

Chapter 1

Introduction

1.1 General

For **physics students** the computational quantum physics courses is a recommended prerequisite for any computationally oriented semester thesis, proseminar, master thesis or doctoral thesis.

For **computational science and engineering (RW) students** the computational quantum physics courses is part of the “Vertiefung” in theoretical physics.

1.1.1 Exercises

Programming Languages

Except when a specific programming language or tool is explicitly requested you are free to choose any programming language you like. Solutions will often be given either as C++ programs or Mathematica Notebooks.

Computer Access

The lecture rooms offer both Linux workstations, for which accounts can be requested with the computer support group of the physics department in the HPT building, as well as connections for your notebook computers.

1.1.2 Prerequisites

As a prerequisite for this course we expect knowledge of the following topics. Please contact us if you have any doubts or questions.

Computing

- Basic knowledge of UNIX
- At least one procedural programming language such as C, C++, Pascal, Java or FORTRAN. C++ knowledge is preferred.
- Knowledge of a symbolic mathematics program such as Mathematica or Maple.
- Ability to produce graphical plots.

Numerical Analysis

- Numerical integration and differentiation
- Linear solvers and eigensolvers
- Root solvers and optimization
- Statistical analysis

Quantum Mechanics

Basic knowledge of quantum mechanics, at the level of the quantum mechanics taught to computational scientists, should be sufficient to follow the course. If you feel lost at any point, please ask the lecturer to explain whatever you do not understand. We want you to be able to follow this course without taking an advanced quantum mechanics class.

1.1.3 References

1. J.M. Thijssen, *Computational Physics*, Cambridge University Press (1999) ISBN 0521575885
2. Nicholas J. Giordano, *Computational Physics*, Pearson Education (1996) ISBN 0133677230.
3. Harvey Gould and Jan Tobochnik, *An Introduction to Computer Simulation Methods*, 2nd edition, Addison Wesley (1996), ISBN 00201506041
4. Tao Pang, *An Introduction to Computational Physics*, Cambridge University Press (1997) ISBN 0521485924

1.2 Overview

In this class we will learn how to simulate quantum systems, starting from the simple one-dimensional Schrödinger equation to simulations of interacting quantum many body problems in condensed matter physics and in quantum field theories. In particular we will study

- The one-body Schrödinger equation and its numerical solution
- The many-body Schrödinger equation and second quantization
- Approximate solutions to the many body Schrödinger equation
- Path integrals and quantum Monte Carlo simulations
- Numerically exact solutions to (some) many body quantum problems
- Some simple quantum field theories

Chapter 2

Quantum mechanics in one hour

2.1 Introduction

The purpose of this chapter is to refresh your knowledge of quantum mechanics and to establish notation. Depending on your background you might not be familiar with all the material presented here. If that is the case, please ask the lecturers and we will expand the introduction. Those students who are familiar with advanced quantum mechanics are asked to glance over some omissions.

2.2 Basis of quantum mechanics

2.2.1 Wave functions and Hilbert spaces

Quantum mechanics is nothing but simple linear algebra, albeit in huge Hilbert spaces, which makes the problem hard. The foundations are pretty simple though.

A pure state of a quantum system is described by a “wave function” $|\Psi\rangle$, which is an element of a Hilbert space \mathcal{H} :

$$|\Psi\rangle \in \mathcal{H} \quad (2.1)$$

Usually the wave functions are normalized:

$$\| |\Psi\rangle \| = \sqrt{\langle \Psi | \Psi \rangle} = 1. \quad (2.2)$$

Here the “bra-ket” notation

$$\langle \Phi | \Psi \rangle \quad (2.3)$$

denotes the scalar product of the two wave functions $|\Phi\rangle$ and $|\Psi\rangle$.

The simplest example is the spin-1/2 system, describing e.g. the two spin states of an electron. Classically the spin \vec{S} of the electron (which can be visualized as an internal angular momentum), can point in any direction. In quantum mechanics it is described by a two-dimensional complex Hilbert space $\mathcal{H} = \mathbb{C}^2$. A common choice of

basis vectors are the “up” and “down” spin states

$$|\uparrow\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (2.4)$$

$$|\downarrow\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (2.5)$$

This is similar to the classical Ising model, but in contrast to a classical Ising spin that can point only either up or down, the quantum spin can exist in any complex superposition

$$|\Psi\rangle = \alpha|\uparrow\rangle + \beta|\downarrow\rangle \quad (2.6)$$

of the basis states, where the normalization condition (2.2) requires that $|\alpha|^2 + |\beta|^2 = 1$.

For example, as we will see below the state

$$|\rightarrow\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle + |\downarrow\rangle) \quad (2.7)$$

is a superposition that describes the spin pointing in the positive x -direction.

2.2.2 Mixed states and density matrices

Unless specifically prepared in a pure state in an experiment, quantum systems in Nature rarely exist as pure states but instead as probabilistic superpositions. The most general state of a quantum system is then described as a density matrix ρ , with unit trace

$$\text{Tr}\rho = 1. \quad (2.8)$$

The density matrix of a pure state is just the projector onto that state

$$\rho_{\text{pure}} = |\Psi\rangle\langle\Psi|. \quad (2.9)$$

For example, the density matrix of a spin pointing in the positive x -direction is

$$\rho_{\rightarrow} = |\rightarrow\rangle\langle\rightarrow| = \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix}. \quad (2.10)$$

Instead of being in a coherent superposition of up and down, the system could also be in a probabilistic mixed state, with a 50% probability of pointing up and a 50% probability of pointing down, which would be described by the density matrix

$$\rho_{\text{mixed}} = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}. \quad (2.11)$$

2.2.3 Observables

Any physical observable is represented by a self-adjoint linear operator acting on the Hilbert space, which in a final dimensional Hilbert space can be represented by a Hermitian matrix. For our spin-1/2 system, using the basis introduced above, the components

S^x , S^y and S^z of the spin in the x -, y -, and z -directions are represented by the Pauli matrices

$$S^x = \frac{\hbar}{2} \sigma_x = \frac{\hbar}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (2.12)$$

$$S^y = \frac{\hbar}{2} \sigma_y = \frac{\hbar}{2} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad (2.13)$$

$$S^z = \frac{\hbar}{2} \sigma_z = \frac{\hbar}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (2.14)$$

The spin component along an arbitrary unit vector \hat{e} is the linear superposition of the components, i.e.

$$\hat{e} \cdot \vec{S} = e^x S^x + e^y S^y + e^z S^z = \frac{\hbar}{2} \begin{pmatrix} e^z & e^x - ie^y \\ e^x + ie^y & -e^z \end{pmatrix} \quad (2.15)$$

The fact that these observables do not commute but instead satisfy the non-trivial commutation relations

$$[S^x, S^y] = S^x S^y - S^y S^x = i\hbar S^z, \quad (2.16)$$

$$[S^y, S^z] = i\hbar S^x, \quad (2.17)$$

$$[S^z, S^x] = i\hbar S^y, \quad (2.18)$$

is the root of the differences between classical and quantum mechanics .

2.2.4 The measurement process

The outcome of a measurement in a quantum system is usually intrusive and not deterministic. After measuring an observable A , the new wave function of the system will be an eigenvector of A and the outcome of the measurement the corresponding eigenvalue. The state of the system is thus changed by the measurement process!

For example, if we start with a spin pointing up with wave function

$$|\Psi\rangle = |\uparrow\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (2.19)$$

or alternatively density matrix

$$\rho_{\uparrow} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad (2.20)$$

and we measure the x -component of the spin S^x , the resulting measurement will be either $+\hbar/2$ or $-\hbar/2$, depending on whether the spin after the measurement points in the $+$ or $-$ x -direction, and the wave function after the measurement will be either of

$$|\rightarrow\rangle = \frac{1}{\sqrt{2}} (|\uparrow\rangle + |\downarrow\rangle) = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \quad (2.21)$$

$$|\leftarrow\rangle = \frac{1}{\sqrt{2}} (|\uparrow\rangle - |\downarrow\rangle) = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix} \quad (2.22)$$

Either of these states will be picked with a probability given by the overlap of the initial wave function by the individual eigenstates:

$$p_{\rightarrow} = ||\langle \rightarrow | \Psi \rangle||^2 = 1/2 \quad (2.23)$$

$$p_{\leftarrow} = ||\langle \leftarrow | \Psi \rangle||^2 = 1/2 \quad (2.24)$$

The final state is a probabilistic superposition of these two outcomes, described by the density matrix

$$\rho = p_{\rightarrow} |\rightarrow\rangle\langle\rightarrow| + p_{\leftarrow} |\leftarrow\rangle\langle\leftarrow| = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}. \quad (2.25)$$

which differs from the initial density matrix ρ_{\uparrow} .

If we are not interested in the result of a particular outcome, but just in the average, the expectation value of the measurement can easily be calculated from a wave function $|\Psi\rangle$ as

$$\langle A \rangle = \langle \Psi | A | \Psi \rangle \quad (2.26)$$

or from a density matrix ρ as

$$\langle A \rangle = \text{Tr}(\rho A). \quad (2.27)$$

For pure states with density matrix $\rho_{\Psi} = |\Psi\rangle\langle\Psi|$ the two formulations are identical:

$$\text{Tr}(\rho_0 A) = \text{Tr}(|\Psi\rangle\langle\Psi| A) = \langle \Psi | A | \Psi \rangle. \quad (2.28)$$

2.2.5 The uncertainty relation

If two observables A and B do not commute $[A, B] \neq 0$, they cannot be measured simultaneously. If A is measured first, the wave function is changed to an eigenstate of A , which changes the result of a subsequent measurement of B . As a consequence the values of A and B in a state Ψ cannot be simultaneously known, which is quantified by the famous Heisenberg uncertainty relation which states that if two observables A and B do not commute but satisfy

$$[A, B] = i\hbar \quad (2.29)$$

then the product of the root-mean-square deviations ΔA and ΔB of simultaneous measurements of A and B has to be larger than

$$\Delta A \Delta B \geq \hbar/2 \quad (2.30)$$

For more details about the uncertainty relation, the measurement process or the interpretation of quantum mechanics we refer interested students to an advanced quantum mechanics class or text book.

2.2.6 The Schrödinger equation

The time-dependent Schrödinger equation

After so much introduction the Schrödinger equation is very easy to present. The wave function $|\Psi\rangle$ of a quantum system evolves according to

$$i\hbar \frac{\partial}{\partial t} |\Psi(t)\rangle = H |\Psi(t)\rangle, \quad (2.31)$$

where H is the Hamilton operator. This is just a first order linear differential equation.

The time-independent Schrödinger equation

For a stationary time-independent problem the Schrödinger equation can be simplified. Using the ansatz

$$|\Psi(t)\rangle = \exp(-iEt/\hbar)|\Psi\rangle, \quad (2.32)$$

where E is the energy of the system, the Schrödinger equation simplifies to a linear eigenvalue problem

$$H|\Psi\rangle = E|\Psi\rangle. \quad (2.33)$$

The rest of the semester will be spent solving just this simple eigenvalue problem!

The Schrödinger equation for the density matrix

The time evolution of a density matrix $\rho(t)$ can be derived from the time evolution of pure states, and can be written as



$$i\hbar \frac{\partial}{\partial t} \rho(t) = [H, \rho(t)] \quad (2.34)$$

The proof is left as a simple exercise.

2.2.7 The thermal density matrix

Finally we want to describe a physical system not in the ground state but in thermal equilibrium at a given inverse temperature $\beta = 1/k_B T$. In a classical system each microstate i of energy E_i is occupied with a probability given by the Boltzmann distribution

$$p_i = \frac{1}{Z} \exp(-\beta E_i), \quad (2.35)$$

where the partition function

$$Z = \sum_i \exp(-\beta E_i) \quad (2.36)$$

normalizes the probabilities.

In a quantum system, if we use a basis of eigenstates $|i\rangle$ with energy E_i , the density matrix can be written analogously as

$$\rho_\beta = \frac{1}{Z} \sum_i \exp(-\beta E_i) |i\rangle \langle i| \quad (2.37)$$

For a general basis, which is not necessarily an eigenbasis of the Hamiltonian H , the density matrix can be obtained by diagonalizing the Hamiltonian, using above equation, and transforming back to the original basis. The resulting density matrix is

$$\rho_\beta = \frac{1}{Z} \exp(-\beta H) \quad (2.38)$$

where the partition function now is

$$Z = \text{Tr} \exp(-\beta H) \quad (2.39)$$

Calculating the thermal average of an observable A in a quantum system is hence formally very easy:

$$\langle A \rangle = \text{Tr}(A\rho_\beta) = \frac{\text{Tr}A \exp(-\beta H)}{\text{Tr} \exp(-\beta H)}, \quad (2.40)$$

but actually evaluating this expression is a hard problem.

2.3 The spin- S problem

Before discussing solutions of the Schrödinger equation we will review two very simple systems: a localized particle with general spin S and a free quantum particle.

In section 2.2.1 we have already seen the Hilbert space and the spin operators for the most common case of a spin-1/2 particle. The algebra of the spin operators given by the commutation relations (2.12)-(2.12) allows not only the two-dimensional representation shown there, but a series of $2S + 1$ -dimensional representations in the Hilbert space \mathbb{C}^{2S+1} for all integer and half-integer values $S = 0, \frac{1}{2}, 1, \frac{3}{2}, 2, \dots$. The basis states $\{|s\rangle\}$ are usually chosen as eigenstates of the S^z operator

$$S^z|s\rangle = \hbar s|s\rangle, \quad (2.41)$$

where s can take any value in the range $-S, -S + 1, -S + 2, \dots, S - 1, S$. In this basis the S_z operator is diagonal, and the S^x and S^y operators can be constructed from the “ladder operators”

$$S^+|s\rangle = \sqrt{S(S+1) - s(s+1)}|s+1\rangle \quad (2.42)$$

$$S^-|s\rangle = \sqrt{S(S+1) - s(s-1)}|s-1\rangle \quad (2.43)$$

which increment or decrement the S^z value by 1 through

$$S^x = \frac{1}{2}(S^+ + S^-) \quad (2.44)$$

$$S^y = \frac{1}{2i}(S^+ - S^-). \quad (2.45)$$

The Hamiltonian of the spin coupled to a magnetic field \vec{h} is

$$H = -g\mu_B \vec{h} \cdot \vec{S}, \quad (2.46)$$

which introduces nontrivial dynamics since the components of \vec{S} do not commute. As a consequence the spin precesses around the magnetic field direction.

Exercise: Derive the differential equation governing the rotation of a spin starting along the $+x$ -direction rotating under a field in the $+z$ -direction

2.4 A quantum particle in free space

Our second example is a single quantum particle in an n -dimensional free space. Its Hilbert space is given by all twice-continuously differentiable complex functions over

the real space \mathbb{R}^n . The wave functions $|\Psi\rangle$ are complex-valued functions $\Psi(\vec{x})$ in n -dimensional space. In this representation the operator \hat{x} , measuring the position of the particle is simple and diagonal

$$\hat{x} = \vec{x}, \quad (2.47)$$

while the momentum operator \hat{p} becomes a differential operator

$$\hat{p} = -i\hbar\nabla. \quad (2.48)$$

These two operators do not commute but their commutator is

$$[\hat{x}, \hat{p}] = i\hbar. \quad (2.49)$$

The Schrödinger equation of a quantum particle in an external potential $V(\vec{x})$ can be obtained from the classical Hamilton function by replacing the momentum and position variables by the operators above. Instead of the classical Hamilton function

$$H(\vec{x}, \vec{p}) = \frac{\vec{p}^2}{2m} + V(\vec{x}) \quad (2.50)$$

we use the quantum mechanical Hamiltonian operator

$$H = \frac{\hat{p}^2}{2m} + V(\hat{x}) = -\frac{\hbar^2}{2m}\nabla^2 + V(\vec{x}), \quad (2.51)$$

which gives the famous form

$$i\hbar\frac{\partial\psi}{\partial t} = -\frac{\hbar^2}{2m}\nabla^2\psi + V(\vec{x})\psi \quad (2.52)$$

of the one-body Schrödinger equation.

2.4.1 The harmonic oscillator

As a special exactly solvable case let us consider the one-dimensional quantum harmonic oscillator with mass m and potential $\frac{K}{2}x^2$. Defining momentum \hat{p} and position operators \hat{q} in units where $m = \hbar = K = 1$, the time-independent Schrödinger equation is given by

$$H|n\rangle = \frac{1}{2}(\hat{p}^2 + \hat{q}^2)|n\rangle = E_n|n\rangle \quad (2.53)$$

Inserting the definition of \hat{p} we obtain an eigenvalue problem of an ordinary differential equation

$$-\frac{1}{2}\phi_n''(q) + \frac{1}{2}q^2\phi_n(q) = E_n\phi_n(q) \quad (2.54)$$

whose eigenvalues $E_n = (n + 1/2)$ and eigenfunctions

$$\phi_n(q) = \frac{1}{\sqrt{2^n n! \sqrt{\pi}}} \exp\left(-\frac{1}{2}q^2\right) H_n(q), \quad (2.55)$$

are known analytically. Here the H_n are the Hermite polynomials and $n = 0, 1, \dots$

Using these eigenstates as a basis sets we need to find the representation of \hat{q} and \hat{p} . Performing the integrals

$$\langle m|\hat{q}|n\rangle \quad \text{and} \quad \langle m|\hat{p}|n\rangle \quad (2.56)$$

it turns out that they are nonzero only for $m = n \pm 1$ and they can be written in terms of “ladder operators” a and a^\dagger :

$$\hat{q} = \frac{1}{\sqrt{2}}(a^\dagger + a) \quad (2.57)$$

$$\hat{p} = \frac{1}{i\sqrt{2}}(a^\dagger - a) \quad (2.58)$$

$$(2.59)$$

where the raising and lowering operators a^\dagger and a only have the following nonzero matrix elements:

$$\langle n+1|a^\dagger|n\rangle = \langle n|a|n+1\rangle = \sqrt{n+1}. \quad (2.60)$$

and commutation relations

$$[a, a] = [a^\dagger, a^\dagger] = 0 \quad (2.61)$$

$$[a, a^\dagger] = 1. \quad (2.62)$$

It will also be useful to introduce the number operator $\hat{n} = a^\dagger a$ which is diagonal with eigenvalue n : elements

$$\hat{n}|n\rangle = a^\dagger a|n\rangle = \sqrt{n}a^\dagger|n-1\rangle = n|n\rangle. \quad (2.63)$$

To check this representation let us plug the definitions back into the Hamiltonian to obtain

$$\begin{aligned} H &= \frac{1}{2}(\hat{p}^2 + \hat{q}^2) \\ &= \frac{1}{4}[-(a^\dagger - a)^2 + (a^\dagger + a)^2] \\ &= \frac{1}{2}(a^\dagger a + a a^\dagger) \\ &= \frac{1}{2}(2a^\dagger a + 1) = \hat{n} + \frac{1}{2}, \end{aligned} \quad (2.64)$$

which has the correct spectrum. In deriving the last lines we have used the commutation relation (2.62).

Chapter 3

The quantum one-body problem

3.1 The time-independent 1D Schrödinger equation

We start the numerical solution of quantum problems with the time-indepent one-dimensional Schrödinger equation for a particle with mass m in a Potential $V(x)$. In one dimension the Schrödinger equation is just an ordinary differential equation

$$-\frac{\hbar^2}{2m} \frac{\partial^2 \psi}{\partial x^2} + V(x)\psi(x) = E\psi(x). \quad (3.1)$$

We start with simple finite-difference schemes and discretize space into intervals of length Δx and denote the space points by

$$x_n = n\Delta x \quad (3.2)$$

and the wave function at these points by

$$\psi_n = \psi(x_n). \quad (3.3)$$

3.1.1 The Numerov algorithm

After rewriting the second order differential equation to a coupled system of two first order differential equations, any ODE solver such as the Runge-Kutta method could be applied, but there exist better methods. For the special form

$$\psi''(x) + k(x)\psi(x) = 0, \quad (3.4)$$

of the Schrödinger equation, with $k(x) = 2m(E - V(x))/\hbar^2$ we can derive the Numerov algorithm by starting from the Taylor expansion of ψ_n :

$$\psi_{n\pm 1} = \psi_n \pm \Delta x \psi'_n + \frac{\Delta x^2}{2} \psi''_n \pm \frac{\Delta x^3}{6} \psi^{(3)}_n + \frac{\Delta x^4}{24} \psi^{(4)}_n \pm \frac{\Delta x^5}{120} \psi^{(5)}_n + O(\Delta x^6) \quad (3.5)$$

Adding ψ_{n+1} and ψ_{n-1} we obtain

$$\psi_{n+1} + \psi_{n-1} = 2\psi_n + (\Delta x)^2 \psi''_n + \frac{(\Delta x)^4}{12} \psi^{(4)}_n. \quad (3.6)$$

Replacing the fourth derivatives by a finite difference second derivative of the second derivatives

$$\psi_n^{(4)} = \frac{\psi''_{n+1} + \psi''_{n-1} - 2\psi''_n}{\Delta x^2} \quad (3.7)$$

and substituting $-k(x)\psi(x)$ for $\psi''(x)$ we obtain the Numerov algorithm

$$\begin{aligned} \left(1 + \frac{(\Delta x)^2}{12} k_{n+1}\right) \psi_{n+1} &= 2 \left(1 - \frac{5(\Delta x)^2}{12} k_n\right) \psi_n \\ &\quad - \left(1 + \frac{(\Delta x)^2}{12} k_{n-1}\right) \psi_{n-1} + O(\Delta x^6), \end{aligned} \quad (3.8)$$

which is locally of sixth order!

Initial values

To start the Numerov algorithm we need the wave function not just at one but at two initial values and will now present several ways to obtain these.

For potentials $V(x)$ with reflection symmetry $V(x) = V(-x)$ the wave functions need to be either even $\psi(x) = \psi(-x)$ or odd $\psi(x) = -\psi(-x)$ under reflection, which can be used to find initial values:

- For the even solution we use a half-integer mesh with mesh points $x_{n+1/2} = (n + 1/2)\Delta x$ and pick initial values $\psi(x_{-1/2}) = \psi(x_{1/2}) = 1$.
- For the odd solution we know that $\psi(0) = -\psi(0)$ and hence $\psi(0) = 0$, specifying the first starting value. Using an integer mesh with mesh points $x_n = n\Delta x$ we pick $\psi(x_1) = 1$ as the second starting value.

In general potentials we need to use other approaches. If the potential vanishes for large distances: $V(x) = 0$ for $|x| \geq a$ we can use the exact solution of the Schrödinger equation at large distances to define starting points, e.g.

$$\psi(-a) = 1 \quad (3.9)$$

$$\psi(-a - \Delta x) = \exp(-\Delta x \sqrt{2mE/\hbar}). \quad (3.10)$$

Finally, if the potential never vanishes we need to begin with a single starting value $\psi(x_0)$ and obtain the second starting value $\psi(x_1)$ by performing an integration over the first time step $\Delta\tau$ with an Euler or Runge-Kutta algorithm.

3.1.2 The one-dimensional scattering problem

The scattering problem is the numerically easiest quantum problem since solutions exist for all energies $E > 0$, if the potential vanishes at large distances ($V(x) \rightarrow 0$ for $|x| \rightarrow \infty$). The solution becomes particularly simple if the potential is nonzero only on a finite interval $[0, a]$. For a particle approaching the potential barrier from the left ($x < 0$) we can make the following ansatz for the free propagation when $x < 0$:

$$\psi_L(x) = A \exp(-iqx) + B \exp(iqx) \quad (3.11)$$

where A is the amplitude of the incoming wave and B the amplitude of the reflected wave. On the right hand side, once the particle has left the region of finite potential ($x > a$), we can again make a free propagation ansatz,

$$\psi_R(x) = C \exp(-iqx) \quad (3.12)$$

The coefficients A , B and C have to be determined self-consistently by matching to a numerical solution of the Schrödinger equation in the interval $[0, a]$. This is best done in the following way:

- Set $C = 1$ and use the two points a and $a + \Delta x$ as starting points for a Numerov integration.
- Integrate the Schrödinger equation numerically – backwards in space, from a to 0 – using the Numerov algorithm.
- Match the numerical solution of the Schrödinger equation for $x < 0$ to the free propagation ansatz (3.11) to determine A and B .

Once A and B have been determined the reflection and transmission probabilities R and T are given by

$$R = |B|^2/|A|^2 \quad (3.13)$$

$$T = 1/|A|^2 \quad (3.14)$$

3.1.3 Bound states and solution of the eigenvalue problem

While there exist scattering states for all energies $E > 0$, bound states solutions of the Schrödinger equation with $E < 0$ exist only for discrete energy eigenvalues. Integrating the Schrödinger equation from $-\infty$ to $+\infty$ the solution will diverge to $\pm\infty$ as $x \rightarrow \infty$ for almost all values. These functions cannot be normalized and thus do not constitute solutions to the Schrödinger equation. Only for some special eigenvalues E , will the solution go to zero as $x \rightarrow \infty$.

A simple eigensolver can be implemented using the following shooting method, where we again will assume that the potential is zero outside an interval $[0, a]$:

- Start with an initial guess E
- Integrate the Schrödinger equation for $\psi_E(x)$ from $x = 0$ to $x_f \gg a$ and determine the value $\psi_E(x_f)$
- use a root solver, such as a bisection method (see appendix A.1), to look for an energy E with $\psi_E(x_f) \approx 0$

This algorithm is not ideal since the divergence of the wave function for $x \pm \infty$ will cause roundoff error to proliferate.

A better solution is to integrate the Schrödinger equation from both sides towards the center:

- We pick a starting point b and choose as energy $E = V(b)$

- Starting from $x = 0$ we integrate the left hand side solution $\psi_L(x)$ to a chosen point b and obtain $\psi_L(b)$ and a numerical estimate for $\psi'_L(b) = (\psi_L(b) - \psi_L(b - \Delta x)) / \Delta x$.
- Starting from $x = a$ we integrate the right hand solution $\psi_R(x)$ down to the same point b and obtain $\psi_R(b)$ and a numerical estimate for $\psi'_R(b) = (\psi_R(b + \Delta x) - \psi_R(b)) / \Delta x$.
- At the point b the wave functions and their first two derivatives have to match, since solutions to the Schrödinger equation have to be twice continuously differentiable. Keeping in mind that we can multiply the wave functions by an arbitrary factor we obtain the conditions

$$\psi_L(b) = \alpha\psi_R(b) \quad (3.15)$$

$$\psi'_L(b) = \alpha\psi'_R(b) \quad (3.16)$$

$$\psi''_L(b) = \alpha\psi''_R(b) \quad (3.17)$$

The last condition is automatically fulfilled since by the choice $V(b) = E$ the Schrödinger equation at b reduces to $\psi''(b) = 0$. The first two conditions can be combined to the condition that the logarithmic derivatives vanish:

$$\frac{d \log \psi_L}{dx} \Big|_{x=b} = \frac{\psi'_L(b)}{\psi_L(b)} = \frac{\psi'_R(b)}{\psi_R(b)} = \frac{d \log \psi_R}{dx} \Big|_{x=b} \quad (3.18)$$

- This last equation has to be solved for in a shooting method, e.g. using a bisection algorithm

Finally, at the end of the calculation, normalize the wave function.

3.2 The time-independent Schrödinger equation in higher dimensions

The time independent Schrödinger equation in more than one dimension is a partial differential equation and cannot, in general, be solved by a simple ODE solver such as the Numerov algorithm. Before employing a PDE solver we should thus always first try to reduce the problem to a one-dimensional problem. This can be done if the problem factorizes.

3.2.1 Factorization along coordinate axis

A first example is a three-dimensional Schrödinger equation in a cubic box with potential $V(\vec{r}) = V(x)V(y)V(z)$ with $\vec{r} = (x, y, z)$. Using the product ansatz

$$\psi(\vec{r}) = \psi_x(x)\psi_y(y)\psi_z(z) \quad (3.19)$$

the PDE factorizes into three ODEs which can be solved as above.

3.2.2 Potential with spherical symmetry

Another famous trick is possible for spherically symmetric potentials with $V(\vec{r}) = V(|\vec{r}|)$ where an ansatz using spherical harmonics

$$\psi_{l,m}(\vec{r}) = \psi_{l,m}(r, \theta, \phi) = \frac{u(r)}{r} Y_{lm}(\theta, \phi) \quad (3.20)$$

can be used to reduce the three-dimensional Schrödinger equation to a one-dimensional one for the radial wave function $u(r)$:

$$\left[-\frac{\hbar^2}{2m} \frac{d^2}{dr^2} + \frac{\hbar^2 l(l+1)}{2mr^2} + V(r) \right] u(r) = Eu(r) \quad (3.21)$$

in the interval $[0, \infty[$. Given the singular character of the potential for $r \rightarrow 0$, a numerical integration should start at large distances r and integrate towards $r = 0$, so that the largest errors are accumulated only at the last steps of the integration.

In the exercises we will solve a three-dimensional scattering problem and calculate the scattering length for two atoms.

3.2.3 Finite difference methods

The simplest solvers for partial differential equations, the finite difference solvers can also be used for the Schrödinger equation. Replacing differentials by differences we convert the Schrödinger equation to a system of coupled linear equations. Starting from the three-dimensional Schrödinger equation (we set $\hbar = 1$ from now on)

$$\nabla^2 \psi(\vec{x}) + 2m(V - E(\vec{x}))\psi(\vec{x}) = 0, \quad (3.22)$$

we discretize space and obtain the system of linear equations

$$\begin{aligned} & \frac{1}{\Delta x^2} [\psi(x_{n+1}, y_n, z_n) + \psi(x_{n-1}, y_n, z_n) \\ & \quad + \psi(x_n, y_{n+1}, z_n) + \psi(x_n, y_{n-1}, z_n) \\ & \quad + \psi(x_n, y_n, z_{n+1}) + \psi(x_n, y_n, z_{n-1})] \\ & + \left[2m(V(\vec{x}) - E) - \frac{6}{\Delta x^2} \right] \psi(x_n, y_n, z_n) = 0. \end{aligned} \quad (3.23)$$

For the scattering problem a linear equation solver can now be used to solve the system of equations. For small linear problems Mathematica can be used, or the `dssysv` function of the LAPACK library. For larger problems it is essential to realize that the matrices produced by the discretization of the Schrödinger equation are usually very sparse, meaning that only $O(N)$ of the N^2 matrix elements are nonzero. For these sparse systems of equations, optimized iterative numerical algorithms exist¹ and are implemented in numerical libraries such as in the ITL library.²

¹R. Barret, M. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods* (SIAM, 1993)

²J.G. Siek, A. Lumsdaine and Lie-Quan Lee, *Generic Programming for High Performance Numerical Linear Algebra in Proceedings of the SIAM Workshop on Object Oriented Methods for Inter-operable Scientific and Engineering Computing (OO'98)* (SIAM, 1998); the library is available on the web at: <http://www.osl.iu.edu/research/itl/>

To calculate bound states, an eigenvalue problem has to be solved. For small problems, where the full matrix can be stored in memory, Mathematica or the `dSYEV` eigen-solver in the LAPACK library can be used. For bigger systems, sparse solvers such as the Lanczos algorithm (see appendix A.2) are best. Again there exist efficient implementations³ of iterative algorithms for sparse matrices.⁴

3.2.4 Variational solutions using a finite basis set

In the case of general potentials, or for more than two particles, it will not be possible to reduce the Schrödinger equation to a one-dimensional problem and we need to employ a PDE solver. One approach will again be to discretize the Schrödinger equation on a discrete mesh using a finite difference approximation. A better solution is to expand the wave functions in terms of a finite set of basis functions

$$|\phi\rangle = \sum_{i=1}^N a_i |u_i\rangle. \quad (3.24)$$

To estimate the ground state energy we want to minimize the energy of the variational wave function

$$E^* = \frac{\langle\phi|H|\phi\rangle}{\langle\phi|\phi\rangle}. \quad (3.25)$$

Keep in mind that, since we only chose a finite basis set $\{|u_i\rangle\}$ the variational estimate E^* will always be larger than the true ground state energy E_0 , but will converge towards E_0 as the size of the basis set is increased, e.g. by reducing the mesh size in a finite element basis.

To perform the minimization we denote by

$$H_{ij} = \langle u_i | H | u_j \rangle = \int d\vec{r} u_i(\vec{r})^* \left(-\frac{\hbar^2}{2m} \nabla^2 + V \right) u_j(\vec{r}) \quad (3.26)$$

the matrix elements of the Hamilton operator H and by

$$S_{ij} = \langle u_i | u_j \rangle = \int d\vec{r} u_i(\vec{r})^* u_j(\vec{r}) \quad (3.27)$$

the overlap matrix. Note that for an orthogonal basis set, S_{ij} is the identity matrix δ_{ij} . Minimizing equation (3.25) we obtain a generalized eigenvalue problem

$$\sum_j H_{ij} a_j = E \sum_k S_{ik} a_k. \quad (3.28)$$

or in a compact notation with $\vec{a} = (a_1, \dots, a_N)$

$$H\vec{a} = E S \vec{a}. \quad (3.29)$$

³<http://www.comp-phys.org/software/ietl/>

⁴Z. Bai, J. Demmel and J. Dongarra (Eds.), *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide* (SIAM, 2000).

If the basis set is orthogonal this reduces to an ordinary eigenvalue problem and we can use the Lanczos algorithm.

In the general case we have to find orthogonal matrices U such that $U^T S U$ is the identity matrix. Introducing a new vector $\vec{b} = U^{-1} \vec{a}$, we can then rearrange the problem into

$$\begin{aligned} H\vec{a} &= E\vec{a} \\ HU\vec{b} &= ESU\vec{b} \\ U^T H U \vec{b} &= EU^T S U \vec{b} = E\vec{b} \end{aligned} \quad (3.30)$$

and we end up with a standard eigenvalue problem for $U^T H U$. Mathematica and LAPACK both contain eigensolvers for such generalized eigenvalue problems.

Example: the anharmonic oscillator

The final issue is the choice of basis functions. It is advantageous to make use of known solutions to a similar problem as we will illustrate in the case of an anharmonic oscillator with Hamilton operator

$$\begin{aligned} H &= H_0 + \lambda q^4 \\ H_0 &= \frac{1}{2}(p^2 + q^2), \end{aligned} \quad (3.31)$$

where the harmonic oscillator H_0 was already discussed in section 2.4.1. It makes sense to use the N lowest harmonic oscillator eigenvectors $|n\rangle$ as basis states of a finite basis and write the Hamiltonian as

$$H = \frac{1}{2}\hat{n} + \lambda\hat{q}^4 = \frac{1}{2}\hat{n} + \frac{\lambda}{4}(a^\dagger + a)^4 \quad (3.32)$$

Since the operators a and a^\dagger are nonzero only in the first sub or superdiagonal, the resulting matrix is a banded matrix of bandwidth 9. A sparse eigensolver such as the Lanczos algorithm can again be used to calculate the spectrum. Note that since we use the orthonormal eigenstates of H_0 as basis elements, the overlap matrix S here is the identity matrix and we have to deal only with a standard eigenvalue problem.

The finite element method

In cases where we have irregular geometries or want higher precision than the lowest order finite difference method, and do not know a suitable set of basis function, the finite element method (FEM) should be chosen over the finite difference method. Since explaining the FEM can take a full semester in itself, we refer interested students to classes on solving partial differential equations.

3.3 The time-dependent Schrödinger equation

Finally we will reintroduce the time dependence to study dynamics in non-stationary quantum systems.

3.3.1 Spectral methods

By introducing a basis and solving for the complete spectrum of energy eigenstates we can directly solve the time-dependent problem in the case of a stationary Hamiltonian. This is a consequence of the linearity of the Schrödinger equation.

To calculate the time evolution of a state $|\psi(t_0)\rangle$ from time t_0 to t we first solve the stationary eigenvalue problem $H|\phi\rangle = E|\phi\rangle$ and calculate the eigenvectors $|\phi_n\rangle$ and eigenvalues ϵ_n . Next we represent the initial wave function $|\psi\rangle$ by a spectral decomposition

$$|\psi(t_0)\rangle = \sum_n c_n |\phi_n\rangle. \quad (3.33)$$

Since each of the $|\phi_n\rangle$ is an eigenvector of H , the time evolution $e^{-i\hbar H(t-t_0)}$ is trivial and we obtain at time t :

$$|\psi(t)\rangle = \sum_n c_n e^{-i\hbar\epsilon_n(t-t_0)} |\phi_n\rangle. \quad (3.34)$$

3.3.2 Direct numerical integration

If the number of basis states is too large to perform a complete diagonalization of the Hamiltonian, or if the Hamiltonian changes over time we need to perform a direct integration of the Schrödinger equation. Like other initial value problems of partial differential equations the Schrödinger equation can be solved by the method of lines. After choosing a set of basis functions or discretizing the spatial derivatives we obtain a set of coupled ordinary differential equations which can be evolved for each point along the time line (hence the name) by standard ODE solvers.

In the remainder of this chapter we use the symbol H to refer the representation of the Hamiltonian in the chosen finite basis set. A forward Euler scheme

$$|\psi(t_{n+1})\rangle = |\psi(t_n)\rangle - i\hbar\Delta_t H |\psi(t_n)\rangle \quad (3.35)$$

is not only numerically unstable. It also violates the conservation of the norm of the wave function $\langle\psi|\psi\rangle = 1$. Since the exact quantum evolution

$$\psi(x, t + \Delta_t) = e^{-i\hbar H \Delta_t} \psi(x, t). \quad (3.36)$$

is unitary and thus conserves the norm, we want to look for a unitary approximant as integrator. Instead of using the forward Euler method (3.35) which is just a first order Taylor expansion of the exact time evolution

$$e^{-i\hbar H \Delta_t} = 1 - i\hbar H \Delta_t + \mathcal{O}(\Delta_t^2), \quad (3.37)$$

we reformulate the time evolution operator as

$$e^{-i\hbar H \Delta_t} = (e^{i\hbar H \Delta_t/2})^{-1} e^{-i\hbar H \Delta_t/2} = \left(1 + i\hbar H \frac{\Delta_t}{2}\right)^{-1} \left(1 - i\hbar H \frac{\Delta_t}{2}\right) + \mathcal{O}(\Delta_t^3), \quad (3.38)$$

which is unitary!

This gives the simplest stable and unitary integrator algorithm

$$\psi(x, t + \Delta_t) = \left(1 + i\hbar H \frac{\Delta_t}{2}\right)^{-1} \left(1 - i\hbar H \frac{\Delta_t}{2}\right) \psi(x, t) \quad (3.39)$$

or equivalently

$$\left(1 + i\hbar H \frac{\Delta_t}{2}\right) \psi(x, t + \Delta_t) = \left(1 - i\hbar H \frac{\Delta_t}{2}\right) \psi(x, t). \quad (3.40)$$

Unfortunately this is an implicit integrator. At each time step, after evaluating the right hand side a linear system of equations needs to be solved. For one-dimensional problems the matrix representation of H is often tridiagonal and a tridiagonal solver can be used. In higher dimensions the matrix H will no longer be simply tridiagonal but still very sparse and we can use iterative algorithms, similar to the Lanczos algorithm for the eigenvalue problem. For details about these algorithms we refer to the nice summary at <http://mathworld.wolfram.com/topics/Templates.html> and especially the biconjugate gradient (BiCG) algorithm. Implementations of this algorithm are available, e.g. in the Iterative Template Library (ITL).

3.3.3 The split operator method

A simpler and explicit method is possible for a quantum particle in the real space picture with the “standard” Schrödinger equation (2.52). Writing the Hamilton operator as

$$H = \hat{T} + \hat{V} \quad (3.41)$$

with

$$\hat{T} = \frac{1}{2m} \hat{p}^2 \quad (3.42)$$

$$\hat{V} = V(\vec{x}) \quad (3.43)$$

it is easy to see that \hat{V} is diagonal in position space while \hat{T} is diagonal in momentum space. If we split the time evolution as

$$e^{-i\hbar\Delta_t H} = e^{-i\hbar\Delta_t \hat{V}/2} e^{-i\hbar\Delta_t \hat{T}} e^{-i\hbar\Delta_t \hat{V}/2} + O(\Delta_t^3) \quad (3.44)$$

we can perform the individual time evolutions $e^{-i\hbar\Delta_t \hat{V}/2}$ and $e^{-i\hbar\Delta_t \hat{T}}$ exactly:

$$\left[e^{-i\hbar\Delta_t \hat{V}/2} |\psi\rangle \right] (\vec{x}) = e^{-i\hbar\Delta_t V(\vec{x})/2} \psi(\vec{x}) \quad (3.45)$$

$$\left[e^{-i\hbar\Delta_t \hat{T}/2} |\psi\rangle \right] (\vec{k}) = e^{-i\hbar\Delta_t \|\vec{k}\|^2/2m} \psi(\vec{k}) \quad (3.46)$$

in real space for the first term and momentum space for the second term. This requires a basis change from real to momentum space, which is efficiently performed using a Fast Fourier Transform (FFT) algorithm. Propagating for a time $t = N\Delta_t$, two consecutive

applications of $e^{-i\hbar\Delta_t \hat{V}/2}$ can easily be combined into a propagation by a full time step $e^{-i\hbar\Delta_t \hat{V}}$, resulting in the propagation:

$$\begin{aligned} e^{-i\hbar\Delta_t H} &= \left(e^{-i\hbar\Delta_t \hat{V}/2} e^{-i\hbar\Delta_t \hat{T}} e^{-i\hbar\Delta_t \hat{V}/2} \right)^N + O(\Delta_t^2) \\ &= e^{-i\hbar\Delta_t \hat{V}/2} \left[e^{-i\hbar\Delta_t \hat{T}} e^{-i\hbar\Delta_t \hat{V}} \right]^{N-1} e^{-i\hbar\Delta_t \hat{T}} e^{-i\hbar\Delta_t \hat{V}/2} \end{aligned} \quad (3.47)$$

and the discretized algorithm starts as

$$\psi_1(\vec{x}) = e^{-i\hbar\Delta_t V(\vec{x})/2} \psi_0(\vec{x}) \quad (3.48)$$

$$\psi_1(\vec{k}) = \mathcal{F}\psi_1(\vec{x}) \quad (3.49)$$

where \mathcal{F} denotes the Fourier transform and \mathcal{F}^{-1} will denote the inverse Fourier transform. Next we propagate in time using full time steps:

$$\psi_{2n}(\vec{k}) = e^{-i\hbar\Delta_t |\vec{k}|^2/2m} \psi_{2n-1}(\vec{k}) \quad (3.50)$$

$$\psi_{2n}(\vec{x}) = \mathcal{F}^{-1}\psi_{2n}(\vec{k}) \quad (3.51)$$

$$\psi_{2n+1}(\vec{x}) = e^{-i\hbar\Delta_t V(\vec{x})} \psi_{2n}(\vec{x}) \quad (3.52)$$

$$\psi_{2n+1}(\vec{k}) = \mathcal{F}\psi_{2n+1}(\vec{x}) \quad (3.53)$$

except that in the last step we finish with another half time step in real space:

$$\psi_{2N+1}(\vec{x}) = e^{-i\hbar\Delta_t V(\vec{x})/2} \psi_{2N}(\vec{x}) \quad (3.54)$$

This is a fast and unitary integrator for the Schrödinger equation in real space. It could be improved by replacing the locally third order splitting (3.44) by a fifth-order version involving five instead of three terms.

Chapter 4

Introduction to many-body quantum mechanics

4.1 The complexity of the quantum many-body problem

After learning how to solve the 1-body Schrödinger equation, let us next generalize to more particles. If a single body quantum problem is described by a Hilbert space \mathcal{H} of dimension $\dim \mathcal{H} = d$ then N *distinguishable* quantum particles are described by the tensor product of N Hilbert spaces

$$\mathcal{H}^{(N)} \equiv \mathcal{H}^{\otimes N} \equiv \bigotimes_{i=1}^N \mathcal{H} \quad (4.1)$$

with dimension d^N .

As a first example, a single spin-1/2 has a Hilbert space $\mathcal{H} = \mathbb{C}^2$ of dimension 2, but N spin-1/2 have a Hilbert space $\mathcal{H}^{(N)} = \mathbb{C}^{2^N}$ of dimension 2^N . Similarly, a single particle in three dimensional space is described by a complex-valued wave function $\psi(\vec{x})$ of the position \vec{x} of the particle, while N distinguishable particles are described by a complex-valued wave function $\psi(\vec{x}_1, \dots, \vec{x}_N)$ of the positions $\vec{x}_1, \dots, \vec{x}_N$ of the particles. Approximating the Hilbert space \mathcal{H} of the single particle by a finite basis set with d basis functions, the N -particle basis approximated by the same finite basis set for single particles needs d^N basis functions.

This exponential scaling of the Hilbert space dimension with the number of particles is a big challenge. Even in the simplest case – a spin-1/2 with $d = 2$, the basis for $N = 30$ spins is already of size $2^{30} \approx 10^9$. A single complex vector needs 16 GByte of memory and will not fit into the memory of your personal computer anymore.

This challenge will be addressed later in this course by learning about

1. approximative methods, reducing the many-particle problem to a single-particle problem
2. quantum Monte Carlo methods for bosonic and magnetic systems
3. brute-force methods solving the exact problem in a huge Hilbert space for modest numbers of particles

4.2 Indistinguishable particles

4.2.1 Bosons and fermions

In quantum mechanics we assume that elementary particles, such as the electron or photon, are indistinguishable: there is no serial number painted on the electrons that would allow us to distinguish two electrons. Hence, if we exchange two particles the system is still the same as before. For a two-body wave function $\psi(\vec{q}_1, \vec{q}_2)$ this means that

$$\psi(\vec{q}_2, \vec{q}_1) = e^{i\phi} \psi(\vec{q}_1, \vec{q}_2), \quad (4.2)$$

since upon exchanging the two particles the wave function needs to be identical, up to a phase factor $e^{i\phi}$. In three dimensions the first homotopy group is trivial and after doing two exchanges we need to be back at the original wave function¹

$$\psi(\vec{q}_1, \vec{q}_2) = e^{i\phi} \psi(\vec{q}_2, \vec{q}_1) = e^{2i\phi} \psi(\vec{q}_1, \vec{q}_2), \quad (4.3)$$

and hence $e^{2i\phi} = \pm 1$:

$$\psi(\vec{q}_2, \vec{q}_1) = \pm \psi(\vec{q}_1, \vec{q}_2) \quad (4.4)$$

The many-body Hilbert space can thus be split into orthogonal subspaces, one in which particles pick up a $-$ sign and are called fermions, and the other where particles pick up a $+$ sign and are called bosons.

Bosons

For bosons the general many-body wave function thus needs to be symmetric under permutations. Instead of an arbitrary wave function $\psi(\vec{q}_1, \dots, \vec{q}_N)$ of N particles we use the symmetrized wave function

$$\Psi^{(S)} = \mathcal{S}_+ \psi(\vec{q}_1, \dots, \vec{q}_N) \equiv \mathcal{N}_S \sum_p \psi(\vec{q}_{p(1)}, \dots, \vec{q}_{p(N)}), \quad (4.5)$$

where the sum goes over all permutations p of N particles, and \mathcal{N}_S is a normalization factor.

¹As a side remark we want to mention that in two dimensions the first homotopy group is \mathbb{Z} and not trivial: it matters whether we move the particles clock-wise or anti-clock wise when exchanging them, and two clock-wise exchanges are not the identity anymore. Then more general, anyonic, statistics are possible.

Fermions

For fermions the wave function has to be antisymmetric under exchange of any two fermions, and we use the anti-symmetrized wave function

$$\Psi^{(A)} \mathcal{S}_- \psi(\vec{q}_1, \dots, \vec{q}_N) \equiv \mathcal{N}_A \sum_p \text{sgn}(p) \psi(\vec{q}_{p(1)}, \dots, \vec{q}_{p(N)}), \quad (4.6)$$

where $\text{sgn}(p) = \pm 1$ is the sign of the permutation and \mathcal{N}_A again a normalization factor.

A consequence of the antisymmetrization is that no two fermions can be in the same state as a wave function

$$\psi(\vec{q}_1, \vec{q}_2) = \phi(\vec{q}_1)\phi(\vec{q}_2) \quad (4.7)$$

since this vanishes under antisymmetrization:

$$\Psi(\vec{q}_1, \vec{q}_2) = \psi(\vec{q}_1, \vec{q}_2) - \psi(\vec{q}_2, \vec{q}_1) = \phi(\vec{q}_1)\phi(\vec{q}_2) - \phi(\vec{q}_2)\phi(\vec{q}_1) = 0 \quad (4.8)$$

Spinful fermions

Fermions, such as electrons, usually have a spin-1/2 degree of freedom in addition to their orbital wave function. The full wave function as a function of a generalized coordinate $\vec{x} = (\vec{q}, \sigma)$ including both position \vec{q} and spin σ .

4.2.2 The Fock space

The Hilbert space describing a quantum many-body system with $N = 0, 1, \dots, \infty$ particles is called the Fock space. It is the direct sum of the appropriately symmetrized single-particle Hilbert spaces \mathcal{H} :

$$\bigoplus_{N=0}^{\infty} S_{\pm} \mathcal{H}^{\otimes n} \quad (4.9)$$

where S_+ is the symmetrization operator used for bosons and S_- is the anti-symmetrization operator used for fermions.

The occupation number basis

Given a basis $\{|\phi_1\rangle, \dots, |\phi_L\rangle\}$ of the single-particle Hilbert space \mathcal{H} , a basis for the Fock space is constructed by specifying the number of particles n_i occupying the single-particle wave function $|f_1\rangle$. The wave function of the state $|n_1, \dots, n_L\rangle$ is given by the appropriately symmetrized and normalized product of the single particle wave functions. For example, the basis state $|1, 1\rangle$ has wave function

$$\frac{1}{\sqrt{2}} [\phi_1(\vec{x}_1)\phi_2(\vec{x}_2) \pm \phi_1(\vec{x}_2)\phi_2(\vec{x}_1)] \quad (4.10)$$

where the + sign is for bosons and the - sign for fermions.

For bosons the occupation numbers n_i can go from 0 to ∞ , but for fermions they are restricted to $n_i = 0$ or 1 since no two fermions can occupy the same state.

The Slater determinant

The antisymmetrized and normalized product of N single-particle wave functions ϕ_i can be written as a determinant, called the Slater determinant

$$\mathcal{S}_- \prod_{i_1}^N \phi_i(\vec{x}_i) = \frac{1}{\sqrt{N}} \begin{vmatrix} \phi_1(\vec{x}_1) & \cdots & \phi_N(\vec{x}_1) \\ \vdots & & \vdots \\ \phi_1(\vec{x}_N) & \cdots & \phi_N(\vec{x}_N) \end{vmatrix}. \quad (4.11)$$

Note that while the set of Slater determinants of single particle basis functions forms a basis of the fermionic Fock space, the general fermionic many body wave function is a linear superposition of many Slater determinants and cannot be written as a single Slater determinant. The Hartree Fock method, discussed below, will simplify the quantum many body problem to a one body problem by making the approximation that the ground state wave function can be described by a single Slater determinant.

4.2.3 Creation and annihilation operators

Since it is very cumbersome to work with appropriately symmetrized many body wave functions, we will mainly use the formalism of second quantization and work with creation and annihilation operators.

The annihilation operator $a_{i,\sigma}$ associated with a basis function $|\phi_i\rangle$ is defined as the result of the inner product of a many body wave function $|\Psi\rangle$ with this basis function $|\phi_i\rangle$. Given an N -particle wave function $|\Psi^{(N)}\rangle$ the result of applying the annihilation operator is an $N - 1$ -particle wave function $|\tilde{\Psi}^{(N)}\rangle = a_i |\Psi^{(N)}\rangle$. It is given by the appropriately symmetrized inner product

$$\tilde{\Psi}(\vec{x}_1, \dots, \vec{x}_{N-1}) = \mathcal{S}_\pm \int d\vec{x}_N f_i^\dagger(\vec{x}_N) \Psi(\vec{x}_1, \dots, \vec{x}_N). \quad (4.12)$$

Applied to a single-particle basis state $|\phi_j\rangle$ the result is

$$a_i |\phi_j\rangle = \delta_{ij} |0\rangle \quad (4.13)$$

where $|0\rangle$ is the “vacuum” state with no particles.

The creation operator a_i^\dagger is defined as the adjoint of the annihilation operator a_i . Applying it to the vacuum “creates” a particle with wave function ϕ_i :

$$|\phi_i\rangle = a_i^\dagger |0\rangle \quad (4.14)$$

For sake of simplicity and concreteness we will now assume that the L basis functions $|\phi_i\rangle$ of the single particle Hilbert space factor into $L/(2S + 1)$ orbital wave functions $f_i(\vec{q})$ and $2S + 1$ spin wave functions $|\sigma\rangle$, where $\sigma = -S, -S + 1, \dots, S$. We will write creation and annihilation operators $a_{i,\sigma}^\dagger$ and $a_{i,\sigma}$ where i is the orbital index and σ the spin index. The most common cases will be spinless bosons with $S = 0$, where the spin index can be dropped and spin-1/2 fermions, where the spin can be up ($+1/2$) or down ($-1/2$).

Commutation relations

The creation and annihilation operators fulfill certain canonical commutation relations, which we will first discuss for an orthogonal set of basis functions. We will later generalize them to non-orthogonal basis sets.

For bosons, the commutation relations are the same as that of the ladder operators discussed for the harmonic oscillator (2.62):

$$[a_i, a_j] = [a_i^\dagger, a_j^\dagger] = 0 \quad (4.15)$$

$$[a_i, a_j^\dagger] = \delta_{ij}. \quad (4.16)$$

For fermions, on the other hand, the operators anticommute

$$\begin{aligned} \{a_{j\sigma'}^\dagger, a_{i\sigma}\} &= \{a_{i\sigma}^\dagger, a_{j\sigma'}\} = \delta_{\sigma\sigma'}\delta_{ij} \\ \{a_{i\sigma}, a_{j\sigma'}\} &= \{a_{i\sigma}^\dagger, a_{j\sigma'}^\dagger\} = 0. \end{aligned} \quad (4.17)$$

The anti-commutation implies that

$$(a_i^\dagger)^2 = a_i^\dagger a_i^\dagger = -a_i^\dagger a_i^\dagger \quad (4.18)$$

and that thus

$$(a_i^\dagger)^2 = 0, \quad (4.19)$$

as expected since no two fermions can exist in the same state.

Fock basis in second quantization and normal ordering

The basis state $|n_1, \dots, n_L\rangle$ in the occupation number basis can easily be expressed in terms of creation operators:

$$|n_1, \dots, n_L\rangle = \prod_{i=1}^L (a_i^\dagger)^{n_i} |0\rangle = (a_1^\dagger)^{n_1} (a_2^\dagger)^{n_2} \cdots (a_L^\dagger)^{n_L} |0\rangle \quad (4.20)$$

For bosons the ordering of the creation operators does not matter, since the operators commute. For fermions, however, the ordering matters since the fermionic creation operators anticommute: and $a_1^\dagger a_2^\dagger |0\rangle = -a_2^\dagger a_1^\dagger |0\rangle$. We thus need to agree on a specific ordering of the creation operators to define what we mean by the state $|n_1, \dots, n_L\rangle$. The choice of ordering does not matter but we have to stay consistent and use e.g. the convention in equation (4.20).

Once the normal ordering is defined, we can derive the expressions for the matrix elements of the creation and annihilation operators in that basis. Using above normal ordering the matrix elements are

$$a_i |n_1, \dots, n_i, \dots, n_L\rangle = \delta_{n_i,1} (-1)^{\sum_{j=1}^{i-1} n_j} |n_1, \dots, n_i - 1, \dots, n_L\rangle \quad (4.21)$$

$$a_i^\dagger |n_1, \dots, n_i, \dots, n_L\rangle = \delta_{n_i,0} (-1)^{\sum_{j=1}^{i-1} n_j} |n_1, \dots, n_i + 1, \dots, n_L\rangle \quad (4.22)$$

where the minus signs come from commuting the annihilation and creation operator to the correct position in the normal ordered product.

4.2.4 Nonorthogonal basis sets

In simulating the electronic properties of atoms and molecules below we will see that the natural choice of single particle basis functions centered around atoms will necessarily give a non-orthogonal set of basis functions. This is no problem, as long as the definition of the annihilation and creation operators is carefully generalized. For this generalization it will be useful to introduce the fermion field operators $\psi_\sigma^\dagger(\vec{r})$ and $\psi_\sigma(\vec{r})$, creating and annihilating a fermion localized at a single point \vec{r} in space. Their commutation relations are simply

$$\begin{aligned} \{\psi_{\sigma'}^\dagger(\vec{r}), \psi_\sigma(\vec{r}')\} &= \{\psi_\sigma^\dagger(\vec{r}), \psi_{\sigma'}(\vec{r}')\} = \delta_{\sigma\sigma'}\delta(\vec{r} - \vec{r}') \\ \{\psi_\sigma(\vec{r}), \psi_{\sigma'}(\vec{r}')\} &= \{\psi_\sigma^\dagger(\vec{r}), \psi_{\sigma'}^\dagger(\vec{r}')\} = 0. \end{aligned} \quad (4.23)$$

The scalar products of the basis functions define a matrix

$$S_{ij} = \int d^3\vec{r} f_i^*(\vec{r}) f_j(\vec{r}), \quad (4.24)$$

which is in general *not* the identity matrix. The associated annihilation operators $a_{i\sigma}$ are again defined as scalar products

$$a_{i\sigma} = \sum_j (S^{-1})_{ij} \int d^3\vec{r} f_j^*(\vec{r}) \psi_\sigma(\vec{r}). \quad (4.25)$$

The non-orthogonality causes the commutation relations of these operators to differ from those of normal fermion creation- and annihilation operators:

$$\begin{aligned} \{a_{i\sigma}^\dagger, a_{j\sigma'}\} &= \delta_{\sigma\sigma'}(S^{-1})_{ij} \\ \{a_{i\sigma}, a_{j\sigma'}\} &= \{a_{i\sigma}^\dagger, a_{j\sigma'}^\dagger\} = 0. \end{aligned} \quad (4.26)$$

Due to the non-orthogonality the adjoint $a_{i\sigma}^\dagger$ does *not* create a state with wave function f_i . This is done by the operator $\hat{a}_{i\sigma}^\dagger$, defined through:

$$\hat{a}_{i\sigma}^\dagger = \sum_j S_{ji} a_{i\sigma}^\dagger, \quad (4.27)$$

which has the following simple commutation relation with $a_{j\sigma}$:

$$\{\hat{a}_{i\sigma}^\dagger, a_{j\sigma}\} = \delta_{ij}. \quad (4.28)$$

The commutation relations of the $\hat{a}_{i\sigma}^\dagger$ and the $\hat{a}_{j\sigma'}$ are:

$$\begin{aligned} \{\hat{a}_{i\sigma}^\dagger, \hat{a}_{j\sigma'}\} &= \delta_{\sigma\sigma'} S_{ij} \\ \{\hat{a}_{i\sigma}, \hat{a}_{j\sigma'}\} &= \{\hat{a}_{i\sigma}^\dagger, \hat{a}_{j\sigma'}^\dagger\} = 0. \end{aligned} \quad (4.29)$$

We will need to keep the distinction between a and \hat{a} in mind when dealing with non-orthogonal basis sets.

Chapter 5

Quantum Monte Carlo

This chapter is devoted to the study of quantum many body systems using Monte Carlo techniques. We analyze two of the methods that belong to the large family of the quantum Monte Carlo techniques, namely the Path-Integral Monte Carlo (PIMC) and the Diffusion Monte Carlo (DMC, also named Green's function Monte Carlo). In the first section we start by introducing PIMC.

5.1 Path Integrals in Quantum Statistical Mechanics

In this section we introduce the path-integral description of the properties of quantum many-body systems. We show that path integrals permit to calculate the static properties of systems of Bosons at thermal equilibrium by means of Monte Carlo methods.

We consider a many-particle system described by the non-relativistic Hamiltonian

$$\hat{H} = \hat{T} + \hat{V}; \quad (5.1)$$

in coordinate representation the kinetic operator \hat{T} and the potential operator \hat{V} are defined as:

$$\hat{T} = -\frac{\hbar^2}{2m} \sum_{i=1}^N \Delta_i, \text{ and} \quad (5.2)$$

$$\hat{V} = V(\mathbf{R}). \quad (5.3)$$

In these equations \hbar is the Plank's constant divided by 2π , m the particles mass, N the number of particles and the vector $\mathbf{R} \equiv (\mathbf{r}_1, \dots, \mathbf{r}_N)$ describes their positions. We consider here systems in d dimensions, with fixed number of particles, temperature T , contained in a volume V .

In most case, the potential $V(\mathbf{R})$ is determined by inter-particle interactions, in which case it can be written as the sum of pair contributions $V(\mathbf{R}) = \sum_{i < j} v(\mathbf{r}_i - \mathbf{r}_j)$, where $v(\mathbf{r})$ is the inter-particle potential; it can also be due to an external field, call it $v_{\text{ext}}(\mathbf{r})$, in which case it is just the sum of single particle contributions $V(\mathbf{R}) = \sum_i v_{\text{ex}}(\mathbf{r}_i)$.

We first assume that particles, although being identical, are distinguishable. Therefore, they obey Boltzmann statistics. In section 5.1.3 we will describe the treatment of identical particles obeying Bose statistics.

All the static properties of a quantum many-body system in thermal equilibrium are obtainable from the thermal density matrix $\exp(-\beta \hat{H})$, where $\beta = 1/k_B T$, with k_B the Boltzmann's constant. The expectation value of an observable operator \hat{O} is:

$$\langle \hat{O} \rangle = \text{Tr} \left(\hat{O} \exp(-\beta \hat{H}) \right) / Z, \quad (5.4)$$

where the partition function Z is the trace of the density matrix:

$$Z = \text{Tr} \left(\exp(-\beta \hat{H}) \right). \quad (5.5)$$

In the following we will find convenient to use the density matrix in coordinate representation. We denote its matrix elements as:

$$\rho(\mathbf{R}, \mathbf{R}', \beta) \equiv \left\langle \mathbf{R} \left| \exp(-\beta \hat{H}) \right| \mathbf{R}' \right\rangle. \quad (5.6)$$

The partition function is the integral of the diagonal matrix elements over all possible configurations:

$$Z(N, T, V) = \int \rho(\mathbf{R}, \mathbf{R}, \beta) d\mathbf{R}. \quad (5.7)$$

The product of two density matrices is again a density matrix:

$$\exp\left(-(\beta_1 + \beta_2) \hat{H}\right) = \exp\left(-\beta_1 \hat{H}\right) \exp\left(-\beta_2 \hat{H}\right). \quad (5.8)$$

This property, often referred to as ‘product property’, written in coordinate representation gives a convolution integral:

$$\rho(\mathbf{R}_1, \mathbf{R}_3, \beta_1 + \beta_2) = \int \rho(\mathbf{R}_1, \mathbf{R}_2, \beta_1) \rho(\mathbf{R}_2, \mathbf{R}_3, \beta_2) d\mathbf{R}_2. \quad (5.9)$$

If we apply the product property M times we obtain the density matrix at the inverse temperature β as the product of M density matrices at the inverse temperature $\tau = \beta/M$. In operator form:

$$\exp\left(-\beta \hat{H}\right) = \left(\exp\left(-\tau \hat{H}\right) \right)^M. \quad (5.10)$$

We call *time step* the quantity τ . Eq. (5.10) written in coordinate representation becomes:

$$\begin{aligned} \rho(\mathbf{R}_1, \mathbf{R}_{M+1}, \beta) &= \int \cdots \int d\mathbf{R}_2 d\mathbf{R}_3 \cdots d\mathbf{R}_M \\ &\quad \rho(\mathbf{R}_1, \mathbf{R}_2, \tau) \rho(\mathbf{R}_2, \mathbf{R}_3, \tau) \cdots \rho(\mathbf{R}_M, \mathbf{R}_{M+1}, \tau). \end{aligned} \quad (5.11)$$

Eq. (5.11) is not useful as it is since the density matrices $\rho(\mathbf{R}_j, \mathbf{R}_{j+1}, \tau)$ are, in general, unknown quantities. We note, however, that if M is a large number, then the time-step τ , which corresponds to the high temperature MT , is small. If in eq. (5.11) we replace the exact density matrix $\rho(\mathbf{R}_j, \mathbf{R}_{j+1}, \tau)$ with a ‘short time’ or ‘high temperature’ approximation we obtain a multidimensional integral of known functions. Furthermore, in coordinate representation the density matrix is positive definite. It is known that many-variable integrals of positive functions can be calculated efficiently by means of Monte Carlo methods.

The simplest expression for the ‘high temperature’ density matrix is the so called *primitive approximation*. It consists in neglecting all terms beyond the one which is linear in τ in the left-hand side exponent of the following operator identity (*Baker-Campbell-Hausdorff relation*):

$$\exp\left(-\tau\left(\hat{T} + \hat{V}\right) + \frac{\tau^2}{2}\left[\hat{T}, \hat{V}\right] + \dots\right) = \exp\left(-\tau\hat{T}\right)\exp\left(-\tau\hat{V}\right). \quad (5.12)$$

(In this equation dots indicate terms which contain powers of τ higher than the second.) One obtains the following approximate expression for the density matrix operator:

$$\exp\left(-\tau\hat{H}\right) \cong \exp\left(-\tau\hat{T}\right)\exp\left(-\tau\hat{V}\right). \quad (5.13)$$

It is easy to write the matrix elements of the kinetic density matrix $\exp\left(-\tau\hat{T}\right)$ and the potential density matrix $\exp\left(-\tau\hat{V}\right)$ in coordinate representation. The latter is diagonal:

$$\langle \mathbf{R}_i | \exp\left(-\tau\hat{V}\right) | \mathbf{R}_{i+1} \rangle = \exp(-\tau V(\mathbf{R}_i)) \delta(\mathbf{R}_i - \mathbf{R}_{i+1}), \quad (5.14)$$

given that we consider potentials that are diagonal in coordinate space. The former, in free space, is a gaussian propagator (see section 5.1.2):

$$\langle \mathbf{R}_i | \exp\left(-\tau\hat{T}\right) | \mathbf{R}_{i+1} \rangle = (2\pi\hbar^2\tau/m)^{-dN/2} \exp\left[-\frac{(\mathbf{R}_i - \mathbf{R}_{i+1})^2}{2\hbar^2\tau/m}\right]. \quad (5.15)$$

For later convenience we introduce the following definition:

$$\rho^{\text{free}}(\mathbf{R}, \mathbf{R}', \tau) \equiv (2\pi\hbar^2\tau/m)^{-dN/2} \exp\left[-\frac{(\mathbf{R} - \mathbf{R}')^2}{2\hbar^2\tau/m}\right]. \quad (5.16)$$

In the limit of large Trotter number M equation (5.10) remains exact if we use the primitive approximation eq. (5.12) in its right hand side. This is guaranteed by the Trotter formula:

$$\exp\left(-\beta\left(\hat{T} + \hat{V}\right)\right) = \lim_{M \rightarrow +\infty} \left[\exp\left(-\tau\hat{T}\right)\exp\left(-\tau\hat{V}\right) \right]^M, \quad (5.17)$$

which holds for any pairs of operators bounded from below. The kinetic operator \hat{T} and the potential operators \hat{V} of interest to us satisfy this requirement. To make the

consequence of the Trotter formula explicit in coordinate representation we substitute the matrix elements of the kinetic and the potential density matrices eqs. (5.15) and (5.14) in the path-integral formula (5.11). We arrive at the following $dN(M - 1)$ -dimensional integral:

$$\rho(\mathbf{R}_1, \mathbf{R}_{M+1}, \beta) \cong \int \cdots \int \prod_{j=2}^M d\mathbf{R}_j \prod_{j=1}^M \{\rho^{\text{free}}(\mathbf{R}_j, \mathbf{R}_{j+1}, \tau) \exp[-\tau V(\mathbf{R}_j)]\}. \quad (5.18)$$

The Trotter formula guarantees that in the limit $M \rightarrow \infty$ this is an exact equation. If M is a large, but finite, number the integral (5.18) can be computed using the Monte Carlo procedure. One big issue is the determination of the lowest value of M for which the systematic error due to M being finite is smaller than the unavoidable statistical error associated to the Monte Carlo evaluation.

At this point it is useful to introduce some definitions we will employ extensively in the next lectures.

Many-particle path: also called ‘*system configuration*’, it is the set of the dNM coordinates $\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_M$.

Time-slice: the j -th term of a system configuration, indicated with \mathbf{R}_j , contains the dN coordinates of the N particles at imaginary time $(j - 1)\tau$ and will be called ‘*time-slice*’.

World line: the ‘*world line*’ i is the set of coordinates describing the path of the particle i in imaginary time: $\{\mathbf{r}_1^i, \mathbf{r}_2^i, \dots, \mathbf{r}_j^i, \dots, \mathbf{r}_M^i\}$.

Bead: we call ‘*beads*’ the M components of a world line.

The trace of the density matrix (5.18) gives the partition function:

$$Z(N, V, T) = \int \rho(\mathbf{R}_1, \mathbf{R}_1, \beta) d\mathbf{R}_1 = \int \cdots \int \prod_{j=1}^M d\mathbf{R}_j \prod_{j=1}^M \{\rho^{\text{free}}(\mathbf{R}_j, \mathbf{R}_{j+1}, \tau) \exp[-\tau V(\mathbf{R}_j)]\}. \quad (5.19)$$

For distinguishable particles $\mathbf{R}_{M+1} \equiv \mathbf{R}_1$. Note that eq. (5.19) represents the partition function of a classical system of polymers. Every polymer is a necklace of beads interacting as if they were connected by ideal springs. This harmonic interaction is due to the kinetic density matrix. In the primitive approximation beads with the same imaginary time index j , i.e., belonging to the same time-slice, interact with the inter-particle potential $v(r)$. With higher order approximations one generally introduces effective interparticle interactions. This is the famous mapping of quantum to classical systems introduced by Feynman to describe the properties of superfluid helium. Each quantum particle has been substituted by a classical polymer. The size of polymers is of order $\lambda_T = \sqrt{2\pi\hbar^2\beta/m}$, the de Broglie thermal wave-length, and represents the indetermination on the position of the corresponding quantum particle. In the section 5.1.3 we will

see how the indistinguishability of identical particles modifies the ‘polymer’ description of the quantum many body system.

5.1.1 Analogy inverse temperature – imaginary time

In the previous sections we have shown that the partition function of a quantum system can be decomposed using path-integrals. It is interesting to notice that a path-integral can be regarded as a time-evolution in *imaginary time*. To understand this, let us consider the time-dependent Schrödinger equation:

$$i\hbar \frac{\partial}{\partial t} \phi(\mathbf{R}, t) = \hat{H} \phi(\mathbf{R}, t). \quad (5.20)$$

The Green’s function of eq. (5.20) is:

$$G(\mathbf{R}, \mathbf{R}', t) = \left\langle \mathbf{R} \left| \exp \left(-it/\hbar \hat{H} \right) \right| \mathbf{R}' \right\rangle. \quad (5.21)$$

It is the solution of the Schrödinger equation with the initial condition $\phi(\mathbf{R}, 0) = \delta(\mathbf{R} - \mathbf{R}')$. It governs the time-evolution of the wave function. In fact, using the Green’s function one can write the differential equation (5.20) in the integral form:

$$\phi(\mathbf{R}, t) = \int G(\mathbf{R}, \mathbf{R}', t) \phi(\mathbf{R}', 0) d\mathbf{R}'. \quad (5.22)$$

Now, we can notice that eq. (5.21) is analogous to the thermal density matrix (5.6) once one substitutes $\beta \rightarrow it/\hbar$ in eq. (5.6).

5.1.2 Free-particle density matrix

Let us consider a free particle in 1D. The Hamiltonian describing this system is:

$$\hat{H} = -\frac{\hbar^2}{2m} \frac{d^2}{dx^2}. \quad (5.23)$$

It is easy to determine the thermal density matrix corresponding to this Hamiltonian. We start from the definition:

$$\rho(x, x', \beta) = \left\langle x \left| \exp \left(-\beta \hat{H} \right) \right| x' \right\rangle; \quad (5.24)$$

We introduce twice the completeness relation $\int |p\rangle \langle p| dp = \mathbf{I}$, where $|p\rangle$ are the eigenstates of the momentum operator:

$$\begin{aligned} \rho(x, x', \beta) &= \int dp \int dp' \langle x | p \rangle \left\langle p \left| \exp \left(-\beta \hat{H} \right) \right| p' \right\rangle \langle p' | x' \rangle = \\ &= \frac{1}{2\pi} \int dp/\hbar \exp(i(x-x')p/\hbar) \exp\left(-\frac{\beta}{2m}p^2\right). \end{aligned} \quad (5.25)$$

Here we have used the expression of the momentum eigenstates in coordinate space $\langle x|p\rangle = \frac{1}{\sqrt{2\pi\hbar}} \exp(ixp/\hbar)$, and their orthogonality $\langle p|p'\rangle = \delta(p-p')$. In the last integral in eq. (5.25) we recognize the inverse-Fourier transform of a Gaussian function. The Fourier transform $F(k)$ of the function $f(x) = \exp(-x^2/(4a^2))$ is again a Gaussian function:

$$F(k) = \sqrt{2a} \exp(ak^2). \quad (5.26)$$

Using this result in eq. (5.25) we obtain that the free-particle density matrix is a Gaussian propagator:

$$\rho(x, x', \beta) = \sqrt{\frac{m}{2\pi\beta\hbar^2}} \exp\left(-\frac{m}{2\beta\hbar^2}(x-x')^2\right). \quad (5.27)$$

5.1.3 Bose symmetry

The expression (5.19) for the partition function is not symmetrical under particle exchange, so it holds for distinguishable particles only. The correct expression for identical particles obeying Bose (Fermi) statistics should be symmetrical (anti-symmetrical) under particle exchange. A convenient way to symmetrize the density matrix (5.18) is to sum over all possible permutations of the particle labels in one of the two arguments:

$$\rho_{\text{Bose}}(\mathbf{R}_1, \mathbf{R}_2, \beta) = \frac{1}{N!} \sum_P \rho(\mathbf{R}_1, \mathbf{P}\mathbf{R}_2, \beta), \quad (5.28)$$

where \mathbf{P} is one of the $N!$ permutations of the particle labels; this means that $\mathbf{P}\mathbf{R} = (\mathbf{r}^{p(1)}, \mathbf{r}^{p(2)}, \dots, \mathbf{r}^{p(N)})$, where $p(i)$, with $i = 1, 2, \dots, N$, is the particle label in permutation with the i -th particle. If we trace the symmetrized density matrix eq. (5.28) we obtain the partition function for identical Bose particles:

$$Z_{\text{Bose}}(N, V, T) = \frac{1}{N!} \sum_P \int \cdots \int \prod_{j=1}^M d\mathbf{R}_j \prod_{j=1}^M \left\{ \rho^{\text{free}}(\mathbf{R}_j, \mathbf{R}_{j+1}, \tau) \exp[-\tau V(\mathbf{R}_j)] \right\}, \quad (5.29)$$

with the new boundary condition $\mathbf{R}_{M+1} = \mathbf{P}\mathbf{R}_1$. As a consequence of symmetrization the necklaces constituting the polymers are not closed on themselves. The last bead of the i -th world line, \mathbf{r}_M^i , is connected to the first bead of the $p(i)$ -th world-line, $\mathbf{r}_1^{p(i)}$.

At low temperatures, where the thermal wave-length λ_T is comparable to the average inter-particle distance, large permutations cycles form. These are responsible for macroscopic quantum phenomena such as superfluidity and Bose-Einstein condensation.

An exact evaluation of the $N!$ addends summed in eq.(5.29) becomes soon unfeasible by increasing N . Fortunately, all terms are positive definite, then we can still arrange a Monte Carlo procedure for the evaluation of eq. (5.29). If we considered Fermi particles, an additional ‘+’ or ‘-’ sign would appear in front of each term, the former for even permutations, the latter for odd permutations. A Monte Carlo evaluation of the Fermi partition function would lead to an exponentially small signal to noise ratio going to

small T and large N . As a consequence of this *sign problem* the path-integral calculation becomes unfeasible unless one introduces some systematic approximations.

5.1.4 Path sampling methods

In this section we describe the Monte Carlo procedure to sample path-integrals. One has to set a random walk through configuration space. Let $P(X, X')$ be the probability to jump from configuration X to X' . One can prove that if the transition matrix $P(X, X')$ satisfies the *detailed balance condition*:

$$\pi(X) P(X, X') = \pi(X') P(X', X), \quad (5.30)$$

then the random walk samples points with probability $\pi(X)$.

One very flexible algorithm that satisfies eq. (5.30) is the famous *Metropolis algorithm*. This algorithm is divided in two steps. The first is the proposal of a transition from point X to X' with an arbitrary probability $T(X, X')$. The second consists in an acceptance/rejection stage. The proposal is accepted with the probability defined by:

$$A(X, X') = \min(1, \chi(X, X')), \quad (5.31)$$

where

$$\chi(X, X') = \frac{\pi(X') T(X', X)}{\pi(X) T(X, X')}. \quad (5.32)$$

If, for example, we choose to displace one bead, say \mathbf{r}_j^i , to another point, call it $\mathbf{r}_j^{i'}$, that we sample uniformly from a sphere with center in the old position, then one has that $T(X', X) = T(X, X')$ by symmetry and that the probability to accept the move is determined by

$$\chi(X, X') = \frac{\exp\left[-\frac{(\mathbf{r}_{j-1}^i - \mathbf{r}_j^{i'})^2 + (\mathbf{r}_j^{i'} - \mathbf{r}_{j+1}^i)^2}{2\hbar^2\tau/m}\right]}{\exp\left[-\frac{(\mathbf{r}_{j-1}^i - \mathbf{r}_j^i)^2 + (\mathbf{r}_j^i - \mathbf{r}_{j+1}^i)^2}{2\hbar^2\tau/m}\right]} \exp\left[-\tau(V(\mathbf{R}'_j) - V(\mathbf{R}_j))\right]. \quad (5.33)$$

This type of ‘single bead’ move becomes extremely inefficient when the number of time-slices M increases (*critical slowing down*), so one faces ergodicity problems. To increase efficiency one can implement a direct sampling of the kinetic-energy part of the probability distribution for one bead or for a larger piece of a word-line. There are several algorithms that permit drawing a free-particle path (see references). With this type of move rejections are only determined by inter-particle interactions and/or external potentials.

5.1.5 Calculating properties

The expectation value of any operator \hat{O} associated to a physical observable can be written as a path integral in the following form:

$$\bar{O} \equiv \langle O(X) \rangle \equiv \frac{1}{N!} \sum_P \int O(X) \pi(X, \mathbf{P}) dX. \quad (5.34)$$

The energy per particle E/N of a quantum many body system is the expectation value of the Hamiltonian operator \hat{H} divided by the number of particles N . According to its thermodynamic definition we can also obtain E/N through a β -derivative of the partition function Z :

$$\frac{E(N, V, \beta)}{N} = -\frac{1}{NZ} \frac{\partial Z(N, V, \beta)}{\partial \beta}.$$

If we apply this derivative to the symmetrized partition function defined in eq. (5.29) we obtain the following estimator for the energy per particle (called *thermodynamic estimator*):

$$\frac{E_{\text{th}}}{N} = \left\langle \frac{d}{2\tau} - \frac{m}{2(\hbar\tau)^2 MN} \sum_{j=1}^M (\mathbf{R}_j - \mathbf{R}_{j+1})^2 + \frac{1}{MN} \sum_{j=1}^M V(\mathbf{R}_j) \right\rangle. \quad (5.35)$$

5.1.6 Useful references

- A statistical approach to Quantum Mechanics, by M. Creutz and B. Freedman, Annals of Physics 132 (1981) 427.
- A Java demonstration of Path integral Monte Carlo by A. Santamaria can be found at <http://fisteo12.ific.uv.es/~santamar/qapplet/metro.html>. Note that the parameters of the quartic potential can be adjusted interactively.
- D. M. Ceperley, Review of Modern Physics **67**, 279 (1995).

5.2 Diffusion Monte Carlo

Diffusion Monte Carlo (DMC) is a tool to study the ground-state properties of quantum systems. This means that using DMC one can simulate many-body systems at zero temperature. When applied to bosons, DMC provides the exact result for the ground-state energy and for other diagonal properties. By introducing some approximation, one can also treat fermionic systems. One approximation which has proven to be reliable is the so-called *fixed-node approximation*. Similarly, one can extend DMC to study excited states.

DMC is based on the solution of the time-dependent Schrödinger equation written in imaginary time:

$$-\frac{\partial}{\partial \beta} \phi(\mathbf{R}, \beta) = \hat{H} \phi(\mathbf{R}, \beta), \quad (5.36)$$

where $\beta = it/\hbar$. The formal solution of eq. (5.36) is:

$$\phi(\mathbf{R}, \beta) = \exp(-\beta \hat{H}) \phi(\mathbf{R}, 0). \quad (5.37)$$

Let us expand $\phi(\mathbf{R}, \beta)$ on the basis of the eigenstates $\phi_n(\mathbf{R}, \beta)$:

$$\phi(\mathbf{R}, \beta) = \sum_{n=0}^{\infty} c_n \phi_n(\mathbf{R}, \beta) = \sum_{n=0}^{\infty} c_n \phi_n(\mathbf{R}) \exp(-E_n \beta). \quad (5.38)$$

The states ϕ_n are the solution of the time independent Schrödinger equation $\hat{H}\phi_n = E_n\phi_n$ with eigenvalues E_n . We order them in such a way that E_n monotonically increases with the quantum number n . In the long time limit $\beta \rightarrow \infty$ eq. (5.38) reduces to:

$$\phi(\mathbf{R}, \beta) \approx c_0 \phi_0(\mathbf{R}) \exp(-E_0 \beta). \quad (5.39)$$

In other words, the contribution of the ground state dominates the sum in eq. (5.38). States with $n \neq 0$ decay exponentially faster. In the following we will see that by introducing an energy shift we can obtain a normalized wave function.

In the case of Bose systems at zero temperature one can assume, without loss of generality, that $\phi_0(\mathbf{R})$ is real and positive definite¹. Fermi systems and excited states of bosons will be addressed in subsection 5.2.2.

Let us introduce the Green's function of eq. (5.36):

$$\rho(\mathbf{R}, \mathbf{R}', \beta) = \langle \mathbf{R} | \exp(-\beta \hat{H}) | \mathbf{R}' \rangle. \quad (5.40)$$

Notice that $\rho(\mathbf{R}, \mathbf{R}', \beta)$ is equal to the thermal density matrix (5.6). The Green's function permits to write the eq. (5.36) in the integral form:

$$\phi(\mathbf{R}, \beta) = \int \rho(\mathbf{R}, \mathbf{R}', \beta) \phi(\mathbf{R}', 0) d\mathbf{R}'. \quad (5.41)$$

This integral equation may be interpreted as a diffusion process guided by $\rho(\mathbf{R}, \mathbf{R}', \beta)$ from the initial state $\phi(\mathbf{R}', 0)$ to the final state $\phi(\mathbf{R}, \beta)$ at time β .

The evolution during the long time interval β can be generated repeating a large number of short time-steps τ . In the limit $\tau \rightarrow 0$, one can make use of the *primitive approximation* (see section 5.1):

$$\rho(\mathbf{R}_1, \mathbf{R}_3, \tau) \approx \left(\frac{m}{2\pi\hbar^2\tau} \right)^{dN/2} \exp \left[-\frac{(\mathbf{R}_1 - \mathbf{R}_2)^2}{2\hbar^2\tau/m} \right] \exp[-\tau V(\mathbf{R}_2)] \delta(\mathbf{R}_2 - \mathbf{R}_3). \quad (5.42)$$

In a DMC simulation, one treats $\phi(\mathbf{R}, \beta)$ as the density distribution of a large ensemble of equivalent copies of the many-body system, usually called *walkers*. The simulation starts with an arbitrary initial distribution. The population of walkers diffuses according to the Green's function (5.42). The first term corresponds to a free-particle diffusion, which can be implemented by adding to \mathbf{R}_1 a vector whose components are sampled from a gaussian distribution. The second term in eq. (5.42), instead, does not cause displacement of particles. It only determines a change in the probability density. This effect, usually called *branching*, can be implemented by allowing variations in the number of walkers. We have to assign to each walker a number of descendant n_d proportional to the weight $\exp[-\tau(V(\mathbf{R}_2) - E)]$. Notice that we have included the energy shift E , which serves to normalize the density distribution. One could simply set n_d to be equal to the integer number which is closest to w . However, this discretization of the weight w would result in a considerable loss of information. A much more efficient procedure is obtained by calculating n_d according to the following rule:

$$n_d = \text{int}(w + \eta), \quad (5.43)$$

¹If a magnetic field is present the wave function must have an imaginary part also.

where η is a uniform random variable in the range $[0, 1]$, and the function $\text{int}()$ takes the integer part of the argument. In this way one makes use of the full information contained in the signal w . If $n_d > 1$, one has to create $n_d - 1$ identical copies of the walker and include them in the total population. If $n_d = 0$, one has to erase the current walker from the population. The parameter E acts as a normalization factor. It must be adjusted during the simulation in order to maintain the total number of walkers close to an average value, call it n_{ave} . This is an algorithm parameter which has to be optimized. For small values of n_{ave} one has systematic deviations from the exact results. On the other hand, large values of n_{ave} result in computationally demanding simulations.

If we generate a long random walk performing sequentially the two types of update that we have described, the asymptotic distribution $\phi(\mathbf{R}, \beta \rightarrow \infty)$ converges to the exact ground state $\phi_0(\mathbf{R})$.

5.2.1 Importance Sampling

The algorithm described in the previous subsection is extremely inefficient for large particle numbers, especially if the inter-particle interaction is not smooth. The efficiency can be enormously enhanced by using the *importance sampling* technique. To implement this method one has to design a *trial wave function*, call it ϕ_T , that approximately describes the exact ground-state ϕ_0 . For example, in the case of homogeneous liquid or gases an accurate approximation of the ground-state is given by the *Jastrow wave function*:

$$\phi_J(\mathbf{R}) = \prod_{i < j} f_2(|\mathbf{r}_i - \mathbf{r}_j|), \quad (5.44)$$

where the function $f_2(r)$ describes the direct correlation between particles i and j . In dilute systems, like ultracold gases, it can be set equal to the solution of the two-body problem for the relative motion of the pair.

One then solves the modified Schrödinger equation (in imaginary time) for the product $f(\mathbf{R}, \beta) = \phi_T(\mathbf{R})\phi(\mathbf{R}, \beta)$:

$$-\frac{\partial}{\partial \beta} f(\mathbf{R}, \beta) = -\frac{\hbar^2}{2m} \Delta f(\mathbf{R}, \beta) + \frac{\hbar^2}{2m} \nabla (\mathbf{F} f(\mathbf{R}, \beta)) + (E_{\text{loc}}(\mathbf{R}) - E) f(\mathbf{R}, \beta), \quad (5.45)$$

where $\mathbf{F} = \frac{2\nabla\phi_T(\mathbf{R})}{\phi_T(\mathbf{R})}$ is called *pseudo force* and the *local energy* $E_{\text{loc}}(\mathbf{R})$ is defined by:

$$E_{\text{loc}}(\mathbf{R}) = \frac{\hat{H}\phi_T(\mathbf{R})}{\phi_T(\mathbf{R})}. \quad (5.46)$$

The function $f(\mathbf{R}, \beta)$ is interpreted as density distribution of the population of walkers. In the long time limit it converges to the product $\phi_T(\mathbf{R})\phi_0(\mathbf{R})$. It is easy to see that the average of the local energy (5.46) is equal to the ground-state energy. Instead, for observable operators that do not commute with the Hamiltonian, one obtains the *mixed estimator* $\langle \phi_0 | \hat{O} | \phi_T \rangle / \langle \phi_0 | \phi_T \rangle$.²

²For diagonal operators, one can implement exact estimators using the *forward walking* technique (see references).

The diffusion process that solves eq. (5.45) is similar to the one described above. The free-particle diffusion must be implemented in the same way. Between this free-particle diffusion and the branching term, one must introduce an additional update which consists in a drift of particle coordinates guided by the pseudo-force \mathbf{F} :

$$\mathbf{R}_2 = \mathbf{R}_1 + \frac{\hbar^2 \tau}{2m} \mathbf{F}(\mathbf{R}_1). \quad (5.47)$$

This drift guides the walkers in regions with high probability. The branching term has to be implemented similarly to what described before, but substituting the local energy $E_{\text{loc}}(\mathbf{R})$ to the bare potential $V(\mathbf{R})$. In fact, with an accurate choice of the trial wave function ϕ_T , the local energy has small fluctuations. This permits to stabilize the populations of walkers, which, if no importance sampling was implemented, would instead oscillate widely rendering the simulation unfeasible.

5.2.2 Fixed Node Diffusion Monte Carlo

The conclusion that the DMC algorithm samples, after long times, a density distribution proportional to the exact ground state ϕ_0 , is based on the hypothesis that ϕ_0 and ψ_T are not orthogonal. If, instead, they are orthogonal, the asymptotic distribution is proportional to the lowest excited state ϕ_1 not orthogonal to ψ_T . This property is often used to simulate excited states of bosons or the ground state of fermions, which can be considered as the first fully antisymmetric eigenstate of the Hamiltonian. Having to deal with non-positive definite wave functions introduces the well known *sign problem*. Several procedures exist to circumvent this pathology. Here we describe the *fixed-node* approximation. This approximation consists in forcing the ground state of the Fermi system ϕ_F to have the same nodal structure as the trial wave function. It is evident that, if ϕ_F and ψ_T change sign together, the probability distribution is always positive. It can be proven that the fixed-node constraint provides an upper bound to the ground-state energy of fermions. In particular, if the nodes of ψ_T were exact, the FNDMC would provide the exact ground-state energy. In a DMC simulation, the nodal constraint on ϕ_F corresponds to forcing the walkers not to cross the nodal surface.

Just to show an example, we describe now one type of antisymmetric trial wave function which has proven to capture the essential properties of several Fermi systems in the homogeneous normal phase. This is the so-called *Jastrow-Slater* wave function. If we consider a spin-polarized system (all fermions have the same spin-projection) in a box of size L with periodic boundary conditions, the Jastrow-Slater wave function ϕ_{JS} is the product of a Jastrow factor (5.44) and a Slater determinant of plane waves:

$$\phi_{JS}(\mathbf{R}) = \phi_J(\mathbf{R}) \text{Det}_{\alpha,n} [\exp(i\mathbf{k}_\alpha \cdot \mathbf{r}_n)], \quad (5.48)$$

where the index $n = 1, \dots, N$ labels particles and \mathbf{k}_α are the wave vectors compatible with periodic boundary conditions.

Techniques to go beyond the fixed-node approximation exist, but they have not proven to be robust. The sign problem has to be considered still unsolved.

5.2.3 Useful references

- J. Boronat, in *Microscopic approaches to quantum liquids in confined geometries*, chapter 2, ed. by E. Krotscheck and J. Navarro, World Scientific (2002).
- B. L. Hammond, W. A. Lester and Peter James Reynolds, *Monte Carlo methods in Ab Initio quantum chemistry*, World Scientific (1994).
- I. Kosztin, B. Faber and K. Schulten, *Introduction to the Diffusion Monte Carlo Method*, arXiv:physics/9702023.
- M. H. Kalos and P. A. Whitlock, *Monte Carlo methods*, Wiley pub. (1986).

Chapter 6

Electronic structure of molecules and atoms

6.1 Introduction

In this chapter we will discuss the arguably most important quantum many body problem – the electronic structure problem – relevant for almost all properties of matter relevant in our daily life. With $O(10^{23})$ atoms in a typical piece of matter, the exponential scaling of the Hilbert space dimension with the number of particles is a nightmare. In this chapter we will discuss first the exact solution by exact diagonalization of simplified effective models, and then approximate methods that reduce the problem to a polynomial one, typically scaling like $O(N^4)$ and even $O(N)$ in modern codes that aim for a sparse matrix structure. These methods map the problem to a single-particle problem and work only as long as correlations between electrons are weak.

This enormous reduction in complexity is however paid for by a crude approximation of electron correlation effects. This is acceptable for normal metals, band insulators and semi-conductors but fails in materials with strong electron correlations, such as almost all transition metal compounds.

6.2 The electronic structure problem

For many atoms (with the notable exception of Hydrogen and Helium which are so light that quantum effects are important in daily life), the nuclei of atoms are so much heavier than the electrons that we can view them as classical particles and can consider them as stationary for the purpose of calculating the properties of the electrons. Using this Born-Oppenheimer approximation the Hamiltonian operator for the electrons becomes

$$H = \sum_{i=1}^N \left(-\frac{\hbar^2}{2m} \nabla^2 + V(\vec{r}_i) \right) + \sum_{i < j} \frac{e^2}{|\vec{r}_i - \vec{r}_j|} \quad (6.1)$$

where the potential of the M atomic nuclei with charges Z_i at the locations \vec{R}_i is given by

$$V(\vec{r}) = -e^2 \sum_{i=1}^M \frac{Z_i}{|\vec{R}_i - \vec{r}|}. \quad (6.2)$$

The Car-Parinello method for molecular dynamics, which we will discuss later, moves the nuclei classically according to electronic forces that are calculated quantum mechanically.

Using a basis set of L orbital wave functions $\{f_i\}$, the matrix elements of the Hamilton operator (6.1) are

$$t_{ij} = \int d^3\vec{r} f_i^*(\vec{r}) \left(\frac{\hbar^2}{2m} \nabla^2 + V(\vec{r}) \right) f_j(\vec{r}) \quad (6.3)$$

$$V_{ijkl} = e^2 \int d^3\vec{r} \int d^3\vec{r}' f_i^*(\vec{r}) f_j(\vec{r}) \frac{1}{|\vec{r} - \vec{r}'|} f_k^*(\vec{r}') f_l(\vec{r}') \quad (6.4)$$

and the Hamilton operator can be written in second quantized notation as

$$H = \sum_{ij\sigma} t_{ij} a_{i\sigma}^\dagger a_{j\sigma} + \frac{1}{2} \sum_{ijkl\sigma\sigma'} V_{ijkl} a_{i\sigma}^\dagger a_{k\sigma'}^\dagger a_{l\sigma'} a_{j\sigma}. \quad (6.5)$$

6.3 Basis functions

Before attempting to solve the many body problem we will discuss basis sets for single particle wave functions.

6.3.1 The electron gas

For the free electron gas with Hamilton operator

$$H = - \sum_{i=1}^N \frac{\hbar^2}{2m} \nabla^2 + e^2 \sum_{i < j} v_{ee}(\vec{r}_i, \vec{r}_j) \quad (6.6)$$

$$v_{ee}(\vec{r}, \vec{r}') = \frac{1}{|\vec{r} - \vec{r}'|} \quad (6.7)$$

the ideal choice for basis functions are plane waves

$$\psi_{\vec{k}}(\vec{r}) = \exp(-i\vec{k}\vec{r}). \quad (6.8)$$

Such plane wave basis functions are also commonly used for band structure calculations of periodic crystals.

At low temperatures the electron gas forms a Wigner crystal. Then a better choice of basis functions are eigenfunctions of harmonic oscillators centered around the classical equilibrium positions.

6.3.2 Atoms and molecules

Which functions should be used as basis functions for atoms and molecules? We can let ourselves be guided by the exact solution of the Hydrogen atom and use the so-called **Slater-Type-Orbitals** (STO):

$$f_{inlm}(r, \theta, \phi) \propto r^{n-1} e^{-\zeta_i r} Y_{lm}(\theta, \phi). \quad (6.9)$$

These wave functions have the correct asymptotic radial dependence and the correct angular dependence. The values ζ_i are optimized so that the eigenstates of isolated atoms are reproduced as accurately as possible.

The main disadvantage of the STOs becomes apparent when trying to evaluate the matrix elements in equation (6.4) for basis functions centered around two different nuclei at position \vec{R}_A and \vec{R}_B . There we have to evaluate integrals containing terms like

$$\frac{1}{|\vec{r} - \vec{r}'|} e^{-\zeta_i |\vec{r} - \vec{R}_A|} e^{-\zeta_j |\vec{r} - \vec{R}_B|} \quad (6.10)$$

which cannot be solved in any closed form.

The **Gauss-Type-Orbitals** (GTO)

$$f_{ilmn}(\vec{r}) \propto x^l y^m z^n e^{-\zeta_i r^2} \quad (6.11)$$

simplify the evaluation of matrix elements, as Gaussian functions can be integrated easily and the product of Gaussian functions centered at two different nuclei is again a single Gaussian function:

$$e^{-\zeta_i |\vec{r} - \vec{R}_A|^2} e^{-\zeta_j |\vec{r} - \vec{R}_B|^2} = K e^{-\zeta |\vec{r} - \vec{R}|^2} \quad (6.12)$$

with

$$K = e^{-\frac{\zeta_i \zeta_j}{\zeta_i + \zeta_j} |\vec{R}_A - \vec{R}_B|^2} \quad (6.13)$$

$$\zeta = \zeta_i + \zeta_j \quad (6.14)$$

$$\vec{R} = \frac{\zeta_i \vec{R}_A + \zeta_j \vec{R}_B}{\zeta_i + \zeta_j} \quad (6.15)$$

Also the term $\frac{1}{|\vec{r} - \vec{r}'|}$ can be rewritten as an integral over a Gaussian function

$$\frac{1}{|\vec{r} - \vec{r}'|} = \frac{2}{\sqrt{\pi}} \int_0^\infty dt e^{-t^2 (\vec{r} - \vec{r}')^2}. \quad (6.16)$$

and thus all the integrals (6.4) reduce to purely Gaussian integrals which can be performed analytically.

Independent of whether one chooses STOs or GTOs, extra care must be taken to account for the non-orthogonality of these basis functions.

6.3.3 Electrons in solids

Neither plane waves nor localized functions are ideal for electrons in solids. The core electrons are mostly localized and would best be described by localized basis functions as discussed in Sec. 6.3.2. The valence orbitals, on the other hand, overlap to form delocalized bands and are better described by a plane wave basis as in Sec. 6.3.1. More complicated bases sets, like linear augmented plane waves (LAPW), which smoothly cross over from localized wave function behavior near the nuclei to plane waves in the region between the atoms are used for such simulations. It is easy to imagine that a full featured electronic structure code with such basis functions gets very complicated to code.

6.3.4 Other basis sets

There is ongoing development of new basis sets, such as finite element or wavelet based approaches. One key problem for the simulation of large molecules is that since there are $O(L^4)$ integrals of the type (6.4), quantum chemistry calculations typically scale as $O(N^4)$. A big goal is thus to find smart basis sets and truncation schemes to reduce the effort to an approximately $O(N)$ method, since the overlap of basis functions at large distances becomes negligibly small. The group of S. Goedecker in Basel is one of the leading groups in this area, using wavelet basis sets.

6.4 Pseudo-potentials

The electrons in inner, fully occupied shells do not contribute in the chemical bindings. To simplify the calculations they can be replaced by pseudo-potentials, modeling the inner shells. Only the outer shells (including the valence shells) are then modeled using basis functions. The pseudo-potentials are chosen such that calculations for isolated atoms are as accurate as possible.

6.5 Effective models

To understand the properties of these materials the Hamilton operator of the full quantum chemical problem (6.1) is usually simplified to effective models, which still contain the same important features, but which are easier to investigate. They can be used to understand the physics of these materials, but not directly to quantitatively fit experimental measurements.

6.5.1 The tight-binding model

The simplest model is the tight-binding model, which concentrates on the valence bands. All matrix elements t_{ij} in equation (6.3), apart from the ones between nearest neighbor atoms are set to zero. The others are simplified, as in:

$$H = \sum_{\langle i,j \rangle, \sigma} (t_{ij} c_{i,\sigma}^\dagger c_{j,\sigma} + \text{H.c.}) \quad (6.17)$$

This model is easily solvable by Fourier transforming it, as there are no interactions.

6.5.2 The Hubbard model

To include effects of electron correlations, the Hubbard model includes only the often dominant intra-orbital repulsion V_{iiii} of the V_{ijkl} in equation (6.4):

$$H = \sum_{\langle i,j \rangle, \sigma} (t_{ij} c_{i,\sigma}^\dagger c_{j,\sigma} + \text{H.c.}) + \sum_i U_i n_{i,\uparrow} n_{i,\downarrow}. \quad (6.18)$$

The Hubbard model is a long-studied, but except for the 1D case still not completely understood model for correlated electron systems.

In contrast to band insulators, which are insulators because all bands are either completely filled or empty, the Hubbard model at large U is insulating at half filling, when there is one electron per orbital. The reason is the strong Coulomb repulsion U between the electrons, which prohibit any electron movement in the half filled case at low temperatures.

6.5.3 The Heisenberg model

In this insulating state the Hubbard model can be simplified to a *quantum* Heisenberg model, containing exactly one spin per site.

$$H = \sum_{\langle i,j \rangle} J_{ij} \vec{S}_i \vec{S}_j \quad (6.19)$$

For large U/t the perturbation expansion gives $J_{ij} = 2t_{ij}^2(1/U_i + 1/U_j)$. The Heisenberg model is the relevant effective models at temperatures $T \ll t_{ij}, U$ (10^4 K in copper oxides). The derivation will be shown in the lecture.

6.5.4 The t - J model

The t - J model is the effective model for large U at low temperatures away from half-filling. Its Hamiltonian is

$$H = \sum_{\langle i,j \rangle, \sigma} \left[(1 - n_{i,-\sigma}) t_{ij} c_{i,\sigma}^\dagger c_{j,\sigma} (1 - n_{j,-\sigma}) + \text{H.c.} \right] + \sum_{\langle i,j \rangle} J_{ij} (\vec{S}_i \vec{S}_j - n_i n_j / 4). \quad (6.20)$$

As double-occupancy is prohibited in the t - J model there are only three instead of four states per orbital, greatly reducing the Hilbert space size.

6.6 Exact diagonalization

The most accurate method is exact diagonalization of the Hamiltonian matrix using the Lanczos algorithm, discussed in appendix A.2. The size of the Hilbert space of an N -site system (4^N for a Hubbard model , 3^N for a t - J model and $(2S+1)^N$ for a spin- S model) can be reduced by making use of symmetries. Translational symmetries can be

employed by using Bloch waves with fixed momentum as basis states. Conservation of particle number and spin allows to restrict a calculation to subspaces of fixed particle number and magnetization.

As an example we will sketch how to implement exact diagonalization for a simple one-dimensional spinless fermion model with nearest neighbor hopping t and nearest neighbor repulsion V :

$$H = -t \sum_{i=1}^{L-1} (c_i^\dagger c_{i+1} + \text{H.c.}) + V \sum_{i=1}^{L-1} n_i n_{i+1}. \quad (6.21)$$

The first step is to construct a basis set. We describe a basis state using “multi-bit coding”. A many-body state of fermions can be represented as an unsigned integer where bit i set to one corresponds to an occupied site i . For spinful fermions we take either two integers, one for the up and one for the down spins, or two bits per site.

As the Hamiltonian conserves the total particle number we thus want to construct a basis of all states with N particles on L sites (or N bits set to one in L bits). In the code fragment below we use the following variables:

- **states_** is a vector storing the integers whose bit patterns correspond to the basis states. It can be accessed using the following functions:
 - **dimension()** returns the number of basis states.
 - **state(i)** returns the i -th basis state, where i runs from 0 to **dimension()** – 1.
- **index_** is a much larger vector of size 2^L . It is used to obtain the number of a state in the basis, given the integer representation of the state. It can be accessed using the function
 - **index(s)** which returns the index i of the state in the basis, or the largest integer to denote an invalid state, if the bit pattern of the integer does not correspond to a basis state.

Since this vector is very large, it will limit the size of system that can be studied. To save space, the **index_** array could be omitted and the **index(s)** function implemented by a binary search on the **states_** array.

Here is the C++ code for this class:

```
#include <vector>
#include <alps/bitops.h>
#include <limits>
#include <valarray>
#include <cassert>

class FermionBasis {
public:
    typedef unsigned int state_type;
```

```

typedef unsigned int index_type;
FermionBasis (int L, int N);

state_type state(index_type i) const {return states_[i];}
index_type index(state_type s) const {return index_[s];}
unsigned int dimension() const { return states_.size();}

private:
    std::vector<state_type> states_;
    std::vector<index_type> index_;
};


```

In the constructor we build the basis states. For N spinless fermions on L sites the valid basis states are all the ways to place N particles on L sites, which is equivalent to all integers between 0 and $2^L - 1$ that have N bits set. The constructor uses the `alps::popcnt` function of the ALPS library.

```

FermionBasis::FermionBasis(int L, int N)
{
    index_.resize(1<<L); //  $2^L$  entries
    for (state_type s=0;s<index_.size();++s)
        if(alps::popcnt(s)==N) {
            // correct number of particles
            states_.push_back(s);
            index_[s]=states_.size()-1;
        }
        else
            // invalid state
            index_[s]=std::numeric_limits<index_type>::max();
}

```

Finally we want to implement a matrix-vector multiplication $v = Hw$ for our Hamiltonian and derive a Hamiltonian class. We do not want to store the matrix at all, neither in dense nor in sparse form but instead implement a fast function to perform the matrix-vector multiplication on-the-fly.

```

class HamiltonianMatrix : public FermionBasis {
public:
    HamiltonianMatrix(int L, int N, double t, double V)
        : FermionBasis(L,N), t_(t), V_(V), L_(L) {}

    void multiply(std::valarray<double>& v, const std::valarray<double>& w);

private:
    double t_, V_;
    int L_;
};

```

Finally we show the implementation of the matrix-vector multiplication. It might look like magic but we will explain it all in detail during the lecture.

```

void HamiltonianMatrix::multiply(std::valarray<double>& v,
                                  const std::valarray<double>& w)
{
    // check dimensions
    assert(v.size()==dimension());
    assert(w.size()==dimension());

    // do the V-term
    for (int i=0;i<dimension();++i) {
        state_type s = state(i);
        // count number of neighboring fermion pairs
        v[i]=w[i]*V_*alps::popcnt(s&(s>>1));
    }

    // do the t-term
    for (int i=0;i<dimension();++i) {
        state_type s = state(i);
        // inside the chain
        for (int r=0;r<L_-1;++r) {
            state_type shop = s^(3<<r);    // exchange two neighbors
            index_type idx = index(shop); // get the index
            if(idx!=std::numeric_limits<index_type>::max())
                v[idx]+=-t_*w[i];
        }
        // across the boundary
        state_type shop = s^(1|(1<<(L-1))); // exchange the first and last
        index_type idx = index(shop);          // get the index
        if(idx!=std::numeric_limits<index_type>::max())
            // watch out for Fermi sign since we hop over some particles
            v[idx]+=-t*(alps::popcnt(s&((1<<(L-1))-2))%2==0 ? 1 : -1)*w[i];
    }
}

```

This class can now be used with the Lanczos algorithm to calculate the energies and wave functions of the low lying states of the Hamiltonian.

In production codes one uses all symmetries to reduce the dimension of the Hilbert space as much as possible. In this example translational symmetry can be used if periodic boundary conditions are applied. The implementation gets much harder then.

In order to make the implementation of exact diagonalization much easier we have generalized the expression templates technique developed by Todd Veldhuizen for array expression to expressions including quantum operators. Using this expression template

library we can write a multiplication

$$|\psi\rangle = H|\phi\rangle = \left(-t \sum_{i=1}^{L-1} (c_i^\dagger c_{i+1} + \text{H.c.}) + V \sum_i n_i n_{i+1}\right) |\phi\rangle \quad (6.22)$$

simply as:

```
Range i(1,L-1);
psi = sum(i,(-t*(cdag(i)*c(i+1)+HermitianConjugate)+V*n(i)*n(i+1))*phi);
```

The advantage of the above on-the-fly calculation of the matrix in the multiplication routine is that the matrix need not be stored in memory, which is an advantage for the biggest systems where just a few vectors of the Hilbert space will fit into memory.

If one is not as demanding and wants to simulate a slightly smaller system, where the (sparse) matrix can be stored in memory, then a less efficient but more flexible function can be used to create the matrix and store it in memory. Such a program is available through the ALPS project at <http://alps.comp-phys.org/>. It allows to perform the above calculation just by describing the lattice and model in an XML input file.

6.7 The Hartree Fock method

6.7.1 The Hartree-Fock approximation

The Hartree-Fock approximation is based on the assumption of independent electrons. It starts from an ansatz for the N -particle wave function as a Slater determinant of N single-particle wave functions:

$$\Phi(\vec{r}_1, \sigma_1; \dots; \vec{r}_N, \sigma_N) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \phi_1(\vec{r}_1, \sigma_1) & \dots & \phi_N(\vec{r}_1, \sigma_1) \\ \vdots & & \vdots \\ \phi_1(\vec{r}_N, \sigma_N) & \dots & \phi_N(\vec{r}_N, \sigma_N) \end{vmatrix}. \quad (6.23)$$

The orthogonal single particle wave functions ϕ_μ are chosen so that the energy is minimized.

For numerical calculations a finite basis has to be introduced, as discussed in the previous section. Quantum chemists distinguish between the self-consistent-field (SCF) approximation in a finite basis set and the Hartree-Fock (HF) limit, working in a complete basis. In physics both are known as Hartree-Fock approximation.

6.7.2 The Hartree-Fock equations in nonorthogonal basis sets

It will be easiest to perform the derivation of the Hartree-Fock equations in a second quantized notation. To simplify the discussion we assume closed-shell conditions, where each orbital is occupied by both an electron with spin \uparrow and one with spin \downarrow . We start by writing the Hartree Fock wave function (6.23) in second quantized form:

$$|\Phi\rangle = \prod_{\mu,\sigma} c_{\mu\sigma}^\dagger |0\rangle, \quad (6.24)$$

where $c_{\mu\sigma}^\dagger$ creates an electron in the orbital $\phi_\mu(\mathbf{r}, \sigma)$. As these wave functions are orthogonal the $c_{\mu\sigma}^\dagger$ satisfy the usual fermion anticommutation relations. Greek subscripts refer to the Hartree-Fock single particle orbitals and roman subscripts to the single particle basis functions. Next we expand the $c_{\mu\sigma}^\dagger$ in terms of the creation operators $\hat{a}_{n\sigma}^\dagger$ of our finite basis set:

$$c_{\mu\sigma}^\dagger = \sum_{n=1}^L d_{\mu n} \hat{a}_{n\sigma}^\dagger \quad (6.25)$$

and find that

$$a_{j\sigma}|\Phi\rangle = a_{j\sigma} \prod_{\mu, \sigma'} c_{\mu\sigma'}^\dagger |0\rangle = \sum_{\nu} d_{\nu j} \prod_{\mu\sigma' \neq \nu\sigma} c_{\mu\sigma'}^\dagger |0\rangle. \quad (6.26)$$

In order to evaluate the matrix elements $\langle \Phi | H | \Phi \rangle$ of the Hamiltonian (6.5) we introduce the bond-order matrix

$$P_{ij} = \sum_{\sigma} \langle \Phi | a_{i\sigma}^\dagger a_{j\sigma} | \Phi \rangle = 2 \sum_{\nu} d_{\nu i}^* d_{\nu j}, \quad (6.27)$$

where we have made use of the closed-shell conditions to sum over the spin degrees of freedom. The kinetic term of H is now simply $\sum_{ij} P_{ij} t_{ij}$. Next we rewrite the interaction part $\langle \Phi | a_{i\sigma}^\dagger a_{k\sigma'}^\dagger a_{l\sigma'} a_{j\sigma} | \Phi \rangle$ in terms of the P_{ij} . We find that if $\sigma = \sigma'$

$$\langle \Phi | a_{i\sigma}^\dagger a_{k\sigma}^\dagger a_{l\sigma} a_{j\sigma} | \Phi \rangle = \langle \Phi | a_{i\sigma}^\dagger a_{j\sigma} | \Phi \rangle \langle \Phi | a_{k\sigma}^\dagger a_{l\sigma} | \Phi \rangle - \langle \Phi | a_{i\sigma}^\dagger a_{l\sigma} | \Phi \rangle \langle \Phi | a_{k\sigma}^\dagger a_{j\sigma} | \Phi \rangle \quad (6.28)$$

and if $\sigma \neq \sigma'$:

$$\langle \Phi | a_{i\sigma}^\dagger a_{k\sigma'}^\dagger a_{l\sigma'} a_{j\sigma} | \Phi \rangle = \langle \Phi | a_{i\sigma}^\dagger a_{j\sigma} | \Phi \rangle \langle \Phi | a_{k\sigma'}^\dagger a_{l\sigma'} | \Phi \rangle \quad (6.29)$$

Then the energy is (again summing over the spin degrees of freedom):

$$E_0 = \sum_{ij} t_{ij} P_{ij} + \frac{1}{2} \sum_{ijkl} \left(V_{ijkl} - \frac{1}{2} V_{ilkj} \right) P_{ij} P_{kl}. \quad (6.30)$$

We now need to minimize the energy E_0 under the condition that the $|\phi_\mu\rangle$ are normalized:

$$1 = \langle \phi_\mu | \phi_\mu \rangle = \sum_{i,j} d_{\mu i}^* d_{\mu j} S_{ij}. \quad (6.31)$$

Using Lagrange multipliers to enforce this constraint we have to minimize

$$\sum_{ij} t_{ij} P_{ij} + \frac{1}{2} \sum_{ijkl} \left(V_{ijkl} - \frac{1}{2} V_{ilkj} \right) P_{ij} P_{kl} - \sum_{\mu} \epsilon_{\mu} \sum_{i,j} d_{\mu i}^* d_{\mu j} S_{ij} \quad (6.32)$$

Setting the derivative with respect to $d_{\mu i}$ to zero we end up with the Hartree-Fock equations for a finite basis set:

$$\sum_{j=1}^L (f_{ij} - \epsilon_{\mu} S_{ij}) d_{\mu j} = 0, \quad (6.33)$$

where

$$f_{ij} = t_{ij} + \sum_{kl} \left(V_{ijkl} - \frac{1}{2} V_{ilkj} \right) P_{kl}. \quad (6.34)$$

This is again a generalized eigenvalue problem of the form $Ax = \lambda Bx$ and looks like a one-particle Schrödinger equation. However, since the potential depends on the solution it is a nonlinear and not a linear eigenvalue problem. The equation is solved iteratively, always using the new solution for the potential, until convergence to a fixed point is achieved.

The eigenvalues ϵ_μ of f do not directly correspond to energies of the orbitals, as the Fock operator counts the V -terms twice. Thus we obtain the total ground state energy from the Fock operator eigenvalues by subtracting the double counted part:

$$E_0 = \sum_{\mu=1}^N \epsilon_\mu - \frac{1}{2} \sum_{ijkl} \left(V_{ijkl} - \frac{1}{2} V_{ilkj} \right) P_{ij} P_{kl} \quad (6.35)$$

6.7.3 Configuration-Interaction

The approximations used in Hartree-Fock and density functional methods are based on non-interacting electron pictures. They do not treat correlations and interactions between electrons correctly. To improve these methods, and to allow the calculation of excited states, often the “configuration-interaction” (CI) method is used.

Starting from the Hartree-Fock ground state

$$|\psi_{HF}\rangle = \prod_{\mu=1}^N c_\mu^\dagger |0\rangle \quad (6.36)$$

one or two of the c_μ^\dagger are replaced by other orbitals c_i^\dagger :

$$|\psi_0\rangle = \left(1 + \sum_{i,\mu} \alpha_\mu^i c_i^\dagger c_\mu + \sum_{i < j, \mu < \nu} \alpha_{\mu\nu}^{ij} c_i^\dagger c_j^\dagger c_\mu c_\nu \right) |\psi_{HF}\rangle. \quad (6.37)$$

The energies are then minimized using this variational ansatz. In a problem with N occupied and M empty orbitals this leads to a matrix eigenvalue problem with dimension $1 + NM + N^2 M^2$. Using the Lanczos algorithm the low lying eigenstates can then be calculated in $O((N + M)^2)$ steps.

Further improvements are possible by allowing more than only double-substitutions. The optimal method treats the full quantum problem of dimension $(N + M)!/N!M!$. Quantum chemists call this method “full-CI”. Physicists simplify the Hamilton operator slightly to obtain simpler models with fewer matrix elements, and call that method “exact diagonalization”. This method will be discussed later in the course.

6.8 Density functional theory

Another commonly used method, for which the Nobel prize in chemistry was awarded to Walter Kohn, is the density functional theory. In density functional theory the many-body wave function living in \mathbb{R}^{3N} is replaced by the electron density, which lives just in \mathbb{R}^3 . Density functional theory again reduces the many body problem to a one-dimensional problem. In contrast to Hartree-Fock theory it has the advantage that it

could – in principle – be exact if there were not the small problem of the unknown exchange-correlation functional.

It is based on two fundamental theorems by Hohenberg and Kohn. The first theorem states that the ground state energy E_0 of an electronic system in an external potential V is a functional of the electron density $\rho(\vec{r})$:

$$E_0 = E[\rho] = \int d^3\vec{r} V(\vec{r})\rho(\vec{r}) + F[\rho], \quad (6.38)$$

with a universal functional F . The second theorem states that the density of the ground state wave function minimizes this functional. The proof of both theorems will be shown in the lecture.

These theorems make our life very easy: we only have to minimize the energy functional and we obtain both the ground state energy and the electron density in the ground state – and everything is exact!

The problem is that, while the functional F is universal, it is also unknown! Thus we need to find good approximations for the functional. One usually starts from the ansatz:

$$F[\rho] = E_h[\rho] + E_k[\rho] + E_{xc}[\rho]. \quad (6.39)$$

The Hartree-term E_h given by the Coulomb repulsion between two electrons:

$$E_h[\rho] = \frac{e^2}{2} \int d^3\vec{r} d^3\vec{r}' \frac{\rho(\vec{r})\rho(\vec{r}')}{|\vec{r} - \vec{r}'|}. \quad (6.40)$$

The kinetic energy $E_k[\rho]$ is that of a non-interacting electron gas with the same density. The exchange- and correlation term $E_{xc}[\rho]$ contains the remaining unknown contribution, which we will discuss a bit later.

To calculate the ground state density we have to minimize this energy, solving the variational problem

$$0 = \delta E[\rho] = \int d^3\vec{r} \delta\rho(\vec{r}) \left(V(\vec{r}) + e^2 \int d^3\vec{r}' \frac{\rho(\vec{r}')}{|\vec{r} - \vec{r}'|} + \frac{\delta E_h[\rho]}{\delta\rho(\vec{r})} + \frac{\delta E_{xc}[\rho]}{\delta\rho(\vec{r})} \right) \quad (6.41)$$

0 subject to the constraint that the total electron number to be conserved

$$\int d^3\vec{r} \delta\rho(\vec{r}) = 0. \quad (6.42)$$

Comparing this variational equation to the one for noninteracting system

$$\left(-\frac{1}{2m} \nabla^2 + V_{eff}(\vec{r}) \right) \phi_\mu(\vec{r}) = \epsilon_\mu \phi_\mu(\vec{r}), \quad (6.43)$$

we realize that they are the same if we define the potential of the non-interacting system as

$$V_{eff}(\vec{r}) = V(\vec{r}) + e^2 \int d^3\vec{r}' \frac{\rho(\vec{r}')}{|\vec{r} - \vec{r}'|} + v_{xc}(\vec{r}), \quad (6.44)$$

where the exchange-correlation potential is defined by

$$v_{xc}(\vec{r}) = \frac{\delta E_{xc}[\rho]}{\delta\rho(\vec{r})}. \quad (6.45)$$

The form (6.43) arises because we have separated the kinetic energy of the non-interacting electron system from the functional. The variation of this kinetic energy just gives the kinetic term of this Schrödinger-like equation.

The non-linear equation is again solved iteratively, making an ansatz using $N/2$ normalized single-electron wave functions, which we occupy with spin \uparrow and spin \downarrow electrons to get the electron density.

$$\rho(\vec{r}) = 2 \sum_{\mu=1}^{N/2} |\phi_{\mu}(\vec{r})|^2, \quad (6.46)$$

6.8.1 Local Density Approximation

Apart from the restricted basis set everything was exact up to this point. As the functional $E_{xc}[\rho]$ and thus the potential $v_{xc}(\vec{r})$ is not known, we need to introduce approximations.

The simplest approximation is the “local density approximation” (LDA), which replaces v_{xc} by that of a uniform electron gas with the same density. Instead of taking a functional $E[\rho](\vec{r})$ which could be a function of $\rho(\vec{r}), \nabla\rho(\vec{r}), \nabla\nabla\rho(\vec{r}), \dots$ we ignore all the gradients and just take the local density

$$E_{xc}[\rho](r) = E_{\text{LDA}}(\rho(r)); \quad (6.47)$$

Defining

$$r_s^{-1} = a_B \left(\frac{4\pi}{3} \rho \right)^{1/3} \quad (6.48)$$

the exchange correlation potential is

$$v_{xc} = -\frac{e^2}{a_B} \left(\frac{3}{2\pi} \right)^{2/3} \frac{1}{r_s} [1 + 0.0545 r_s \ln(1 + 11.4/r_s)] \quad (6.49)$$

where the first part corresponds to uncorrelated electrons and the last factor is a correlation correction determined by fitting to quantum Monte Carlo (QMC) simulations of an electron gas.

6.8.2 Improved approximations

Improvements over the LDA have been an intense field of research in quantum chemistry. I will just mention two improvements. The “local spin density approximation” (LSDA) uses separate densities for electrons with spin \uparrow and \downarrow . The “generalized gradient approximation” (GGA) and its variants use functionals depending not only on the density, but also on its derivatives.

6.9 Car-Parinello molecular dynamics

In the lecture on “Computational Statistical Physics” you have learned about the molecular dynamics method, in which atoms move on classical trajectories under forces, such

as those from the Lennard-Jones potential, which have been previously calculated in quantum mechanical simulations. It would be nicer, and more accurate, to use a full quantum mechanical force calculation at every time step instead of using such static forces that have been extracted from previous simulations.

Roberto Car (currently in Princeton) and Michele Parrinello (currently at ETH) have combined density functional theory with molecular dynamics to do just that. Their method, Car-Parrinello molecular dynamics (CPMD) allows much better simulations of molecular vibration spectra and of chemical reactions.

The atomic nuclei are propagated using classical molecular dynamics, but the electronic forces which move them are estimated using density functional theory:

$$M_n \frac{d^2 \vec{R}_n}{dt^2} = -\frac{\partial E[\rho(\vec{r}, t), \vec{R}_n]}{\partial \vec{R}_n}. \quad (6.50)$$

Here M_n and \vec{R}_n are the masses and locations of the atomic nuclei.

As the solution of the full electronic problem at every time step is a very time consuming task we do not want to perform it all the time from scratch. Instead CPMD uses the previous values of the noninteracting electron wave functions $\{\phi_\mu\}$ of the DFT calculation (6.43) [don't confuse it with the Hartee-Fock orbitals!] and evolves them to the ground state for the current positions of the nuclei by an artificial molecular dynamics evolution. Hence both the nuclei $\{\vec{R}_n\}$ and the wave functions $\{\phi_\mu\}$ evolve in the same molecular dynamics scheme. The electronic degrees of freedoms are updated using an artificial dynamics:

$$m \frac{d^2 \phi_\mu(\vec{r}, t)}{dt^2} = -\frac{1}{2} \frac{\delta E[\rho(\vec{r}, t), \vec{R}_n]}{\delta \phi_\mu^\dagger(\vec{r}, t)} + \sum_\nu \Lambda_{\mu\nu} \phi_\nu(\vec{r}, t), \quad (6.51)$$

where m is an artificial mass that needs to be chosen much lighter than the nuclear masses so that the electronic structure adapts quickly to the move of the nuclei. The Lagrange multipliers $\Lambda_{\mu\nu}$ need to be chose to ensure proper orthonormalization of the wave functions.

Since the exact form of the artifical dynamics of the electronic structure does not matter, we can evolve the expansion coefficients $d_{\mu n}$ of an expansion in terms of the basis functions as in equation (6.25) instead of evolving the wave functions. This gives the equations of motion

$$m \frac{d^2 d_{\mu n}}{dt^2} = -\frac{\partial E}{\partial d_{\mu n}} + \sum_\nu \Lambda_{\mu\nu} \sum_l S_{nl} d_{\nu l} \quad (6.52)$$

There are various algorithms to determine the $\Lambda_{\mu\nu}$ so that the wave functions stay orthonormal. We refer to text books and special lectures on CPMD for details.

6.10 Program packages

As the model Hamiltonian and the types of basis sets are essentially the same for all quantum chemistry applications flexible program packages have been written. There is thus usually no need to write your own programs – unless you want to implement a new algorithm.

Appendix A

Numerical methods

A.1 Numerical root solvers

The purpose of a root solver is to find a solution (a root) to the equation

$$f(x) = 0, \quad (\text{A.1})$$

or in general to a multi-dimensional equation

$$\vec{f}(\vec{x}) = 0. \quad (\text{A.2})$$

Numerical root solvers should be well known from the numerics courses and we will just review three simple root solvers here. Keep in mind that in any serious calculation it is usually best to use a well optimized and tested library function over a hand-coded root solver.

A.1.1 The Newton and secant methods

The Newton method is one of best known root solvers, however it is not guaranteed to converge. The key idea is to start from a guess x_0 , linearize the equation around that guess

$$f(x_0) + (x - x_0)f'(x_0) = 0 \quad (\text{A.3})$$

and solve this linearized equation to obtain a better estimate x_1 . Iterating this procedure we obtain the **Newton method**:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \quad (\text{A.4})$$

If the derivative f' is not known analytically, as is the case in our shooting problems, we can estimate it from the difference of the last two points:

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} \quad (\text{A.5})$$

Substituting this into the Newton method (A.4) we obtain the **secant method**:

$$x_{n+1} = x_n - (x_n - x_{n-1}) \frac{f(x_n)}{f(x_n) - f(x_{n-1})}. \quad (\text{A.6})$$

The Newton method can easily be generalized to higher dimensional equations, by defining the matrix of derivatives

$$A_{ij}(\vec{x}) = \frac{\partial f_i(\vec{x})}{\partial x_j} \quad (\text{A.7})$$

to obtain the **higher dimensional Newton method**

$$\vec{x}_{n+1} = \vec{x}_n - A^{-1}\vec{f}(\vec{x}) \quad (\text{A.8})$$

If the derivatives $A_{ij}(\vec{x})$ are not known analytically they can be estimated through finite differences:

$$A_{ij}(\vec{x}) = \frac{f_i(\vec{x} + h_j \vec{e}_j) - f_i(\vec{x})}{h_j} \quad \text{with} \quad h_j \approx x_j \sqrt{\varepsilon} \quad (\text{A.9})$$

where ε is the machine precision (about 10^{-16} for double precision floating point numbers on most machines).

A.1.2 The bisection method and regula falsi

Both the bisection method and the regula falsi require two starting values x_0 and x_1 surrounding the root, with $f(x_0) < 0$ and $f(x_1) > 0$ so that under the assumption of a continuous function f there exists at least one root between x_0 and x_1 .

The **bisection method** performs the following iteration

1. define a mid-point $x_m = (x_0 + x_1)/2$.
2. if $\text{sign}f(x_m) = \text{sign}f(x_0)$ replace $x_0 \leftarrow x_m$ otherwise replace $x_1 \leftarrow x_m$

until a root is found.

The **regula falsi** works in a similar fashion:

1. estimate the function f by a straight line from x_0 to x_1 and calculate the root of this linearized function: $x_2 = (f(x_0)x_1 - f(x_1)x_0)/(f(x_1) - f(x_0))$
2. if $\text{sign}f(x_2) = \text{sign}f(x_0)$ replace $x_0 \leftarrow x_2$ otherwise replace $x_1 \leftarrow x_2$

In contrast to the Newton method, both of these two methods will always find a root.

A.1.3 Optimizing a function

These root solvers can also be used for finding an extremum (minimum or maximum) of a function $f(\vec{x})$, by looking a root of

$$\nabla f(\vec{x}) = 0. \quad (\text{A.10})$$

While this is efficient for one-dimensional problems, but better algorithms exist.

In the following discussion we assume, without loss of generality, that we want to minimize a function. The simplest algorithm for a multi-dimensional optimization is

steepest descent, which always looks for a minimum along the direction of steepest gradient: starting from an initial guess \vec{x}_n a one-dimensional minimization is applied to determine the value of λ which minimizes

$$f(\vec{x}_n + \lambda \nabla f(\vec{x}_n)) \quad (\text{A.11})$$

and then the next guess \vec{x}_{n+1} is determined as

$$\vec{x}_{n+1} = \vec{x}_n + \lambda \nabla f(\vec{x}_n) \quad (\text{A.12})$$

While this method is simple it can be very inefficient if the “landscape” of the function f resembles a long and narrow valley: the one-dimensional minimization will mainly improve the estimate transverse to the valley but takes a long time to traverse down the valley to the minimum. A better method is the **conjugate gradient** algorithm which approximates the function locally by a paraboloid and uses the minimum of this paraboloid as the next guess. This algorithm can find the minimum of a long and narrow parabolic valley in one iteration! For this and other, even better, algorithms we recommend the use of **library functions**.

One final word of warning is that all of these minimizers will only find a **local minimum**. Whether this local minimum is also the global minimum can never be decided by purely numerically. A necessary but never sufficient check is thus to start the minimization not only from one initial guess but to try many initial points and check for consistency in the minimum found.

A.2 The Lanczos algorithm

Sparse matrices with only $O(N)$ non-zero elements are very common in scientific simulations. We have already encountered them in the winter semester when we discretized partial differential equations. Now we have reduced the transfer matrix of the Ising model to a sparse matrix product. We will later see that also the quantum mechanical Hamilton operators in lattice models are sparse.

The importance of sparsity becomes obvious when considering the cost of matrix operations as listed in table A.1. For large N the sparsity leads to memory and time savings of several orders of magnitude.

Here we will discuss the iterative calculation of a few of the extreme eigenvalues of a matrix by the Lanczos algorithm. Similar methods can be used to solve sparse linear systems of equations.

To motivate the Lanczos algorithms we will first take a look at the power method for a matrix A . Starting from a random initial vector u_1 we calculate the sequence

$$u_{n+1} = \frac{Au_n}{\|Au_n\|}, \quad (\text{A.13})$$

which converges to the eigenvector of the largest eigenvalue of the matrix A . The Lanczos algorithm optimizes this crude power method.

Table A.1: Time and memory complexity for operations on sparse and dense $N \times N$ matrices

operation	time	memory
storage	—	N^2
dense matrix	—	$O(N)$
sparse matrix	—	$O(N)$
matrix-vector multiplication		
dense matrix	$O(N^2)$	$O(N^2)$
sparse matrix	$O(N)$	$O(N)$
matrix-matrix multiplication		
dense matrix	$O(N^{\ln 7 / \ln 2})$	$O(N^2)$
sparse matrix	$O(N) \dots O(N^2)$	$O(N) \dots O(N^2)$
all eigen values and vectors		
dense matrix	$O(N^3)$	$O(N^2)$
sparse matrix (iterative)	$O(N^2)$	$O(N^2)$
some eigen values and vectors		
dense matrix (iterative)	$O(N^2)$	$O(N^2)$
sparse matrix (iterative)	$O(N)$	$O(N)$

Lanczos iterations

The Lanczos algorithm builds a basis $\{v_1, v_2, \dots, v_M\}$ for the Krylov-subspace $K_M = \text{span}\{u_1, u_2, \dots, u_M\}$, which is constructed by M iterations of equation (A.13). This is done by the following iterations:

$$\beta_{n+1} v_{n+1} = Av_n - \alpha_n v_n - \beta_n v_{n-1}, \quad (\text{A.14})$$

where

$$\alpha_n = v_n^\dagger A v_n, \quad \beta_n = |v_n^\dagger A v_{n-1}|. \quad (\text{A.15})$$

As the orthogonality condition

$$v_i^\dagger v_j = \delta_{ij} \quad (\text{A.16})$$

does not determine the phases of the basis vectors, the β_i can be chosen to be real and positive. As can be seen, we only need to keep three vectors of size N in memory, which makes the Lanczos algorithm very efficient, when compared to dense matrix eigensolvers which require storage of order N^2 .

In the Krylov basis the matrix A is tridiagonal

$$T^{(n)} := \begin{bmatrix} \alpha_1 & \beta_2 & 0 & \cdots & 0 \\ \beta_2 & \alpha_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \beta_n \\ 0 & \cdots & 0 & \beta_n & \alpha_n \end{bmatrix}. \quad (\text{A.17})$$

The eigenvalues $\{\tau_1, \dots, \tau_M\}$ of T are good approximations of the eigenvalues of A . The extreme eigenvalues converge very fast. Thus $M \ll N$ iterations are sufficient to obtain the extreme eigenvalues.

Eigenvectors

It is no problem to compute the eigenvectors of T . They are however given in the Krylov basis $\{v_1, v_2, \dots, v_M\}$. To obtain the eigenvectors in the original basis we need to perform a basis transformation.

Due to memory constraints we usually do not store all the v_i , but only the last three vectors. To transform the eigenvector to the original basis we have to do the Lanczos iterations a second time. Starting from the same initial vector v_1 we construct the vectors v_i iteratively and perform the basis transformation as we go along.

Roundoff errors and ghosts

In exact arithmetic the vectors $\{v_i\}$ are orthogonal and the Lanczos iterations stop after at most $N - 1$ steps. The eigenvalues of T are then the exact eigenvalues of A .

Roundoff errors in finite precision cause a loss of orthogonality. There are two ways to deal with that:

- Reorthogonalization of the vectors after every step. This requires storing all of the vectors $\{v_i\}$ and is memory intensive.
- Control of the effects of roundoff.

We will discuss the second solution as it is faster and needs less memory. The main effect of roundoff errors is that the matrix T contains extra spurious eigenvalues, called “ghosts”. These ghosts are not real eigenvalues of A . However they converge towards real eigenvalues of A over time and increase their multiplicities.

A simple criterion distinguishes ghosts from real eigenvalues. Ghosts are caused by roundoff errors. Thus they do not depend on the starting vector v_1 . As a consequence these ghosts are also eigenvalues of the matrix \tilde{T} , which can be obtained from T by deleting the first row and column:

$$\tilde{T}^{(n)} := \begin{bmatrix} \alpha_2 & \beta_3 & 0 & \cdots & 0 \\ \beta_3 & \alpha_3 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \beta_n \\ 0 & \cdots & 0 & \beta_n & \alpha_n \end{bmatrix}. \quad (\text{A.18})$$

From these arguments we derive the following heuristic criterion to distinguish ghosts from real eigenvalues:

- All multiple eigenvalues are real, but their multiplicities might be too large.
- All single eigenvalues of T which are *not* eigenvalues of \tilde{T} are also real.

Numerically stable and efficient implementations of the Lanczos algorithm can be obtained from netlib or from <http://www.comp-physics.org/software/ietl/>.

2. MONTE CARLO INTEGRATION

In thermodynamics, as in many other fields of physics, often very high dimensional integrals have to be evaluated. Even in a classical N -body simulation the phase space has dimension $6N$, as there are three coordinates each for the location and position of each particle. In a quantum mechanical problem of N particles the phase space is even exponentially large as a function of N .

2.1. Deterministic integration methods

A Riemannian integral $f(x)$ over an interval $[a, b]$ can be evaluated by replacing it by a finite sum:

$$\int_a^b f(x)dx = \sum_{i=1}^N f(a + i\Delta x)\Delta x + O(\Delta x^2), \quad (1)$$

where $\Delta x = (a - b)/N$. The discretization error decreases as $1/N$ for this simple formula. Better approximations are the trapezoidal rule

$$\int_a^b f(x)dx = \Delta x \left[\frac{1}{2}f(a) + \sum_{i=1}^{N-1} f(a + i\Delta x) + \frac{1}{2}f(b) \right] + O(\Delta x^2), \quad (2)$$

or the Simpson rule

$$\begin{aligned} \int_a^b f(x)dx &= \frac{\Delta x}{3} \left[f(a) + \sum_{i=1}^{N/2} 4f(a + (2i-1)\Delta x) \right. \\ &\quad \left. + \sum_{i=1}^{N/2-1} 2f(a + 2i\Delta x) + f(b) \right] + O(\Delta x^4), \end{aligned} \quad (3)$$

which scales like N^{-4} .

For more elaborate schemes like the Romberg method or Gaussian integration we refer to textbooks.

In higher dimensions the convergence is much slower though. With N points in d dimensions the linear distance between two points scales only as $N^{-1/d}$. Thus the Simpson rule in d dimensions converges only as $N^{-4/d}$, which is very slow for large d . The solution are Monte Carlo integrators.

2.2. Monte Carlo integrators

With randomly chosen points the convergence does not depend on dimensionality. Using N randomly chosen points \mathbf{x}_i the integral can be approximated by

$$\frac{1}{\Omega} \int f(\mathbf{x})d\mathbf{x} \approx \bar{f} := \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i), \quad (4)$$

where $\Omega := \int d\mathbf{x}$ is the integration volume. As we saw in the previous chapter the errors of such a Monte Carlo estimate the errors scale as $N^{-1/2}$. In $d \geq 9$ dimensions Monte Carlo methods are thus preferable to a Simpson rule.

2.2.1. Importance Sampling

This simple Monte Carlo integration is however not the ideal method. The reason is the variance of the function

$$\text{Var}f = \Omega^{-1} \int f(\mathbf{x})^2 d\mathbf{x} - \left[\Omega^{-1} \int f(\mathbf{x}) d\mathbf{x} \right]^2 \approx \frac{N}{N-1} (\bar{f}^2 - \bar{f}^2). \quad (5)$$

The error of the Monte Carlo simulation is

$$\Delta = \sqrt{\frac{\text{Var}f}{N}} \approx \sqrt{\frac{\bar{f}^2 - \bar{f}^2}{N-1}}. \quad (6)$$

In phase space integrals the function is often strongly peaked in a small region of phase space and has a large variance. The solution to this problem is “importance sampling”, where the points \mathbf{x}_i are chosen not uniformly but according to a probability distribution $p(\mathbf{x})$ with

$$\int p(\mathbf{x}) d\mathbf{x} = 1. \quad (7)$$

Using these p -distributed random points the sampling is done according to

$$\langle f \rangle = \Omega^{-1} \int A(\mathbf{x}) d\mathbf{x} = \Omega^{-1} \int \frac{f(\mathbf{x})}{p(\mathbf{x})} p(\mathbf{x}) d\mathbf{x} \approx \frac{1}{N} \sum_{i=1}^N \frac{f(\mathbf{x}_i)}{p(\mathbf{x}_i)} \quad (8)$$

and the error is

$$\Delta = \sqrt{\frac{\text{Var}f/p}{N}}. \quad (9)$$

It is ideal to choose the distribution function p as similar to f as possible. Then the ratio f/p is nearly constant and the variance small.

As an example, the function $f(x) = \exp(-x^2)$ is much better integrated using exponentially distributed random numbers with $p(x) = \exp(-\lambda x)$ instead of uniformly distributed random numbers.

A natural choice for the weighting function p is often given in the case of phase space integrals or sums, where an observable O is averaged over all configurations \mathbf{x} in phase space where the probability of a configuration is $p(\mathbf{x})$. The phase space average $\langle O \rangle$ is:

$$\langle O \rangle = \frac{\int O(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}}{\int p(\mathbf{x}) d\mathbf{x}}. \quad (10)$$

Here we will want to do importance sampling with the distribution $p(\mathbf{x})$.

2.3. Markov chains and the Metropolis algorithm

In general problems with arbitrary distributions p it will not be possible to create p -distributed configuration from scratch. Instead a Markov process can be used.

Starting from an initial point \mathbf{x}_0 a Markov chain of states is generated:

$$\mathbf{x}_0 \rightarrow \mathbf{x}_1 \rightarrow \mathbf{x}_2 \rightarrow \dots \rightarrow \mathbf{x}_n \rightarrow \mathbf{x}_{n+1} \rightarrow \dots \quad (11)$$

A transition matrix $W_{\mathbf{xy}}$ gives the transition probabilities of going from state \mathbf{x} to state \mathbf{y} in one step of the Markov process. As the sum of probabilities of going from state \mathbf{x} to any other state is one, the columns of the matrix W are normalized:

$$\sum_{\mathbf{y}} W_{\mathbf{xy}} = 1 \quad (12)$$

A consequence is that the Markov process conserves the total probability. Another consequence is that the largest eigenvalue of the transition matrix W is 1 and the corresponding eigenvector with only positive entries is the equilibrium distribution which is reached after a large number of Markov steps.

We want to determine the transition matrix W so that we asymptotically reach the desired probability $p_{\mathbf{x}}$ for a configuration i . A set of sufficient conditions is:

1. **Ergodicity:** It has to be possible to reach any configuration \mathbf{x} from any other configuration \mathbf{y} in a finite number of Markov steps. This means that for all \mathbf{x} and \mathbf{y} there exists a positive integer $n_0 < \infty$ such that $\forall n > n_0 : (W^n)_{\mathbf{xy}} \neq 0$.
2. **Detailed balance:** The probability distribution $p_{\mathbf{x}}^{(n)}$ changes at each step of the Markov process:

$$\sum_{\mathbf{x}} p_{\mathbf{x}}^{(n)} W_{\mathbf{xy}} = p_{\mathbf{y}}^{(n+1)}. \quad (13)$$

but converges to the equilibrium distribution $p_{\mathbf{x}}$. This equilibrium distribution $p_{\mathbf{x}}$ is an eigenvector with left eigenvalue 1 and the equilibrium condition

$$\sum_{\mathbf{x}} p_{\mathbf{x}} W_{\mathbf{xy}} = p_{\mathbf{y}} \quad (14)$$

must be fulfilled. It is easy to see that the detailed balance condition

$$\frac{W_{\mathbf{xy}}}{W_{\mathbf{yx}}} = \frac{p_{\mathbf{y}}}{p_{\mathbf{x}}} \quad (15)$$

is sufficient.

Having defined W , we now have to construct it explicitly. Let $W_{\mathbf{xy}}^0$ be the probability of proposing a transition from \mathbf{x} to \mathbf{y} and $a_{\mathbf{xy}}$ the probability of accepting it. $1 - a_{\mathbf{xy}}$ corresponds to the probability of rejecting the move. With $W_{\mathbf{xy}}^0$ and $a_{\mathbf{xy}}$ we build $W_{\mathbf{xy}}$ with:

$$W_{\mathbf{xy}} = \begin{cases} W_{\mathbf{xy}}^0 a_{\mathbf{xy}} & \text{if } y \neq x \\ W_{\mathbf{xx}}^0 + \sum_{z \neq x} W_{\mathbf{xz}}^0 (1 - a_{\mathbf{xz}}) & \text{if } y = x \end{cases} \quad (16)$$

Detailed balance can be satisfied by setting

$$a_{\mathbf{xy}} = F \left(\frac{W_{\mathbf{yx}}^0 P_y}{W_{\mathbf{xy}}^0 P_x} \right). \quad (17)$$

Since

$$a_{\mathbf{yx}} = F \left(\frac{W_{\mathbf{xy}}^0 P_x}{W_{\mathbf{yx}}^0 P_y} \right) = F \left(\frac{1}{\frac{W_{\mathbf{yx}}^0 P_y}{W_{\mathbf{xy}}^0 P_x}} \right), \quad (18)$$

the detailed balance condition reduces to:

$$\frac{F(R)}{F(1/R)} = R \text{ where } R = \frac{W_{\mathbf{yx}}^0 P_y}{W_{\mathbf{xy}}^0 P_x}. \quad (19)$$

There are many possible choices. The Metropolis algorithm [9] is based on the choice:

$$F(R) = \min(R, 1). \quad (20)$$

Thus, one proposes a transition from x to y and accepts it with probability $R = \frac{T_{x,y}^0 P_y}{W_{\mathbf{xy}}^0 P_x}$.

This is implemented by picking a random number r in the interval $[0 : 1]$. If $r < R$ ($r > R$) one accepts (rejects) the move. Alternative choices of $F(R)$ are for example:

$$F(R) = \frac{R}{1+R} \quad (21)$$

which is referred to as the heat bath method.

The observable O may now be estimated with:

$$\langle O \rangle_P \approx \frac{1}{N} \sum_{t=1}^N O(x_t). \quad (22)$$

2.4. Autocorrelations, equilibration and Monte Carlo error estimates

2.4.1. Autocorrelation effects

In the determination of statistical errors of the Monte Carlo estimates we have to take into account correlations between successive points \mathbf{x}_i in the Markov chain. These correlations between configurations manifest themselves in correlations between the measurements of a quantity O measured in the Monte Carlo process. Denote by O_t the measurement of the observable O evaluated at the t -th Monte Carlo point \mathbf{x}_t . The autocorrelations decay exponentially for large time differences Δ :

$$\langle O_t O_{t+\Delta} \rangle - \langle O \rangle^2 \propto \exp(-\Delta/\tau_O^{(exp)}) \quad (23)$$

Note that the autocorrelation time τ_O depends on the quantity O .

An alternative definition is the integrated autocorrelation time $\tau_O^{(int)}$, defined by

$$\tau_O^{(int)} = \frac{\sum_{\Delta=1}^{\infty} (\langle O_t O_{t+\Delta} \rangle - \langle O \rangle^2)}{\langle O^2 \rangle - \langle O \rangle^2} \quad (24)$$

As usual the expectation value of the quantity O can be estimated by the mean:

$$\bar{O} \equiv \frac{1}{N} \sum_i = 1^N O_i \quad (25)$$

The error estimate

$$\Delta O = \sqrt{\frac{\text{Var}O}{N}} \quad (26)$$

has to be modified because consecutive measurements are correlated. The error estimate $(\Delta O)^2$ is calculated as the expectation value of the squared difference between sample average and expectation value:

$$\begin{aligned} (\Delta O)^2 &= \langle (\bar{O} - \langle O \rangle)^2 \rangle = \langle \left(\frac{1}{N} \sum_{t=1}^N O_t - \langle O \rangle \right)^2 \rangle \\ &= \frac{1}{N^2} \sum_{i=1}^N (\langle O_i^2 \rangle - \langle O \rangle^2) \\ &\quad + \frac{2}{N^2} \sum_{t=1}^N \sum_{\Delta=1}^{N-t} (\langle O_t O_{t+\Delta} \rangle - \langle O \rangle^2) \\ &\approx \frac{1}{N} \text{Var}O (1 + 2\tau_O^{(int)}) \\ &\approx \frac{1}{N-1} \langle \bar{O}^2 - \bar{O}^2 \rangle (1 + 2\tau_O^{(int)}) \end{aligned} \quad (27)$$

In going from the second to third line we assumed $\tau_O^{(int)} \ll N$ and extended the summation over Δ to infinity. In the last line we replaced the variance by an estimate obtained from the sample. We see that the number of statistical uncorrelated samples is reduced from N to $N/(1 + 2\tau_O^{(int)})$.

In many Monte Carlo simulations the error analysis is unfortunately not done accurately. Thus we wish to discuss this topic here in more detail.

2.4.2. The binning analysis

The binning analysis is a reliable way to estimate the integrated autocorrelation times. Starting from the original series of measurements $O_i^{(0)}$ with $i = 1, \dots, N$ we iteratively create “binned” series by averaging over two consecutive entries:

$$O_i^{(l)} := \frac{1}{2} (O_{2i-1}^{(l-1)} + O_{2i}^{(l-1)}), \quad i = 1, \dots, N_l \equiv N/2^l. \quad (28)$$

These bin averages $O_i^{(l)}$ are less correlated than the original values $O_i^{(0)}$. The mean value is still the same.

The errors $\Delta O^{(l)}$, estimated incorrectly using equation (26)

$$\Delta O^{(l)} = \sqrt{\frac{\text{Var}O^{(l)}}{N_l - 1}} \approx \frac{1}{N_l} \sqrt{\sum_{i=1}^{N_l} (O_i^{(l)} - \overline{O^{(l)}})^2} \quad (29)$$

however increase as a function of bin size 2^l . For $2^l \gg \tau_O^{(\text{int})}$ the bins become uncorrelated and the errors converge to the correct error estimate:

$$\Delta O = \lim_{l \rightarrow \infty} \Delta O^{(l)}. \quad (30)$$

This binning analysis gives a reliable recipe for estimating errors and autocorrelation times. One has to calculate the error estimates for different bin sizes l and check if they converge to a limiting value. If convergence is observed the limit ΔO is a reliable error estimate, and $\tau_O^{(\text{int})}$ can be obtained from equation (27) as

$$\tau_O^{(\text{int})} = \frac{1}{2} \left[\left(\frac{\Delta A}{\Delta O^{(0)}} \right)^2 - 1 \right] \quad (31)$$

If however no convergence of the $\Delta O^{(l)}$ is observed we know that $\tau_O^{(\text{int})}$ is longer than the simulation time and we have to perform *much* longer simulations to obtain reliable error estimates.

To be really sure about convergence and autocorrelations it is very important to start simulations always on tiny systems and check convergence carefully before simulating larger systems.

2.4.3. Jackknife analysis

The binning procedure is a straightforward way to determine errors and autocorrelation times for Monte Carlo measurements. For functions of measurements like $U = \langle A \rangle / \langle B \rangle$ it becomes difficult because of error propagation and cross-correlations.

Then the jackknife procedure can be used. We again split the measurements into M bins of size $N/M \gg \tau^{(\text{int})}$ that should be much larger than any of the autocorrelation times.

We could now evaluate the complex quantity U in each of the M bins and obtain an error estimate from the variance of these estimates. As each of the bins contains only a rather small number of measurements N/M the statistics will not be good. The jackknife procedure instead works with $M + 1$ evaluations of U . U_0 is the estimate using all bins, and U_i for $i = 1, \dots, M$ is the value when all bins *except* the i -th bin are used. That way we always work with a large data set and obtain good statistics.

The resulting estimate for U will be:

$$U = U_0 - (M - 1)(\overline{U} - U_0) \quad (32)$$

with a statistical error

$$\Delta U = \sqrt{M-1} \left(\frac{1}{M} \sum_{i=1}^M (U_i)^2 - (\bar{U})^2 \right)^{1/2}, \quad (33)$$

where

$$\bar{U} = \frac{1}{M} \sum_{i=1}^M U_i, \quad (34)$$

The jackknife analysis is implemented in the ALPS library [10, 11].

2.4.4. Equilibration

Thermalization is as important as autocorrelations. The Markov chain converges only asymptotically to the desired distribution. Consequently, Monte Carlo measurements should be started only after a large number N_{eq} of equilibration steps, when the distribution is sufficiently close to the asymptotic distribution. N_{eq} has to be much larger than the thermalization time which is defined similar to the autocorrelation time as:

$$\tau_O^{(eq)} = \frac{\sum_{\Delta=1}^{\infty} (\langle O_0 O_{\Delta} \rangle - \langle O \rangle^2)}{\langle O_0 \rangle \langle O \rangle - \langle O \rangle^2} \quad (35)$$

It can be shown that the thermalization time is the maximum of all autocorrelation times for all observables and is related to the second largest eigenvalue Λ_2 of the Markov transition matrix W by $\tau^{(th)} = -1/\ln \Lambda_2$. It is recommended to thermalize the system for at least ten times the thermalization time before starting measurements.

3. CLASSICAL MONTE CARLO SIMULATIONS

Before getting to algorithms for quantum systems we first review the corresponding algorithms for classical systems, starting with the Ising model as the simplest model.

3.1. The Ising model

The Ising model is the simplest model for a magnetic system and a prototype statistical system. We will use it for our discussion of thermodynamic phase transitions. It consists of an array of classical spins $\sigma_i = \pm 1$ that can point either up ($\sigma_i = +1$) or down ($\sigma_i = -1$). The Hamiltonian is

$$H = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j, \quad (36)$$

where the sum goes over nearest neighbor spin pairs.

Two parallel spins contribute an energy of $-J$ while two antiparallel ones contribute $+J$. In the ferromagnetic case the state of lowest energy is the fully polarized state where all spins are aligned, either pointing up or down.

At finite temperatures the spins start to fluctuate and also states of higher energy contribute to thermal averages. The average magnetization thus decreases from its full value at zero temperature. At a critical temperature T_c there is a second order phase transition to a disordered phase. The Ising model is the simplest magnetic model exhibiting such a phase transition and is often used as a prototype model for magnetism.

The thermal average of a quantity A at a finite temperature T is given by a sum over all states:

$$\langle A \rangle = \frac{1}{Z} \sum_i A_i \exp(-\beta E_i), \quad (37)$$

where $\beta = 1/k_B T$ is the inverse temperature. A_i is the value of the quantity A in the configuration i . E_i is the energy of that configuration.

The partition function

$$Z = \sum_i \exp(-\beta E_i) \quad (38)$$

normalizes the probabilities $p_i = \exp(-\beta E_i)/Z$.

For small systems it is possible to evaluate these sums exactly. As the number of states grows like 2^N a straight-forward summation is possible only for very small N . For large higher dimensional systems Monte Carlo summation/integration is the method of choice.

3.2. The single spin flip Metropolis algorithm

As was discussed in connection with integration it is usually not efficient to estimate the average (37) using simple sampling. The optimal method is importance sampling,

where the states i are not chosen uniformly but with the correct probability p_i , which we can again do using the Metropolis algorithm.

The simplest Monte Carlo algorithm for the Ising model is the single spin flip Metropolis algorithm which defines a Markov chain through phase space.

- Starting with a configuration c_i propose to flip a single spin, leading to a new configuration c' .
- Calculate the energy difference $\Delta E = E[c'] - E[c_i]$ between the configurations c' and c_i .
- If $\Delta E < 0$ the next configuration is $c_{i+1} = c'$
- If $\Delta E > 0$ then $c_{i+1} = c'$ with probability $\exp(-\beta \Delta E)$, otherwise $c_{i+1} = c_i$. We do that by drawing a random number r uniformly distributed in the interval $[0, 1[$ and set $c_{i+1} = c'$ if $r < \exp(-\beta \Delta E)$.
- Measure all the quantities of interest in the new configuration.

This algorithm is ergodic since any configuration can be reached from any other in a finite number of spin flips. It also fulfills the detailed balance condition.

3.3. Systematic errors: boundary and finite size effects

In addition to statistical errors due to the Monte Carlo sampling our simulations suffer from systematic errors due to boundary effects and the finite size of the system.

Unless one wants to study finite systems with open boundaries, boundary effects can be avoided completely by using periodic boundary conditions. The lattice is continued periodically, forming a torus. The left neighbor of the leftmost spin is just the rightmost boundary spin, etc..

Although we can avoid boundary effects, finite size effects remain since now all correlations are periodic with the linear system size as period. Here is how we can treat them:

- *Away from phase transitions* the correlation length ξ is finite and finite size effects are negligible if the linear system size $L \gg \xi$. Usually $L > 6\xi$ is sufficient, but this should be checked for each simulation.
- *In the vicinity of continuous phase transitions* we encounter a problem: the correlation length ξ diverges. Finite size scaling can come to the rescue and can be used to obtain the critical behavior. A detailed discussion of finite size scaling is beyond the scope of these notes.

3.4. Critical behavior of the Ising model

Close to the phase transition at T_c again scaling laws characterize the behavior of all physical quantities. The average magnetization scales as

$$m(T) = \langle |M|/V \rangle \propto (T_c - T)^\beta, \quad (39)$$

where M is the total magnetization and V the system volume (number of spins).

The magnetic susceptibility $\chi = \frac{dm}{dh}|_{h=0}$ can be calculated from magnetization fluctuations and diverges with the exponent γ :

$$\chi(T) = \frac{\langle M^2/V \rangle - \langle |M|/V \rangle^2}{T} \propto |T_c - T|^{-\gamma}. \quad (40)$$

The correlation length ξ is defined by the asymptotically exponential decay of the two-spin correlations:

$$\langle \sigma_0 \sigma_r \rangle - \langle |m| \rangle^2 \propto \exp(-r/\xi). \quad (41)$$

It is best calculated from the structure factor $S(\mathbf{q})$, defined as the Fourier transform of the correlation function. For small \mathbf{q} the structure factor has a Lorentzian shape:

$$S(\mathbf{q}) = \frac{1}{1 + q^2 \xi^2} + O(q^4). \quad (42)$$

The correlation length diverges as

$$\xi(p) \propto |T - T_c|^{-\nu}. \quad (43)$$

At the critical point the correlation function itself follows a power law:

$$\langle \sigma_0 \sigma_r \rangle \propto r^{-(d-2+\eta)} \quad (44)$$

where $\eta = 2\beta/\nu - d + 2$.

The specific heat $C(T)$ diverges logarithmically in two dimensions:

$$C(T) \propto \ln|T - T_c| \propto |T - T_c|^{-\alpha} \quad (45)$$

and the critical exponent $\alpha = 0$.

A good estimate of T_c is obtained from the Binder cumulant

$$U = 1 - \frac{\langle M^4 \rangle}{3\langle M^2 \rangle^2}, \quad (46)$$

which has a universal value at p_c , also the Binder cumulant has a universal value at T_c . The curves of $U(T)$ for different system sizes L all cross in one point at T_c . This is a consequence of the finite size scaling ansatz:

$$\begin{aligned} \langle M^4 \rangle &= (T - T_c)^{4\beta} u_4((T - T_c)L^{1/\nu}) \\ \langle M^2 \rangle &= (T - T_c)^{2\beta} u_2((T - T_c)L^{1/\nu}). \end{aligned} \quad (47)$$

Thus

$$U(T, L) = 1 - \frac{u_4((T - T_c)L^{1/\nu})}{3u_2((T - T_c)L^{1/\nu})^2}, \quad (48)$$

which for $T = T_c$ is universal and independent of system size L :

$$U(T_c, L) = 1 - \frac{u_4(0)}{3u_2(0)^2} \quad (49)$$

High precision Monte Carlo simulations actually show that not all lines cross exactly at the same point, but that due to higher order corrections to finite size scaling the crossing point moves slightly, proportional to $L^{-1/\nu}$, allowing a high precision estimate of T_c and ν . For details of the determination of critical points and exponents see e.g. Ref [12, 13].

3.5. “Critical slowing down” and cluster Monte Carlo methods

The importance of autocorrelation becomes clear when we wish to simulate the Ising model at low temperatures. The mean magnetization $\langle m \rangle$ is zero on any finite cluster, as there is a degeneracy between a configuration and its spin reversed counterpart. If, however, we start at low temperatures with a configuration with all spins aligned up it will take extremely long time for all spins to be flipped by the single spin flip algorithm. This problem appears as soon as we get close to the critical temperature, where it was observed that the autocorrelation times diverge as

$$\tau \propto [\min(\xi, L)]^z. \quad (50)$$

with a dynamical critical exponents $z \approx 2$ for all local update methods like the single spin flip algorithm.

The reason is that at low temperatures it is very unlikely that even one spin gets flipped, and even more unlikely for a large cluster of spins to be flipped. The solution to this problem in the form of cluster updates was found in 1987 and 1989 by Swendsen and Wang [14] and by Wolff [15]. Instead of flipping single spins they propose to flip big clusters of spins and choose them in a clever way so that the probability of flipping these clusters is large.

3.5.1. Cluster updates

We use the Fortuin-Kastelyn representation of the Ising model, as generalized by Kandel and Domany. The phase space of the Ising model is enlarged by assigning a set \mathcal{G} of possible “graphs” to each configuration C in the set of configurations \mathcal{C} . We write the partition function as

$$Z = \sum_{C \in \mathcal{C}} \sum_{G \in \mathcal{G}} W(C, G) \quad (51)$$

where the new weights $W(C, G) > 0$ are chosen such that Z is the partition function of the original model by requiring

$$\sum_{G \in \mathcal{G}} W(C, G) = W(C) := \exp(-\beta E[C]), \quad (52)$$

where $E[C]$ is the energy of the configuration C .

The algorithm now proceeds as follows. First we assign a graph $G \in \mathcal{G}$ to the configuration C , chosen with the correct probability

$$P_C(G) = W(C, G)/W(C). \quad (53)$$

Then we choose a new configuration C' with probability $p[(C, G) \rightarrow (C', G)]$, keeping the graph G fixed; next a new graph G' is chosen

$$C \rightarrow (C, G) \rightarrow (C', G) \rightarrow C' \rightarrow (C', G') \rightarrow \dots \quad (54)$$

What about detailed balance? The procedure for choosing graphs with probabilities P_G obeys detailed balance trivially. The non-trivial part is the probability of choosing a new configuration C' . There detailed balance requires:

$$W(C, G)p[(C, G) \rightarrow (C', G)] = W(C', G)p[(C', G) \rightarrow (C, G)], \quad (55)$$

which can be fulfilled using either the heat bath algorithm

$$p[(C, G) \rightarrow (C', G)] = \frac{W(C', G)}{W(C, G) + W(C', G)} \quad (56)$$

or by again using the Metropolis algorithm:

$$p[(C, G) \rightarrow (C', G)] = \max(W(C', G)/W(C, G), 1) \quad (57)$$

The algorithm simplifies a lot if we can find a graph mapping such that the graph weights do not depend on the configuration whenever it is nonzero in that configuration. This means, we want the graph weights to be

$$W(C, G) = \Delta(C, G)V(G), \quad (58)$$

where

$$\Delta(C, G) := \begin{cases} 1 & \text{if } W(C, G) \neq 0, \\ 0 & \text{otherwise.} \end{cases} \quad (59)$$

Then equation (56) simply becomes $p = 1/2$ and equation (57) reduces to $p = 1$ for any configuration C' with $W(C', G) \neq 0$.

3.5.2. The cluster algorithms for the Ising model

Let us now show how this abstract and general algorithm can be applied to the Ising model. Our graphs will be bond-percolation graphs on the lattice. Spins pointing into the same direction can be connected or disconnected. Spins pointing in opposite directions will always be disconnected. In the Ising model we can write the weights $W(C)$ and $W(C, G)$ as products over all bonds b :

$$W(C) = \prod_b w(C_b) \quad (60)$$

$$W(C, G) = \prod_b w(C_b, G_b) = \prod_b \Delta(C_b, G_b)V(G_b) \quad (61)$$

TABLE 1. Local bond weights for the Kandel-Domany representation of the Ising model.

	$c = \uparrow\uparrow$	$c = \downarrow\uparrow$	$c = \uparrow\downarrow$	$c = \downarrow\downarrow$	$V(g)$
$\Delta(c, \text{discon.})$	1	1	1	1	$e^{-\beta J}$
$\Delta(c, \text{con.})$	1	0	0	1	$e^{\beta J} - e^{-\beta J}$
$w(c)$	$\exp(\beta J)$	$\exp(-\beta J)$	$\exp(-\beta J)$	$\exp(\beta J)$	

where the local bond configurations C_b can be one of $\{\uparrow\uparrow, \downarrow\uparrow, \uparrow\downarrow, \downarrow\downarrow\}$ and the local graphs can be “connected” or “disconnected”. The graph selection can thus be done locally on each bond.

Table 1 shows the local bond weights $w(c, g)$, $w(c)$, $\Delta(c, g)$ and $V(g)$. It can easily be checked that the sum rule (52) is satisfied.

The probability of a connected bond is $[\exp(\beta J) - \exp(-\beta J)]/\exp(\beta J) = 1 - \exp(-2\beta J)$ if two spins are aligned and zero otherwise. These connected bonds group the spins into clusters of aligned spins.

A new configuration C' with the same graph G can differ from C only by flipping clusters of connected spins. Thus the name “cluster algorithms”. The clusters can be flipped independently, as the flipping probabilities $p[(C, G) \rightarrow (C', G)]$ are configuration independent constants.

There are two variants of cluster algorithms that can be constructed using the rules derived above.

3.5.3. The Swendsen-Wang algorithm

The Swendsen-Wang or multi-cluster algorithm proceeds as follows:

- Each bond in the lattice is assigned a label “connected” or “disconnected” according to above rules. Two aligned spins are connected with probability $1 - \exp(-2\beta J)$. Two antiparallel spins are never connected.
- Next a cluster labeling algorithm, like the Hoshen-Kopelman algorithm is used to identify clusters of connected spins.
- Measurements are performed, using improved estimators discussed in the next section.
- Each cluster of spins is flipped with probability $1/2$.

3.5.4. The Wolff algorithm

The Swendsen Wang algorithm gets less efficient in dimensions higher than two as the majority of the clusters will be very small ones, and only a few large clusters exist. The Wolff algorithm is similar to the Swendsen-Wang algorithm but builds only one cluster starting from a randomly chosen point. As the probability of this point being on a

cluster of size s is proportional to s the Wolff algorithm builds preferably larger clusters. It works in the following way:

- i) Choose a random spin as the initial cluster.
- ii) If a neighboring spin is parallel to the initial spin it will be added to the cluster with probability $1 - \exp(-2\beta J)$.
- iii) Repeat step ii) for all points newly added to the cluster and repeat this procedure until no new points can be added.
- iv) Perform measurements using improved estimators.
- v) Flip all spins in the cluster.

We will see in the next section that the linear cluster size diverges with the correlation length ξ and that the average number of spins in a cluster is just χT . Thus the algorithm adapts optimally to the physics of the system and the dynamical exponent $z \approx 0$, thus solving the problem of critical slowing down. Close to criticality these algorithms are many orders of magnitudes (a factor L^2) better than the local update methods.

3.6. Improved Estimators

In this section we present a neat trick that can be used in conjunction with cluster algorithms to reduce the variance, and thus the statistical error of Monte Carlo measurements. Not only do these “improved estimators” reduce the variance. They are also much easier to calculate than the usual “simple estimators”.

To derive them we consider the Swendsen-Wang algorithm. This algorithm divides the lattice into N_c clusters, where all spins within a cluster are aligned. The next possible configuration is any of the 2^{N_c} configurations that can be reached by flipping any subset of the clusters. The idea behind the “improved estimators” is to measure not only in the new configuration but in all equally probable 2^{N_c} configurations.

As simplest example we consider the average magnetization $\langle m \rangle$. We can measure it as the expectation value $\langle \sigma_i \rangle$ of a single spin. As the cluster to which the spin belongs can be freely flipped, and the flipped cluster has the same probability as the original one, the improved estimator is

$$\langle m \rangle = \left\langle \frac{1}{2}(\sigma_i - \bar{\sigma}_i) \right\rangle = 0. \quad (62)$$

This result is obvious because of symmetry, but we saw that at low temperatures a single spin flip algorithm will fail to give this correct result since it takes an enormous time to flip all spins. Thus it is encouraging that the cluster algorithms automatically give the exact result in this case.

Correlation functions are not much harder to measure:

$$\langle \sigma_i \sigma_j \rangle = \begin{cases} 1 & \text{if } \vec{i} \text{ and } \vec{j} \text{ are on the same cluster} \\ 0 & \text{otherwise} \end{cases} \quad (63)$$

To derive this result consider the two cases and write down the improved estimators by considering all possible cluster flips.

Using this simple result for the correlation functions the mean square of the magnetization is

$$\langle m^2 \rangle = \frac{1}{N^2} \sum_{\vec{i}, \vec{j}} \langle \sigma_{\vec{i}} \sigma_{\vec{j}} \rangle = \frac{1}{N^2} \left\langle \sum_c S(c)^2 \right\rangle, \quad (64)$$

where $S(c)$ is the number of spins in a cluster c . The susceptibility above T_c is simply given by $\beta \langle m^2 \rangle$ and can also easily be calculated by above sum over the squares of the cluster sizes.

In the Wolff algorithm only a single cluster is built. Above sum (64) can be rewritten to be useful also in case of the Wolff algorithm:

$$\begin{aligned} \langle m^2 \rangle &= \frac{1}{N^2} \left\langle \sum_c S(c)^2 \right\rangle \\ &= \frac{1}{N^2} \sum_{\vec{i}} \frac{1}{S_{\vec{i}}} S_{\vec{i}}^2 \\ &= \frac{1}{N^2} \sum_{\vec{i}} S_{\vec{i}} = \frac{1}{N} \langle S(c) \rangle, \end{aligned} \quad (65)$$

where $S_{\vec{i}}$ is the size of the cluster containing the initial site \vec{i} . The expectation value for m^2 is thus simply the mean cluster size. In this derivation we replaced the sum over all clusters by a sum over all sites and had to divide the contribution of each cluster by the number of sites in the cluster. Next we can replace the average over all lattice sites by the expectation value for the cluster on a randomly chosen site, which in the Wolff algorithm will be just the one Wolff cluster we build.

Generalizations to other quantities, like the structure factor $S(\vec{q})$ are straightforward. While the calculation of $S(\vec{q})$ by Fourier transform needs at least $O(N \ln N)$ steps, it can be done much faster using improved estimators, here derived for the Wolff algorithm:

$$\begin{aligned} \langle S(\vec{q}) \rangle &= \frac{1}{N^2} \sum_{\vec{r}, \vec{r}'} \sigma_{\vec{r}} \sigma_{\vec{r}'} \exp(i\vec{q}(\vec{r} - \vec{r}')) \\ &= \frac{1}{NS(c)} \sum_{\vec{r}, \vec{r}' \in \text{cluster}} \sigma_{\vec{r}} \sigma_{\vec{r}'} \exp(i\vec{q}(\vec{r} - \vec{r}')) \\ &= \frac{1}{NS(c)} \left| \sum_{\vec{r} \in \text{cluster}} \exp(i\vec{q}\vec{r}) \right|^2, \end{aligned} \quad (66)$$

This needs only $O(S(c))$ operations and can be measured directly when constructing the cluster.

Care must be taken for higher order correlation functions. Improved estimators for quantities like m^4 which need at least two clusters and cannot be measured in an improved way using the Wolff algorithm.

3.7. Generalizations of cluster algorithms

Cluster algorithms can be used not only for the Ising model but for a large class of classical, and even quantum spin models. The quantum version is the “loop algorithm”, which will be discussed later in the course. In this section we discuss generalizations to other classical spin models.

Before discussing specific models we remark that generalizations to models with different coupling constants on different bonds, or even random couplings are straightforward. All decisions are done locally, individually for each spin or bond, and the couplings can thus be different at each bond.

3.7.1. Potts models

q -state Potts models are the generalization of the Ising model to more than two states. The Hamilton function is

$$H = -J \sum_{\langle i,j \rangle} \delta_{s_i, s_j}, \quad (67)$$

where the states s_i can take any integer value in the range $1, \dots, q$. The 2-state Potts model is just the Ising model with some trivial rescaling.

The cluster algorithms for the Potts models connect spins with probability $1 - e^{-\beta J}$ if the spins have the same value. The clusters are then “flipped” to any arbitrarily chosen value in the range $1, \dots, q$.

3.7.2. $O(N)$ models

Another, even more important generalization are the $O(N)$ models. Well known examples are the XY -model with $N = 2$ and the Heisenberg model with $N = 3$. In contrast to the Ising model the spins can point into any arbitrary direction on the N -sphere. The spins in the XY model can point into any direction in the plane and can be characterized by a phase. The spins in the Heisenberg model point into any direction on a sphere.

The Hamilton function is:

$$H = -J \sum_{\langle i,j \rangle} \vec{S}_i \cdot \vec{S}_j, \quad (68)$$

where the states \vec{S}_i are $SO(N)$ vectors.

Cluster algorithms are constructed by projecting all spins onto a random direction $\vec{e} \in SO(N)$. The cluster algorithm for the Ising model can then be used for this projection. Two spins \vec{S}_i and \vec{S}_j are connected with probability

$$1 - \exp \left(\min[0, -2\beta J(\vec{e} \cdot \vec{S}_i)(\vec{e} \cdot \vec{S}_j)] \right). \quad (69)$$

The spins are flipped by inverting the projection onto the \vec{e} -direction:

$$\vec{S}_i \rightarrow \vec{S}_i - 2(\vec{e} \cdot \vec{S}_i)\vec{e}. \quad (70)$$

In the next update step a new direction \vec{e} is chosen.

3.8. The multicanonical ensemble and Wang Landau sampling

While cluster updates can solve critical slowing down at second order phase transitions they are usually inefficient at first order phase transitions and in frustrated systems. Let us consider a first order phase transition, such as in a two-dimensional q -state Potts model with Hamilton function

$$H = -J \sum_{\langle i,j \rangle} \delta_{\sigma_i \sigma_j}, \quad (71)$$

where the spins σ_i can now take the integer values $1, \dots, q$. For $q > 4$ this model exhibits a first order phase transition, accompanied by exponential slowing down of local single-spin updates. The exponential slow-down is caused by the free energy barrier between the two coexisting meta-stable states at the first order phase transition.

This barrier can be quantified by considering the energy histogram

$$H_{\text{canonical}}[E] = g(E) P_{\text{Boltzmann}}(E) \propto g(E) \exp(-\beta E) \quad (72)$$

which is the probability of encountering a configuration with energy E during the Monte Carlo simulation. Here

$$g(E) = \sum_c \delta_{E, E(c)} \quad (73)$$

is the density of states. Away from first order phase transitions, $H_{\text{canonical}}[E]$ has approximately Gaussian shape, centered around the mean energy. At first order phase transitions, where the energy jumps discontinuously the histogram $H_{\text{canonical}}[E]$ develops a double-peak structure. The minimum of $H_{\text{canonical}}[E]$ between these two peaks, which the simulation has to cross in order to go from one phase to the other, becomes exponentially small upon increasing the system size. This leads to exponentially large autocorrelation times.

The simplest lattice model showing such a first order thermal phase transition is a two-dimensional Potts model with large q , e.g. $q = 10$. For this model we show in Fig. 1 the probability $P(E, T)$ of visiting a configuration with energy E . This is:

$$P(E, T) = \rho(E) p(E) = \rho(E) e^{-E/k_B T}, \quad (74)$$

where the density of states $\rho(E)$ counts the number of states with energy E . At the critical temperature there are two coexisting phases, the ordered and disordered ones with different energies. In order to tunnel from one to the other one has to continuously change the energy and thus go through the probability minimum between the two peaks. This probability decreases exponentially with system size and we thus have a real problem!

A solution to this tunneling problem are extended ensembles, such as the multicanonical ensemble [16, 17], where the weight of a configuration c is given by

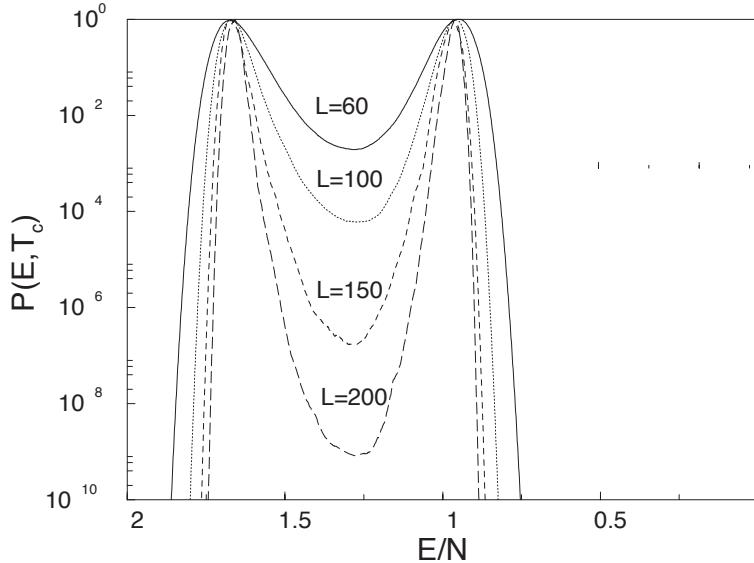


FIGURE 1. The probability $P(E, T_c) = \rho(E) \exp(-E/k_B T_c)$ of visiting a state with energy E in a $q = 10$ -state Potts model at the critical temperature. The tunneling probability between the two phases (the dip between the two maxima) becomes exponentially small for large systems. This figure is taken from the paper: F. Wang and D.P. Landau, Phys. Rev. Lett. **86**, 2050 (2001).

$P_{\text{multicanonical}}(c) \propto 1/g(E(c))$ instead of the Boltzmann weight $\exp(-\beta E(c))$. The multicanonical ensemble leads to a flat histogram in energy space

$$H_{\text{multicanonical}}[E] = g(E)P_{\text{multicanonical}}(E) \propto g(E) \frac{1}{g(E)} = \text{const.} \quad (75)$$

removing the exponentially small minimum. After performing a simulation, measurements in the multicanonical ensemble are reweighted by a factor $P_{\text{Boltzmann}}(E)/P_{\text{multicanonical}}(E)$ to obtain averages in the canonical ensemble.

Since the density of states $g(E)$ and thus the multicanonical weights $P_{\text{multicanonical}}$ are not known initially, a scalable algorithm to estimate these quantities is needed. The Wang-Landau algorithm [18, 19] is a simple iterative method to obtain the density of states $g(E)$ and the multicanonical weights $P_{\text{multicanonical}}(E) \propto 1/g(E)$. The approach by Wang and Landau is crude but simple. They start with a (very bad guess) $\rho(E) = 1$ for all energies and iteratively improve it:

- Start with $\rho(E) = 1$ and $f = e$
- Repeat
 - Reset a histogram of energies $H(E) = 0$
 - Perform simulations until a histogram of energies $H(E)$ is “flat”
 - * pick a random site
 - * attempt a local Metropolis update using $p(E) = 1/\rho(E)$
 - * increase the histogram at the current energy E : $H(E) \leftarrow H(E) + 1$
 - * increase the estimate for $\rho(E)$ at the current energy E : $\rho(E) \leftarrow \rho(E) \cdot f$

- once $H(E)$ is “flat” (e.g. the minimum is at least 80% of the mean), reduce
 $f \leftarrow \sqrt{f}$
- stop once $f \approx 1 + 10^{-8}$

As you can see, only a few lines of code need to be changed in your local update algorithm for the Ising model, but a few remarks are necessary:

1. Check for flatness of the histogram not at every step but only after a reasonable number of sweeps N_{sweeps} . One sweep is defined as one attempted update per site.
2. The initial value for f needs to be carefully chosen, $f = e$ is only a rough guide. As discussed in the papers a good choice is picking the initial f such that $f^{N_{\text{sweeps}}}$ is approximately the total number of states (e.g. q^N for a q -state Potts model with N sites).
3. The flatness criterion is quite arbitrary and some research is still necessary to find the optimal choice.
4. The density of states $\rho(E)$ can become very large and easily exceed 10^{10000} . In order to obtain such large numbers the *multiplicative increase* $\rho(E) \leftarrow \rho(E) \cdot f$ is essential. A naive additive guess $\rho(E) \leftarrow \rho(E) + f$ would never be able to reach the large numbers needed.
5. Since $\rho(E)$ is so large, we only store its *logarithm*. The update step is thus $\ln \rho(E) \leftarrow \ln \rho(E) + \ln f$. The Metropolis acceptance probability will be

$$P = \min[1, \exp(\ln \rho(E_{\text{old}}) - \ln \rho(E_{\text{new}}))] \quad (76)$$

Another advantage of the Wang-Landau algorithm is that, once we know the density of states $\rho(E)$, we can directly calculate the partition function

$$Z = \sum_c E_c e^{-E_c/k_B T} = \sum_E \rho(E) e^{-E/k_B T} \quad (77)$$

and the free energy

$$F = -k_B T \ln Z = -k_B T \ln \sum_E \rho(E) e^{-E/k_B T} \quad (78)$$

which are both not directly accessible in any other Monte Carlo algorithm. All other thermodynamic properties such as the susceptibility or the specific heat can now be calculated simply as derivatives of the free energy.

Recent investigations have shown that a flat histogram, as obtained by the multicanonical ensemble, is not optimal but still shows signs of critical slowing down [20]. Optimized ensembles, assigning larger weights to configurations in the critical region, can be found and lead to further improvements in the efficiency of algorithms by several orders of magnitude [21].

$\exp()$

REFERENCES

10. F. Alet, P. Dayal, A. Grzesik, M. Honecker, A. Laeuchli, S. R. Manmana, I. P. McCulloch, F. Michel, R. M. Noack, G. Schmid, U. Schollwoeck, S. Stoeckli, S. Todo, S. Trebst, M. Troyer, P. Werner, and S. Wessel, *J. Phys. Soc. Jap. Suppl.* **74**, 30 (2005).
11. A. Albuquerque, F. Alet, P. Corboz, P. Dayal, A. Feiguin, S. Fuchs, L. Gamper, E. Gull, S. Gürtler, A. Honecker, R. Igarashi, M. Körner, M. Kozhevnikov, A. Läuchli, S. Manmana, M. Matsumoto, I. McCulloch, F. Michel, R. Noack, G. Pawłowski, L. Pollet, T. Pruschke, U. Schollwöck, S. Todo, S. Trebst, M. Troyer, P. Werner, and S. Wessel, *J. Mag. Mag. Mat.* **310**, 1187 (2007).
12. A. M. Ferrenberg, and D. P. Landau, *Phys. Rev. B* **44**, 5081 (1991).
13. K. Chen, A. M. Ferrenberg, and D. P. Landau, *Phys. Rev. B* **48**, 3249 (1993).
14. R. Swendsen, and J.-S. Wang, *Phys. Rev. Lett.* **58**, 86–89 (1987).
15. U. Wolff, *Phys. Rev. Lett.* **62**, 361–364 (1989).
16. B. A. Berg, and T. Neuhaus, *Phys. Lett. B* **267**, 249–253 (1991).
17. B. A. Berg, and T. Neuhaus, *Phys. Rev. Lett.* **68**, 9–12 (1992).
18. F. Wang, and D. Landau, *Phys. Rev. Lett.* **86**, 2050–2053 (2001).
19. F. Wang, and D. Landau, *Phys. Rev. E* **64**, 056101 (2001).
20. P. Dayal, S. Trebst, S. Wessel, D. Würtz, M. Troyer, S. Sabhapandit, and S. N. Coppersmith, *Phys. Rev. Lett.* **92**, 097201 (2004).
21. S. Trebst, D. Huse, and M. Troyer, *Phys. Rev. E* **70**, 046701 (2004).
22. N. V. Prokof'ev, B. V. Svistunov, and I. S. Tupitsyn, *JETP* **87**, 310–321 (1998).
23. A. Sandvik, and J. Kurkijärvi, *Phys. Rev. B* **43**, 5950–5961 (1991).
24. D. Handscomb, *Proc. Cambridge Philos. Soc.* **58**, 594–598 (1962).
25. A. W. Sandvik, *Phys. Rev. B* **56**, 11678–11690 (1997).
26. E. L. Pollock, and D. M. Ceperley, *Phys. Rev. B* **36**, 8343–8352 (1987).
27. M. Jarrell, and J. Gubernatis, *Physics Reports* **269**, 133–195 (1996).