      CALL SAFEBEAD
c     Store the position of ibead (x0,y0,z0) in
c     a secure array.
c
      CALL POTENTIAL(V0)
c     Calculates the potential part of the action of the
c     system and returns its value in V0
c
      AV0 = tau * V0
c
      CALL KINETIC(K0)
c     Calculates squared distance between neighbouring
c     beads of iatom. Returns its value in K0
c
      AK0 = T * M * K0 / (4. * lambda)
c     where T = temperature, M = number of beads
c
      A0 = AV0 + AK0
c
      CALL MOVEBEAD
```

```
c     Make a random displacement of ibead from its original
c     position x0,y0,z0 to x1,y1,z1. Use periodic boundary
c     conditions.
c
      CALL POTENTIAL(V1)
      AV1 = tau * V1
      CALL KINETIC(K1)
      AK1 = T * M * K1 / (4. * lambda)
c
      AD = A1 - A0
c
      IF (-AD .GE. 0.d0) THEN
         CONTINUE
      ELSE
         IF (RAN1(IDUM) .LT. EXP(-AD)) THEN
            CONTINUE
         ELSE
            CALL RESTORE
c           Restore the old x0,y0,z0 positions with the
c           secure array.
         END IF
      END IF
      CALL PROPERTIES
c     Calculate properties of the system
c
   END DO
END DO
STOP
END
```

Initialising the simulation

In order to start the simulation the program sets N atoms inside a box. The size of the box depends on the desired density, since the number of atoms is a parameter given by the user. Then a set of classical movements is carried out, that is, every atom is considered as a point particle (just one bead per atom) and the code performs a classical simulation at the requested temperature. That classical process stops once it has reached equilibrium. Since classical movements are much faster than quantum ones, it is far more efficient to reach quantum equilibrium starting from a configuration that is in classical thermal equilibrium than starting from another type of configuration, such as a random distribution of polymer rings.

Before beginning with the PIMC code itself we must set up the initial positions of all beads. We put all the beads belonging to a given atom in the same position occupied by the classical atom obtained in the previous step. That is, the initial length of each polymer ring is zero, which means that the initial kinetic energy exactly amounts to $3/2kT$.

We tried sometimes to distribute the beads of a given atom randomly over a sphere centered on the classical atom with a radius equal to a certain fraction of the thermal wavelength. That process did not result in better efficiencies. Neighbouring beads are strongly correlated via the kinetic springs, and giving for granted a random distribution of beads that has not been Metropolis tested only leads to a delay in reaching equilibrium. Other distributions of beads around the position obtained with the classical simulation, lead to the same delay.

CPU time scheduling

All in all, to reach thermal equilibrium is not the time consuming part of the simulation. Most of the CPU time is spent in reducing the variance of the energy estimators. All other quantities are obtained with a reasonable accuracy in a fraction of the time spent for the energy.

Provided that, if is advisable to distribute the CPU time of a simulation in the following way:

1. Carry out a classical simulation, at the desired temperature, until reaches equilibrium. At this moment you should have $3 * N$ variables, with N the number of atoms. Once this step has been accomplished make as many copies of each classical atom as beads are required per atom, yielding a figure of $3 * N * M$ variables, with M the number of beads per atom. This simulation should last a negligible fraction of the whole CPU time that will be spent.

2. Perform a PIMC simulation that does not estimate any quantity. Its only goal is to equilibrate the positions of the polymer rings. This step should take about 15% of the total simulation time.

3. Continue the PIMC simulation started above estimating all the required quantities but the energy. Employ 15% of the time doing so.

4. Keep going with the simulation, but now your code should be only estimating the energy with the virial estimator. This step consumes the remaining 70% of the CPU time. Nevertheless, in the meantime, you will be studying the results obtained in the third step. If you calculated the momentum distribution and the pair distribution function, you will have a good estimation of the kinetic and the potential energy that can be compared with the results obtained in this step.

The timings given above should be taken as a thumb rule, and your final choice will depend mainly on the accuracy desired.

The question of how long does it take a PIMC simulation is more difficult to answer. It depends on many parameters, such as, the computer you are using, the temperature you are simulating, which is inversely proportional to the number of beads, or how optimised is your code. Our simulations lasted from one day to a couple of months.

For example, if you want to be confident with the first decimal figure of the energy of Ne at 25 K, a couple of days will suffice using a Pentium 800 MHz processor. However, if you want to be confident with the second decimal the simulation time will stretch out to almost a month.

Moving the centre of mass

Once the algorithm has passed over all the atoms moving either a single bead of them, in a bead per bead sampling, or a segment of polymer ring using bisection or staging, the centre of mass of every atom must be moved. That is, every polymer ring must be displaced using the Metropolis test keeping the inter–bead distances inside the atom unaltered. This step assures a fast convergence of the algorithm. A PIMC algorithm that does not includes a movement of the centre of mass is doomed to fail.

In preparation for moving the centre of mass, the potential contribution to the action of the whole system is calculated. Then, all the beads of a given atom are displaced preserving the distances amongst the beads of that atom, using

$$r^{\mathrm{new}}_{\mathrm{x,y,z}} = r^{\mathrm{old}}_{\mathrm{x,y,z}} + (2\delta_{\mathrm{x,y,z}} - 1)\Delta_{\mathrm{CM}}, \tag{E.1}$$

with $\delta_{\mathrm{x,y,z}}$ three uniformly random distributed numbers and Δ_{CM} a parameter that is adjusted prior the simulation starts in order to get around a fifty percent acceptance of the centre of mass proposed displacements. Periodic boundary conditions must be applied to the new proposed positions.

The new potential contribution to the action is calculated. The proposed movement of all beads of all atoms is accepted or rejected as a whole depending on a Metropolis scheme. Notice that only the potential contribution to the action amounts for the Metropolis decision, since there is no change in the kinetic contribution because of the inter–bead distances have not changed. For that reason the energy estimator is not calculated at this step, since it would yield a biased result because of the always repetitive contribution of the kinetic energy estimator.

Should you be using a bead per bead sampling, use a different parameter for the displacement of each bead. The differences between the parameter related to bead displacements and the one in charge of the centre of mass are about an order of magnitude.

Optimising the action

It is a common practice in any PIMC program not to calculate the whole action due to all the interactions in every single step. Instead, what is normally done is to do this whole calculation just once. After that, it is only required to calculate the changes introduced in the action due to the accepted displacements. However, when writing a code for the first time, it is advisable to calculate the whole action in every single step during the debugging process. Once you are confident with your code, optimise it calculating only the changes in action as it has been described above.

Check that you still get exactly the same results provided the same initial conditions and the same seed for the random number generator is used.

Another way to obtain better benchmarks is to create a table of potentials for any given distance within the simulation box. We found an improvement of 10% of CPU time when using the table of an Aziz Ne–Ne (Aziz and Slaman, 1989). The same improvement was found for the Aziz ^{4}He potential (Aziz et al., 1997). When using tables it has been a common practice to interpolate between the points of the table in order to improve accuracy. However, we found that using tables of 100,000 or more there is no necessity of an interpolation. Nowadays, desktop computers with 256 MB of RAM can store these huge arrays, and easily manage them (due to improved memory buses). The CPU time employed in a simulation will not significantly change due to a table that size, and on the other hand you shall avoid time consuming interpolations and you shall get better accuracy.

Periodic boundary conditions

Periodic boundary conditions have to be taken with extreme care in a PIMC code. In a classical simulation when an atom leaves one side of the box, enters into it on the opposite side (Allen and Tildesley, 1997). Here the situation is more complicated. An atom is represented by a chain, therefore, in order to say that the atom has exited the box, the whole polymer ring has to be outside of the box. If you apply periodic boundary conditions to beads in the same way as if they were atoms in a classical simulation, you will end up with a piece of the polymer ring in one side and the complementing piece in the opposite one. Equations (4.33) to (4.36) are the ones to be used in order to apply periodic boundary conditions in a bisection sampling rightly.

A symptom of an error in the periodic boundary conditions is to obtain a sequence of plausible energies, that fit the results we are expecting, and once in a while an extremely large value of the energy, that immediately goes down to the normal level. This burst of the energy is the result of evaluating the kinetic energy $(\mathbf{r}_{i\alpha} - (\mathbf{r}_{i\alpha+1})^2$ between two neighbouring beads when one has exited the box and has been send to the opposite side. The figure obtained is of the order of L^2, with L the size of the simulation box.

FORTRAN *oriented tips*

Knowing some insights of FORTRAN can help to improve the benchmarks. FORTRAN accesses any array going by columns, contrary to the normal human way of reading arrays by rows. If one forces the code to read the array row by row, instead of column by column, the running will become slower. The computer would load a whole column for just reading an element of it, then will load the next column and will just read another element, and so on. We have observed that forcing the computer to do so delayed our timings in about 15%. Of course, the delay will be proportional to the size of the arrays involved in the simulation.

The size of the compiled executable had no noticeable influence on the CPU times registered. Therefore, we recommend to use arrays as large as necessary. Our codes were written in FORTRAN 77 which only has fixed array sizes. Since the compiler knows beforehand the size of every array it manages them in an optimal way to fit them into memory. For this reason we discourage the use of allocatable arrays available in FORTRAN 90 and FORTRAN 95, unless for memory reasons they are unavoidable. Allocatable arrays are dynamical arrays that change their size during the simulation freeing memory space when it is not required. They are the only solution when memory problems are an issue, however, they do not give the compiler the opportunity to assign them the optimal position within memory before running. PIMC simulations are extremely CPU demanding, but their requirements of memory are more than sufficiently coped with current computers.

BIBLIOGRAPHY

Allen, M. P. and Tildesley, D. J. (1997). *Computer simulation of liquids.* Oxford University Press.

Aziz, R. A., McCourt, F. R. W., and Wong, C. C. K. (1997). A new determination of the ground state interatomic potential for He_2. *Mol. Phys.*, 61:1487.

Aziz, R. A. and Slaman, M. J. (1989). The Ne-Ne interatomic potential revisited. *Chem. Phys.*, 130:187.

Azuah, R. T., Stirling, W. G., Glyde, H. R., and Boninsegni, M. (1997). Momentum distribution and final state effects in liquid neon. *J. Low Temp. Phys*, 109:287.

Barker, J. A. (1979). A quantum–statistical Monte Carlo method; path integrals with boundary conditions. *J. Chem. Phys.*, 70:2914.

Binder, K., editor (1979). *Monte Carlo methods in statistical physics*, volume 7 of *Topics in current physics.* Springer–Verlag.

Binder, K., editor (1994). *Monte Carlo and molecular dynamics simulations in polymer science.* Clarendon Press, Oxford.

Bohm, M. C., Schulte, J., and Ramirez, R. (2002). Excited state properties of C6H6 and C6D6 studied by Feymnan path integral–ab initio simulations. *J. Phys. Chem. A*, 106:3169.

Boronat, J. (2002). *Quantum Liquids in Confined Geometries*, chapter Monte Carlo simulations at zero temperature: one, two, and three dimensions. World Scientific, Singapore.

Boronat, J. and Casulleras, J. (1994). Monte Carlo analysis of an interatomic potential for He. *Phys. Rev. B*, 49:8920.

Brualla, L., Boronat, J., and Casulleras, J. (2002). Momentum distribution of quantum liquids at finite temperature. *J. Low Temp. Phys.*, 126(5/6):1547.

Burghardt, B., Eicke, J., and Stolze, J. (1998). Evaluation of coherent–state path integrals in statistical mechanics by matrix multiplication. *J. Chem. Phys.*, 108:1562.

Cao, J. and Berne, B. J. (1993). A Born–Oppenheimer approximation for path integrals with an application to electron solvation in polarizable fluids. *J. Chem. Phys.*, 99:2902.

Celli, M., Zoppi, M., and Mayers, J. (1998). Kinetic energy of ^{4}He along the $T = 6.1$K isotherm. *Phys. Rev. B*, 58:242.

Ceperley, D. M. (1995a). Path integrals in the theory of condensed helium. *Rev. Mod. Phys.*, 67(2):279.

Ceperley, D. M. (1995b). Path integrals in the theory of condensed helium. *Rev. Mod. Phys.*, 67:279.

Ceperley, D. M. and Pollock, E. L. (1986). Path–integral computation fo the low–temperature properties of liquid ^{4}He. *Phys. Rev. Lett.*, 56:351.

Ceperley, D. M. and Pollock, E. L. (1992). Path–integral computation techniques for superfluid ^{4}He. In Caracciolo, S. and Fabrocini, A., editors, *Monte Carlo methods in theoretical physics*, page 35. ETS Editrice, Pisa.

Ceperley, D. M., Simmons, R. O., and Blasdell, R. C. (1996). The kinetic energy of liquid and solid ^{4}He. *Phys. Rev. Lett.*, 77:115.

Chakravarty, C., Gordillo, M. C., and Ceperley, D. M. (1998). A comparison of the efficiency of Fourier– and discrete time–path integral Monte Carlo. *J. Chem. Phys.*, 109:2123.

Chandler, D. and Wolynes, P. G. (1981). Exploiting the isomorphism between quantum theory and classical statistical mechanics of polyatomic fluids. *J. Chem. Phys.*, 74:4078.

Chin, S. A. (1997). Symplectic integrators from composite operator factorizations. *Phys. Lett. A*, 226:344.

Crawford, R. K. (1977). *Rare gas solids*, volume II, chapter Melting, vaporization and sublimation. Academic Press, New York.

D. A. Peek, M. C. S., Fujita, I., and Simmons, R. O. (1992). Single–particle kinetic energies in liquid neon. *Phys. Rev. B*, 45:9671.

Feynman, R. P. (1948). Space–time approach to non–relativistic quantum mechanics. *Rev. Mod. Phys.*, 20:367.

Feynman, R. P. (1998). *Statistical mechanics. A set of lectures.* Addison–Wesley.

Feynman, R. P. and Hibbs, A. R. (1965a). *Quantum Mechanics and Path Integrals.* McGraw–Hill, Inc., New York.

Feynman, R. P. and Hibbs, A. R. (1965b). *Quantum mechanics and path integrals.* McGraw–Hill Inc.

Filinov, V. S., Bonitz, M., Ebeling, W., and Fortov, V. E. (2001). Thermodynamics of hot dense H–plasmas: path integral Monte Carlo simulations and analytical approximations. *Plasma Phys. Controlled Fusion*, 43:743.

Fosdick, L. D. and Jordan, H. F. (1965). Path–integral calculation of the two–particle Slater sum for He^4. *Phys. Rev.*, 143:58.

Freeman, D. L. and Doll, J. D. (1984). A Monte Carlo method for quantum Boltzmann statistical mechanics using Fourier representations of path integrals. *J. Chem. Phys.*, 80:5709.

Frenkel, D. and Smit, B. (1996). *Understanding molecular simulation. From algorithms to applications.* Academic Press.

Fugate, R. Q. and Swenson, C. A. (1973). Equation of state for solid neon to 20 Kbar. *J. Low Temp. Phys.*, 10:317.

Fye, R. M. (1986). New results on Trotter–like approximations. *Phys. Rev. B*, 33:6271.

Gillan, M. J. (1990). The path–integral simulation of quantum systems. In Catlow, C. R. A., Parker, S. C., and Allen, M. P., editors, *Computer Modelling of Fluids Polymers and Solids*, page 155. Kluwer Academic Publishers.

Glaesemann, K. R. and Fried, L. E. (2002). An improved thermodynamic energy estimator for path integral simulations. *J. Chem. Phys.*, 116:5951.

Glyde, H. R. (1994). *Excitations in liquid and solid Helium.* Clarendon, Oxford.

Glyde, H. R., Azuah, R. T., Andersen, K. H., and Stirling, W. G. (1995). Single particle dynamics in quantum fluids observed by neutron scattering. In Casas, M., Llano, M. D., Navarro, J., and Polls, A., editors, *Condensed matter theories*, volume 10. Nova Science Publishers, New York.

Goodman, J. and Sokal, A. D. (1986). Multigrid Monte Carlo method for lattice field theories. *Phys. Rev. Lett.*, 56:1015.

Gould, H. and Tobochnik, J. (1996). *An introduction to computer simulation methods. Applications to physical systems*, chapter 18.8. Addison–Wesley, second edition.

Guardiola, R. (1998). *Microscopic quantum many–body theories and their applications*, chapter Monte Carlo methods in quantum many–body theories, page 269. Springer, Berlin.

Guardiola, R., Higón, E., and Ros, J. (1995). *Mètodes numèrics per a la física.* Universitat de València.

Herman, M. F., Bruskin, E. J., and Berne, B. J. (1982). On path integral Monte Carlo simulations. *J. Chem. Phys.*, 76:5150.

Jang, S., Jang, S., and Voth, G. A. (2001). Applications of higher order composite factorization schemes in imaginary time path integral simulations. *J. Chem. Phys.*, 115:7832.

Janke, W. and Sauer, T. (1993). Path integral Monte Carlo using multigrid techniques. *Chem. Phys. Lett.*, 201(5,6):499.

Janke, W. and Sauer, T. (1996). Multigrid method versus staging algorithm for PIMC simulations. *Chem. Phys. Lett.*, 263:488.

Janke, W. and Sauer, T. (1997). Optimal energy estimation in path–integral Monte Carlo simulations. *J. Chem. Phys.*, 107:5821.

Kalos M. H. and Whitlock P. A. (1986). *Monte Carlo methods*, volume I: Basics. John Wiley & Sons, Inc.

Krauth, W. (1998). Introduction to Monte Carlo algorithms. In Kertesz, J. and Kondor, I., editors, *Advances in computer simulation.* Springer Verlag.

Lévy, P. (1939). Sur certains processus stochastiques homogènes. *Compositio Math.*, 7:283.

Li, X. P. and Broughton, J. Q. (1987). High–order correction to the Trotter expansion for use in computer simulation. *J. Chem. Phys.*, 86:5094.

Lippold, J. (1969). Isothermal compressibility and density of liquid argon and neon up to pressures of 1000 kg/cm^2. *Cryogenics*, 9:112.

Mazzanti, F., Boronat, J., and Polls, A. (1996). Finale-state effects on superfluid ^{4}He in the deep inelastic region. *Phys. Rev. B*, 53:5661.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087.

Nightingale, M. P. (1999). Basics, quantum Monte Carlo and statistical mechanics. In Nightingale, M. P. and Umrigar, C. J., editors, *Quantum Monte Carlo methods in physics and chemistry*, volume 525 of *NATO Science Series*, page 1. NATO, Kluwer Academic Publishers.

Peek, D. A., Fujita, I., Schmidt, M. C., and Simmons, R. O. (1992). Single–particle kinetic energies in solid neon. *Phys. Rev. B*, 45:9680.

Pollock, E. L. and Ceperley, D. M. (1984). Simulation of quantum many–body systems by path–integral methods. *Phys. Rev. B*, 30:2555.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1999). *Numerical Recipes in* FORTRAN*. The art of scientific computing*, chapter 16.4. Cambridge University Press, second edition.

Raedt, H. D. and Raedt, B. D. (1983). Applications of the generalized Trotter formula. *Phys. Rev. A*, 28:3575.

Runge, K. J. and Chester, G. V. (1988). Solid–fluid phase transition of quantum hard spheres at finite temperatures. *Phys. Rev. B*, 38:135.

Sears, V. F. (1969). High–energy neutron scattering from liquid ^{4}He. *Phys. Rev.*, 185:200.

Silver, R. N. and Sokol, P. E., editors (1989). *Momentum distributions.* Plenum, New York.

Singer, K. and Smith, W. (1988). Path integral simulations of condensed phase lennard–jones systems. *Mol. Phys.*, 64:1215.

Sosnick, T. R., Snow, W. M., Sokol, P. E., and Silver, R. N. (1989). Momentum distributions in liquid ^{4}He. *Europhys. Lett.*, 9:707.

Sprik, M., Klein, M. L., and Chandler, D. (1985). Staging: a sampling technique for the Monte Carlo evaluation of path integrals. *Phys. Rev. B*, 31:4234.

Takahashi, M. and Imada, M. (1984). Monte Carlo calculation of quantum systems ii. Higher order correction. *J. Phys. Soc. Jpn.*, 53:3765.

Trotter, E. (1958). On the product of semi–groups of operators. *Proc. Am. Math. Soc.*, 10:545.

Tuckerman, M. E., Berne, B. J., Martyna, G. J., and Klein, M. L. (1993). Efficient molecular dynamics and hybrid Monte Carlo algorithms for path integrals. *J. Chem. Phys.*, 99:2796.

Voth, G. A. (1996). Path–integral centroid methods in quantum statistical mechanics and dynamics. *Adv. Chem. Phys.*, XCIII:135.

Weht, R. O., Kohanoff, J., Estrin, D. A., and Chakravarty, C. (1998). An *ab initio* path integral Monte Carlo simulation method for molecules and clusters: Application to Li_4 and Li_5^+. *J. Chem. Phys.*, 108:8848.

Woods, A. D. B. and Sears, V. F. (1977). Momentum distribution in liquid ^{4}He at $T = 1.1$ and 4.2K. *Phys. rev. Lett.*, 39:415.