Computational Quantum Physics

Prof. Matthias Troyer (troyer@phys.ethz.ch)

ETH Zürich, SS 2010

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 HPR 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 $\mathcal{H}$:

$$|\Psi\rangle \in \mathcal{H} \tag{2.1}$$

Usually the wave functions are normalized:

$$|| \, |\Psi\rangle \, || = \sqrt{\langle\Psi|\Psi\rangle} = 1. \tag{2.2}$$

Here the "bra-ket" notation

$$\langle\Phi|\Psi\rangle \tag{2.3}$$

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

$$|\uparrow\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{2.4}$$

$$|\downarrow\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{2.5}$$

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

$$|\Psi\rangle = \alpha|\uparrow\rangle + \beta|\downarrow\rangle \tag{2.6}$$

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

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

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

$$\mathrm{Tr}\rho = 1. \tag{2.8}$$

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

$$\rho_{\mathrm{pure}} = |\Psi\rangle\langle\Psi|. \tag{2.9}$$

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

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

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

$$\rho_{\mathrm{mixed}} = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}. \tag{2.11}$$

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

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

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

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

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

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

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

$$[S^x, S^y] = S^x S^y - S^y S^x = i\hbar S^z, \tag{2.16}$$

$$[S^y, S^z] = i\hbar S^x, \tag{2.17}$$

$$[S^z, S^x] = i\hbar S^y, \tag{2.18}$$

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

$$|\Psi\rangle = |\uparrow\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{2.19}$$

or alternatively density matrix

$$\rho_\uparrow = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \tag{2.20}$$

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

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

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

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

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

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

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

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

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

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

$$\langle A\rangle = \langle\Psi|A|\Psi\rangle \tag{2.26}$$

or from a density matrix ρ as

$$\langle A\rangle = \mathrm{Tr}(\rho A). \tag{2.27}$$

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

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

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

$$[A, B] = i\hbar \tag{2.29}$$

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

$$\Delta A \Delta B \geq \hbar/2 \tag{2.30}$$

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

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

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

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

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

$$H|\Psi\rangle = E|\Psi\rangle. \tag{2.33}$$

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

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

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 Boltzman distribution

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

where the partition function

$$Z = \sum_i \exp(-\beta E_i) \tag{2.36}$$

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

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

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

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

where the partition function now is

$$Z = \mathrm{Tr}\exp(-\beta H) \tag{2.39}$$

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

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

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 $S = 0, \frac{1}{2}, 1, \frac{3}{2}, 2, \ldots$. The basis states $\{|s\rangle\}$ are usually chosen as eigenstates of the S^z operator

$$S^z|s\rangle = \hbar s|s\rangle, \tag{2.41}$$

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

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

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

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

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

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

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

$$H = -g\mu_B \vec{h}\cdot\vec{S}, \tag{2.46}$$

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

$$\hat{x} = \vec{x}, \tag{2.47}$$

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

$$\hat{p} = -i\hbar\nabla. \tag{2.48}$$

These two operators do not commute but their commutator is

$$[\hat{x}, \hat{p}] = i\hbar. \tag{2.49}$$

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

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

we use the quantum mechanical Hamiltonian operator

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

which gives the famous form

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

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

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

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

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

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

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

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

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

$$\langle m|\hat{q}|n\rangle \qquad \text{and} \qquad \langle m|\hat{p}|n\rangle \tag{2.56}$$

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 $a^\dagger$:

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

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

$$\tag{2.59}$$

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

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

and commutation relations

$$[a, a] = [a^\dagger, a^\dagger] = 0 \tag{2.61}$$

$$[a, a^\dagger] = 1. \tag{2.62}$$

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

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

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

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

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

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

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

$$x_n = n\Delta x \tag{3.2}$$

and the wave function at these points by

$$\psi_n = \psi(x_n). \tag{3.3}$$

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

$$\psi''(x) + k(x)\psi(x) = 0, \tag{3.4}$$

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:

$$\psi_{n\pm1} = \psi_n \pm \Delta x\psi'_n + \frac{\Delta x^2}{2}\psi''_n \pm \frac{\Delta x^3}{6}\psi^{(3)}_n + \frac{\Delta x^4}{24}\psi^{(4)}_n \pm \frac{\Delta x^5}{120}\psi^{(5)}_n + \mathrm{O}(\Delta x^6) \tag{3.5}$$

Adding ψ_{n+1} and ψ_{n-1} we obtain

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

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

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

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

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

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 potentials 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.

$$\psi(-a) = 1 \tag{3.9}$$

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

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 ($V(x) \to 0$ for $|x| \to \infty$). 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 $x < 0$:

$$\psi_L(x) = A\exp(-iqx) + B\exp(iqx) \tag{3.11}$$

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,

$$\psi_R(x) = C \exp(-iqx) \tag{3.12}$$

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

$$R = |B|^2/|A|^2 \tag{3.13}$$
$$T = 1/|A|^2 \tag{3.14}$$

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 $x \to \infty$ 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 $x \to \infty$.

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 search for a point b with $V(b) = E$

- 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

$$\begin{aligned} \psi_L(b) &= \alpha\psi_R(b) && (3.15)\\ \psi'_L(b) &= \alpha\psi'_R(b) && (3.16)\\ \psi''_L(b) &= \alpha\psi''_R(b) && (3.17) \end{aligned}$$

  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:

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

- 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

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

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

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

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

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

in the interval $[0, \infty[$. Given the singular character of the potential for $r \to 0$, 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 inear equations. Starting from the three-dimensional Schrödinger equation (we set $\hbar = 1$ from now on)

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

we discretize space and obtain the system of linear equations

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

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 `dsysv` 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 $\mathrm{O}(N)$ of the N^2 matrix elements are nonzero. For these sparse systems of equations, optimized iterative numerical algorithms exist[1] and are implemented in numerical libraries such as in the ITL library.[2]

[1] R. 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)

[2] J.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 availavle 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` eigensolver 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 implementations[3] 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

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

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

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

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

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

the matrix elements of the Hamilton operator H and by

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

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

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

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

$$H\vec{a} = ES\vec{a}. \tag{3.29}$$

[3] http://www.comp-phys.org/software/ietl/

[4] Z. 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

$$\begin{aligned} H\vec{a} &= ES\vec{a} \\ HU\vec{b} &= ESU\vec{b} \\ U^T HU\vec{b} &= EU^T SU\vec{b} = E\vec{b} \end{aligned} \tag{3.30}$$

and we end up with a standard eigenvalue problem for $U^T HU$. 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

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

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

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

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

$$|\psi(t_0)\rangle = \sum_n c_n |\phi_n\rangle. \tag{3.33}$$

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:

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

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

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

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

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

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

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

we reformulate the time evolution operator as

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

which is unitary!

This gives the simplest stable and unitary integrator algorithm

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

or equivalently

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

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

$$H = \hat{T} + \hat{V} \tag{3.41}$$

with

$$\hat{T} = \frac{1}{2m}\hat{p}^2 \tag{3.42}$$

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

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

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

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

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

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

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:

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

and the discretized algorithm starts as

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

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

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:

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

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

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

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

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

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

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

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

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 $\psi(\vec{x}_1, \ldots, \vec{x}_N)$ of the positions $\vec{x}_1, \ldots, \vec{x}_N$ 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 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 to 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

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

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 function[1]

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

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

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

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 $\psi(\vec{q}_1, \ldots, \vec{q}_N)$ of N particles we use the symmetrized wave function

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

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

[1] As 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

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

where $\mathrm{sgn}(p) = \pm 1$ 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

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

since this vanishes under antisymmetrization:

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

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 $N = 0, 1, \ldots, \infty$ particles is called the Fock space. It is the direct sum of the appropriately symmetrized single-particle Hilbert spaces $\mathcal{H}$:

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

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 $\{|\phi_1\rangle,\ldots,|\phi_L\rangle\}$ 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 $|n_1,\ldots,n_L\rangle$ 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

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

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

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

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 Hartee 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

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

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

$$a_i|\phi_j\rangle = \delta_{ij}|0\rangle \tag{4.13}$$

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:

$$|\phi_i\rangle = a_i^\dagger|0\rangle \tag{4.14}$$

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 $\sigma = -S, -S+1, ..., S$. 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):

$$[a_i, a_j] = [a_i^\dagger, a_j^\dagger] = 0 \tag{4.15}$$

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

For fermions, on the other hand, the operators anticommute

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

The anti-commutation implies that

$$(a_i^\dagger)^2 = a_i^\dagger a_i^\dagger = -a_i^\dagger a_i^\dagger \tag{4.18}$$

and that thus

$$(a_i^\dagger)^2 = 0, \tag{4.19}$$

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

Fock basis in second quantization and normal ordering

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

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

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 $a_1^\dagger a_2^\dagger|0\rangle = -a_1^\dagger a_2^\dagger|0\rangle$. We thus need to agree on a specific ordering of the creation operators to define what we mean by the state $|n_1, \ldots, n_L\rangle$. 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

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

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

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 $\psi^\dagger_\sigma(\vec{r})$ and $\psi_\sigma(\vec{r})$, creating and annihilating a fermion localized at a single point $\vec{r}$ in space. Their commutation relations are simply

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

The scalar products of the basis functions define a matrix

$$S_{ij} = \int d^3\vec{r} f^*_i(\vec{r}) f_j(\vec{r}), \tag{4.24}$$

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

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

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

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

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

$$\hat{a}^\dagger_{i\sigma} = \sum_j S_{ji} a^\dagger_{i\sigma}, \tag{4.27}$$

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

$$\{\hat{a}^\dagger_{i\sigma}, a_{j\sigma}\} = \delta_{ij}. \tag{4.28}$$

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

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

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 quatum 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

$$\hat{H} = \hat{T} + \hat{V}; \tag{5.1}$$

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

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

$$\hat{V} = V(\boldsymbol{R}). \tag{5.3}$$

In these equations $\hbar$ is the Plank's constant divided by 2π, m the particles mass, N the number of particles and the vector $\boldsymbol{R} \equiv (\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N)$ 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 $V(\boldsymbol{R})$ is determined by inter-particle interactions, in which case it can be written as the sum of pair contributions $V(\boldsymbol{R}) = \sum_{i<j} v(\boldsymbol{r}_i - \boldsymbol{r}_j)$, where $v(\boldsymbol{r})$ is the inter-particle potential; it can also be due to an external field, call it $v_{\text{ext}}(\boldsymbol{r})$, in which case it is just the sum of single particle contributions $V(\boldsymbol{R}) = \sum_i v_{\text{ex}}(\boldsymbol{r}_i)$.

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 $\exp\left(-\beta\hat{H}\right)$, where $\beta = 1/k_B T$, with k_B the Boltzmann's constant. The expectation value of an observable operator $\hat{O}$ is:

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

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

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

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

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

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

$$Z\left(N, T, V\right) = \int \rho\left(\boldsymbol{R}, \boldsymbol{R}, \beta\right) \mathrm{d}\boldsymbol{R}. \tag{5.7}$$

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

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

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

$$\rho\left(\boldsymbol{R}_1, \boldsymbol{R}_3, \beta_1 + \beta_2\right) = \int \rho\left(\boldsymbol{R}_1, \boldsymbol{R}_2, \beta_1\right)\rho\left(\boldsymbol{R}_2, \boldsymbol{R}_3, \beta_2\right)\mathrm{d}\boldsymbol{R}_2. \tag{5.9}$$

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:

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

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

$$\rho\left(\boldsymbol{R}_1, \boldsymbol{R}_{M+1}, \beta\right) = \int\cdots\int \mathrm{d}\boldsymbol{R}_2\mathrm{d}\boldsymbol{R}_3\cdots\mathrm{d}\boldsymbol{R}_M \\ \rho\left(\boldsymbol{R}_1, \boldsymbol{R}_2, \tau\right)\rho\left(\boldsymbol{R}_2, \boldsymbol{R}_3, \tau\right)\cdots\rho\left(\boldsymbol{R}_M, \boldsymbol{R}_{M+1}, \tau\right). \tag{5.11}$$

Eq. (5.11) is not useful as it is since the density matrices $\rho(\boldsymbol{R}_j, \boldsymbol{R}_{j+1}, \tau)$ 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 $\rho(\boldsymbol{R}_j, \boldsymbol{R}_{j+1}, \tau)$ 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*):

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

(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:

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

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:

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

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):

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

For later convenience we introduce the following definition:

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

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:

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

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:

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

The Trotter formula guarantees that in the limit $M \to \infty$ 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 $\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_M$.

Time-slice: the $j-$th term of a system configuration, indicated with $\boldsymbol{R}_j$, 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: $\left\{\boldsymbol{r}_1^i, \boldsymbol{r}_2^i, \ldots, \boldsymbol{r}_j^i, \ldots, \boldsymbol{r}_M^i\right\}$.

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

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

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

For distinguishable particles $\boldsymbol{R}_{M+1} \equiv \boldsymbol{R}_1$. Note that eq. (5.19) represents the partition function of a classical system of polymers. Every polymer is a necklake 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:

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

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

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

It is the solution of the Schrödinger equation with the initial condition $\phi\left(\boldsymbol{R},0\right)=\delta\left(\boldsymbol{R}-\boldsymbol{R}'\right)$. 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:

$$\phi(\boldsymbol{R},t)=\int G(\boldsymbol{R},\boldsymbol{R}',t)\phi(\boldsymbol{R}',0)\mathrm{d}\boldsymbol{R}'. \tag{5.22}$$

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:

$$\hat{H}=-\frac{\hbar^2}{2m}\frac{\mathrm{d}^2}{\mathrm{d}x^2}. \tag{5.23}$$

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

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

We introduce twice the completeness relation $\int\left|p\right\rangle\left\langle p\right|\mathrm{d}p=\boldsymbol{I}$, where $\left|p\right\rangle$ are the eigenstates of the momentum operator:

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

Here we have used the expression of the momentum eigenstates is 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:

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

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

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

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:

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

where $\mathbf{P}$ is one of the $N!$ permutations of the particle labels; this means that $\mathbf{P}\boldsymbol{R} = \left(\boldsymbol{r}^{p(1)}, \boldsymbol{r}^{p(2)}, \ldots, \boldsymbol{r}^{p(N)}\right)$, where $p(i)$, with $i = 1, 2, \ldots, N$, 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:

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

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

$$\pi(X) P(X, X') = \pi(X') P(X', X), \tag{5.30}$$

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:

$$A(X, X') = \min(1, \chi(X, X')), \tag{5.31}$$

where

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

If, for example, we choose to displace one bead, say $\boldsymbol{r}_j^i$, to another point, call it $\boldsymbol{r}_j^{i\prime}$, 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

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

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:

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

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:

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

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*):

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

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 exited states.

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

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

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

$$\phi(\boldsymbol{R},\beta) = \exp\left(-\beta\hat{H}\right)\phi(\boldsymbol{R},0). \tag{5.37}$$

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

$$\phi(\boldsymbol{R},\beta) = \sum_{n=0}^{\infty} c_n\phi_n(\boldsymbol{R},\beta) = \sum_{n=0}^{\infty} c_n\phi_n(\boldsymbol{R})\exp\left(-E_n\beta\right). \tag{5.38}$$

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 $\beta \to \infty$ eq. (5.38) reduces to:

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

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 $\phi_0(\boldsymbol{R})$ is real and positive definite[1]. Fermi systems and excited stated of bosons will be addressed in subsection 5.2.2.
Let us introduce the Green's function of eq. (5.36):

$$\rho(\boldsymbol{R}, \boldsymbol{R}', \beta) = \left\langle \boldsymbol{R} \left| \exp\left(-\beta\hat{H}\right) \right| \boldsymbol{R}' \right\rangle. \tag{5.40}$$

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

$$\phi(\boldsymbol{R}, \beta) = \int \rho(\boldsymbol{R}, \boldsymbol{R}', \beta)\phi(\boldsymbol{R}', 0)\mathrm{d}\boldsymbol{R}'. \tag{5.41}$$

This integral equation may be interpreted as a diffusion process guided by $\rho(\boldsymbol{R}, \boldsymbol{R}', \beta)$ from the initial state $\phi(\boldsymbol{R}', 0)$ to the final state $\phi(\boldsymbol{R}, \beta)$ 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 $\tau \to 0$, one can make use of the *primitive approximation* (see section 5.1):

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

In a DMC simulation, one treats $\phi(\boldsymbol{R}, \beta)$ 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 $\boldsymbol{R}_1$ 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 $\exp[-\tau(V(\boldsymbol{R}_2) - E)]$. 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:

$$n_d = \mathrm{int}(w + \eta), \tag{5.43}$$

[1] If 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 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 $\phi(\boldsymbol{R}, \beta \to \infty)$ converges to the exact ground state $\phi_0(\boldsymbol{R})$.

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*:

$$\phi_{\text{J}}(\boldsymbol{R}) = \prod_{i<j} f_2(|\boldsymbol{r}_i - \boldsymbol{r}_j|), \tag{5.44}$$

where the function $f_2(r)$ describes the direct correlation between particles i and j. In dilute systems, like ultracold gases, is 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 $f(\boldsymbol{R}, \beta) = \phi_{\text{T}}(\boldsymbol{R})\phi(\boldsymbol{R}, \beta)$:

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

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

$$E_{\text{loc}}(\boldsymbol{R}) = \frac{\hat{H}\phi_{\text{T}}(\boldsymbol{R})}{\phi_{\text{T}}(\boldsymbol{R})}. \tag{5.46}$$

The function $f(\boldsymbol{R}, \beta)$ is interpreted as density distribution of the population of walkers. In the long time limit it converges to the product $\phi_{\text{T}}(\boldsymbol{R})\phi_0(\boldsymbol{R})$. 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* $\left\langle \phi_0 | \hat{O} | \phi_{\text{T}} \right\rangle / \langle \phi_0 | \phi_{\text{T}} \rangle$.[2]

[2]For 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 $\boldsymbol{F}$:

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

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 $E_{\mathrm{loc}}(\boldsymbol{R})$ to the bare potential $V(\boldsymbol{R})$. 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:

$$\phi_{\mathrm{JS}}(\boldsymbol{R}) = \phi_{\mathrm{J}}(\boldsymbol{R}) \mathrm{Det}_{\alpha,n} \left[\exp\left(i \boldsymbol{k}_\alpha \cdot \boldsymbol{r}_n \right) \right], \tag{5.48}$$

where the index $n = 1, \ldots, N$ labels particles and $\boldsymbol{k}_\alpha$ 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 importanr 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

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

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

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

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

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

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

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

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

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

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

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

the ideal choice for basis functions are plane waves

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

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):

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

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

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

which cannot be solved in any closed form.

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

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

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:

$$e^{-\zeta_i|\vec{r}-\vec{R}_A|^2}e^{-\zeta_j|\vec{r}-\vec{R}_B|^2} = Ke^{-\zeta|\vec{r}-\vec{R}|^2} \tag{6.12}$$

with

$$K = e^{-\frac{\zeta_i\zeta_j}{\zeta_i+\zeta_j}|\vec{R}_A-\vec{R_B}|^2} \tag{6.13}$$

$$\zeta = \zeta_i+\zeta_j \tag{6.14}$$

$$\vec{R} = \frac{\zeta_i\vec{R}_A+\zeta_j\vec{R}_B}{\zeta_i+\zeta_j} \tag{6.15}$$

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

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

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

As there are $O(L^4)$ integrals of the type (6.4), quantum chemistry calculations typically scale as $O(N^4)$. Modern methods can be used to reduce the effort to an approximately $O(N)$ method, since the overlap of basis functions at large distances becomes negligibly small.

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

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:

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

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):

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

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.

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

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

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

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:

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

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))-1))%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

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

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:

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

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:

$$|\Phi\rangle = \prod_{\mu,\sigma} c^\dagger_{\mu\sigma}|0\rangle, \tag{6.24}$$

where $c^\dagger_{\mu\sigma}$ creates an electron in the orbital $\phi_\mu(\mathbf{r},\sigma)$. As these wave functions are orthogonal the $c^\dagger_{\mu\sigma}$ satsify 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 $c^\dagger_{\mu\sigma}$ in terms of the creation operators $\hat{a}^\dagger_{n\sigma}$ of our finite basis set:

$$c^\dagger_{\mu\sigma} = \sum_{n=1}^{L} d_{\mu n}\hat{a}^\dagger_{n\sigma} \tag{6.25}$$

and find that

$$a_{j\sigma}|\Phi\rangle = a_{j\sigma}\prod_{\mu,\sigma'} c^\dagger_{\mu\sigma'}|0\rangle = \sum_\nu d_{\nu j}\prod_{\mu\sigma'\neq\nu\sigma} c^\dagger_{\mu\sigma'}|0\rangle. \tag{6.26}$$

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

$$P_{ij} = \sum_\sigma \langle\Phi|a^\dagger_{i\sigma}a_{j\sigma}|\Phi\rangle = 2\sum_\nu d^*_{\nu i}d_{\nu j}, \tag{6.27}$$

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 $\langle\Phi|a^\dagger_{i\sigma}a^\dagger_{k\sigma'}a_{l\sigma'}a_{j\sigma}|\Phi\rangle$ in terms of the P_{ij}. We find that if $\sigma = \sigma'$

$$\langle\Phi|a^\dagger_{i\sigma}a^\dagger_{k\sigma}a_{l\sigma}a_{j\sigma}|\Phi\rangle = \langle\Phi|a^\dagger_{i\sigma}a_{j\sigma}|\Phi\rangle\langle\Phi|a^\dagger_{k\sigma}a_{l\sigma}|\Phi\rangle - \langle\Phi|a^\dagger_{i\sigma}a_{l\sigma}|\Phi\rangle\langle\Phi|a^\dagger_{k\sigma}a_{j\sigma}|\Phi\rangle \tag{6.28}$$

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

$$\langle\Phi|a^\dagger_{i\sigma}a^\dagger_{k\sigma'}a_{l\sigma'}a_{j\sigma}|\Phi\rangle = \langle\Phi|a^\dagger_{i\sigma}a_{j\sigma}|\Phi\rangle\langle\Phi|a^\dagger_{k\sigma'}a_{l\sigma'}|\Phi\rangle \tag{6.29}$$

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

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

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

$$1 = \langle\phi_\mu|\phi_\mu\rangle = \sum_{i,j} d^*_{\mu i}d_{\mu j}S_{ij}. \tag{6.31}$$

Using Lagrange multipliers to enforce this constraint we have to minimize

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

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

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

where

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

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:

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

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

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

one or two of the $c_\mu^\dagger$ are replaced by other orbitals $c_i^\dagger$:

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

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 $1 + NM + N^2M^2$. Using the Lanczos algorithm the low lying eigenstates can then be calculated in $\mathrm{O}((N+M)^2)$ 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 $\rho(\vec{r})$:

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

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:

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

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

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

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

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

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

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

Comparing this variational equation to the one for noninteracting system

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

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

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

where the exchange-correlation potential is defined by

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

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.

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

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 $\rho(\vec{r}), \nabla\rho(\vec{r}), \nabla\nabla\rho(\vec{r}), \ldots$ we ignore all the gradients and just take the local density

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

Defining

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

the exchange correlation potential is

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

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 Parinello (currently at ETH) have combined density functional theory with molecular dynamics to do just that. Their method, Car-Parinello 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:

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

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:

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

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

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

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.

Chapter 7

Cluster quantum Monte Carlo algorithms for lattice models

7.1 World line representations for quantum lattice models

All quantum Monte Carlo algorithms are based on a mapping of a d-dimensional quantum system to a $(d+1)$-dimensional classical system using a path-integral formulation. We then perform classical Monte Carlo updates on the world lines of the particles. We will introduce one modern algorithm for lattice models, the "loop-algorithm" which is a generalization of the classical cluster algorithms for the Ising model to quantum models.

We will discuss the loop algorithm for a spin-1/2 quantum XXZ model with the Hamiltonian

$$\begin{aligned} H &= -\sum_{\langle i,j\rangle} \left(J_z S_i^z S_j^z + J_{xy}(S_i^x S_j^x + S_i^y S_j^y) \right) \\ &= -\sum_{\langle i,j\rangle} \left(J_z S_i^z S_j^z + \frac{J_{xy}}{2}(S_i^+ S_j^- + S_i^- S_j^+) \right). \end{aligned} \tag{7.1}$$

For $J \equiv J_z = J_{xy}$ we have the Heisenberg model ($J > 0$ is ferromagnetic, $J < 0$ antiferromagnetic). $J_{xy} = 0$ is the (classical) Ising model and $J_z = 0$ the quantum XY model.

Continuous-time world lines

In contrast to models in continuous space, where a discrete time step was needed for the path integral, for lattice models the continuum limit can be taken. The spatial lattice is sufficient to regularize any ultraviolet divergencies.

We start by still discretizing discretize the imaginary time (inverse temperature) direction and subdivide $\beta = M\Delta\tau$:

$$e^{-\beta H} = \left(e^{-\Delta\tau H}\right)^M = (1 - \Delta\tau H)^M + O(\Delta\tau) \tag{7.2}$$

In the limit $M \to \infty$ ($\Delta\tau \to 0$) this becomes exact. We will take the limit later, but stay at finite $\Delta\tau$ for now.

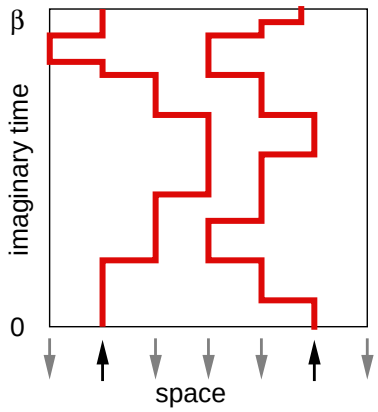


Figure 7.1: Example of a world line configuration for a spin-1/2 quantum Heisenberg model. Drawn are the world lines for up-spins only. Down spin world lines occupy the rest of the configuration.

The next step is to insert the identity matrix, represented by a sum over all basis states $1 = \sum_i |i\rangle\langle i|$ between all operators $(1 - \Delta\tau H)$:

$$\begin{aligned} Z &= \mathrm{Tr} e^{-\beta H} = \mathrm{Tr}\,(1 - \Delta\tau H)^M + O(\Delta\tau) \\ &= \sum_{i_1,\ldots,i_M} \langle i_1|1 - \Delta\tau H|i_2\rangle\langle i_2|1 - \Delta\tau H|i_3\rangle \cdots \langle i_M|1 - \Delta\tau H|i_1\rangle + O(\Delta\tau) \\ &=: P_{i_1,\ldots,i_M} \end{aligned} \tag{7.3}$$

and similarly for the measurement, obtaining

$$\langle A\rangle = \sum_{i_1,\ldots,i_M} \frac{\langle i_1|A(1 - \Delta\tau H)|i_2\rangle}{\langle i_1|1 - \Delta\tau H|i_2\rangle} P_{i_1,\ldots,i_M} + O(\Delta\tau). \tag{7.4}$$

If we choose the basis states $|i\rangle$ to be eigenstates of the local S^z operators we end up with an Ising-like spin system in one higher dimension. Each choice $i_1, \ldots, i_M$ corresponds to one of the possible configurations of this classical spin system. The trace is mapped to periodic boundary conditions in the imaginary time direction of this classical spin system. The probabilities are given by matrix elements $\langle i_n|1-\Delta\tau H|i_{n+1}\rangle$. We can now sample this classical system using classical Monte Carlo methods.

However, most of the matrix elements $\langle i_n|1-\Delta\tau H|i_{n+1}\rangle$ are zero, and thus nearly all configurations have vanishing weight. The only non-zero configurations are those where neighboring states $|i_n\rangle$ and $|i_{n+1}\rangle$ are either equal or differ by one of the off-diagonal matrix elements in H, which are nearest neighbor exchanges by two opposite spins. We can thus uniquely connect spins on neighboring "time slices" and end up with world lines of the spins, sketched in Fig. 7.1. Instead of sampling over all configurations of local spins we thus have to sample only over all world line configurations (the others have vanishing weight). Our update moves are not allowed to break world lines but have to lead to new valid world line configurations.

Finally we take the continuous time limit $\Delta\tau \to 0$. Instead of storing the configurations at all $M \to \infty$ time steps, we will not just store the times τ where a spin flips as

the consequence of an off-diagonal operator acting at that time. The number of such events stays finite as $M \to \infty$: as can be seen from equation (7.2) the probability of H acting at a given time is proportional to $1/M$, and hence the total number of spin flips will stay finite although $M \to \infty$

7.1.1 The stochastic series expansion (SSE)

An alternative representation, which is easier to code, is the stochastic series expansion (SSE) method, developed by Anders Sandvik.[1] We start from a high temperature series expansion of the partition function, which is just a Taylor series

$$Z = \mathrm{Tr} e^{-\beta H} = \sum_{n=0}^{\infty} \sum_{\alpha} \frac{\beta^n}{n!} \langle \alpha | (-H)^n | \alpha \rangle \tag{7.5}$$

where $\{|\alpha\rangle\}$ now denotes the basis states.[2] The next step is to decompose the Hamiltonian into a sum of bond Hamiltonians

$$H = \sum_{b=1}^{M} H_b, \tag{7.6}$$

where each term H_b is associated with one of the M bonds b of the lattice.

We can assume that the Hamiltonian is a sum of bond terms without loss of generality. Any single-site terms, such as a local magnetic field $-hS_i^z$ can be assigned to bond terms connected to the site.

Inserting this decomposition of the Hamiltonian and sets $|\alpha(p)\rangle$ of basis vectors into the partition function we obtain

$$Z = \sum_{n=0}^{\infty} \sum_{\{C_n\}} \sum_{\alpha(0),\ldots,\alpha(n)} \frac{\beta^n}{n!} \prod_{p=1}^{n} \langle \alpha(p) | (-H_{b_p}) | \alpha(p-1) \rangle, \tag{7.7}$$

where $\{C_n\}$ denotes the set of all concatenations of n bond Hamiltonians H_b, each called an operator string. Also, $|\alpha(n)\rangle \equiv |\alpha(0)\rangle$, reflecting the periodicity in the propagation direction given by the trace.

For a finite system and at finite temperature the relevant exponents of this power series are centered around $n\rangle \propto N_s\beta$, where N_s is the number of lattice sites. Hence we can truncate the infinite sum over n at a finite cut-off length $\Lambda \propto N_s\beta$ without introducing any systematic error for practical computations. The best value for Λ can be determined and adjusted during the equilibration part of the simulation, e.g. by setting $\Lambda > (4/3)n$ after each update step.

Instead of working with a variable number of operators n, the code is simpler[3] if we keep a fixed number of operators Λ. This can be done by inserting $(\Lambda - n)$ unit

[1] A.W. Sandvik and J. Kurkijärvi, Phys. Rev. B **43**, 5950 (1991); A. W. Sandvik, J. Phys. A **25**, 3667 (1992).

[2] We switch notation here to stay consistent with the notation used in papers on the SSE method

[3] We actually recently found an way to write a simple code even with variable n, but that method is not written up yet.

operators Id into every operator string of length $n < \Lambda$, and we define $H_0 = \mathrm{Id}$. Taking the number of such possible insertions into account, we obtain

$$Z = \sum_{n=0}^{\Lambda} \sum_{\{C_\Lambda\}} \sum_{\alpha(0),\ldots,\alpha(\Lambda)} \frac{\beta^n (\Lambda - n)!}{\Lambda!} \prod_{p=1}^{\Lambda} \langle \alpha(p)| - H_{b_p} |\alpha(p-1)\rangle, \tag{7.8}$$

where n now denotes the number of non-unity operators in the operator string C_Λ. Each such operator string is thus given by an index sequence $C_\Lambda = (b_1, b_2, ..., b_\Lambda)$, where on each propagation level $p = 1, ..., \Lambda$ either $b_p = 0$ for an unit operator, or $1 \leq b_p \leq M$ for a bond Hamiltonian.

Positivity of the weights

In order to interpret these weights as probabilities, all the matrix elements of $-H_b$ need to be positive, and hence all matrix elements of H_b should be negative. For the diagonal part of the Hamiltonian, one can assure this by subtracting a suitable constant C from each bond Hamiltonian.

For the off-diagonal part of the Hamiltonian an equally simple remedy does not exist. However, if only operator strings with an even number of negative matrix elements have a finite contribution to Eq. (7.8), the relative weights are again well suited to define a probability distribution. One can show that this is in general the case for bosonic models, ferromagnetic spin models, and antiferromagnetic spin models on bipartite lattices (such as the square lattice).

Representing the configuration

In order to represent the configurations $|\alpha(p)\rangle$ we do not need to store the full configuration at avery step p, but is is sufficient to store:

- the initial state $|\alpha(0)\rangle$
- and for each level p
  - a flag indicating whether the operator at a level p is the identity, or a diagonal or off-diagonal operator
  - bond b_p at which each operator p acts
  - the new state on the two sites of the bond b_p.

7.2 Cluster updates

Before discussing cluster updates for quantum systems in continuous time or SSE representations, we will review the cluster algorithms for the classical Ising model which should be known from the computational statistical physics course.

7.2.1 Kandel-Domany framework

To provide a general framework, which can be extended to quantum systems, 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

$$Z = \sum_{C \in \mathcal{C}} \sum_{G \in \mathcal{G}} W(C, G) \tag{7.9}$$

where the new weights $W(C, G) > 0$ are chosen such that Z is the partition function of the original model by requiring

$$\sum_{G \in \mathcal{G}} W(C, G) = W(C) := \exp(-\beta E[C]), \tag{7.10}$$

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

$$P_C(G) = W(C, G)/W(C). \tag{7.11}$$

Then we choose a new configuration C' with probability $p[(C, G) \to (C', G)]$, keeping the graph G fixed; next a new graph G' is chosen

$$C \to (C, G) \to (C', G) \to C' \to (C', G') \to \dots \tag{7.12}$$

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:

$$W(C, G)p[(C, G) \to (C', G)] = W(C', G)p[(C', G) \to (C, G)], \tag{7.13}$$

which can be fulfilled using either the heat bath algorithm

$$p[(C, G) \to (C', G)] = \frac{W(C', G)}{W(C, G) + W(C', G)} \tag{7.14}$$

or by again using the Metropolis algorithm:

$$p[(C, G) \to (C', G)] = \min(W(C', G)/W(C, G), 1) \tag{7.15}$$

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

$$W(C, G) = \Delta(C, G)V(G), \tag{7.16}$$

where

$$\Delta(C, G) := \begin{cases} 1 \text{ if } W(C, G) \neq 0, \\ 0 \text{ otherwise.} \end{cases} . \tag{7.17}$$

Then equation (7.14) simply becomes $p = 1/2$ and equation (7.15) reduces to $p = 1$ for any configuration C' with $W(C', G) \neq 0$.

Table 7.1: 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	$\exp(-\beta J)$
$\Delta(c, \text{con.})$	1	0	0	1	$\exp(\beta J) - \exp(-\beta J)$
w(c)	$\exp(\beta J)$	$\exp(-\beta J)$	$\exp(-\beta J)$	$\exp(\beta J)$	

7.2.2 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:

$$W(C) = \prod_b w(C_b) \tag{7.18}$$

$$W(C, G) = \prod_b w(C_b, G_b) = \prod_b \Delta(C_b, G_b) V(G_b) \tag{7.19}$$

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 7.1 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 (7.10) 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 $p[(C, G) \to (C', G)]$ are configuration independent constants.

There are two variants of cluster algorithms that can be constructed using the rules derived above.

The Swendsen-Wang algorithm

The Swendsen-Wang or multi-cluster algorithm proceeds as follows:

i) 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.

ii) Next a cluster labeling algorithm, like the Hoshen-Kopelman algorithm is used to identify clusters of connected spins.

iii) Measurements are performed, using improved estimators discussed in the next section.

iv) Each cluster of spins is flipped with probability 1/2.

The Wolff single-cluster 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 preferedly 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. Away from criticality sometimes a hybrid method, mixing cluster updates and local updates can be the ideal method.

7.2.3 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 $\langle \sigma_{\vec{i}} \rangle$ 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

$$\langle m \rangle = \langle \frac{1}{2}(\sigma_{\vec{i}} - \sigma_{\vec{i}}) \rangle = 0. \tag{7.20}$$

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:

$$\langle \sigma_{\vec{i}} \sigma_{\vec{j}} \rangle = \begin{cases} 1 & \text{if } \vec{i} \text{ und } \vec{j} \text{ are on the same cluster} \\ 0 & \text{otherwise} \end{cases} \tag{7.21}$$

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

$$\langle m^2 \rangle = \frac{1}{N^2} \sum_{\vec{i},\vec{j}} \langle \sigma_{\vec{i}} \sigma_{\vec{j}} \rangle = \frac{1}{N^2} \langle \sum_{cluster} S(cluster)^2 \rangle, \tag{7.22}$$

where $S(cluster)$ is the number of spins in a cluster. 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 (7.22) can be rewritten to be useful also in case of the Wolff algorithm:

$$\begin{aligned} \langle m^2 \rangle &= \frac{1}{N^2} \langle \sum_{cluster} S(cluster)^2 \rangle \\ &= \frac{1}{N^2} \sum_{\vec{i}} \frac{1}{S(\text{cluster containing } \vec{i})} S(\text{cluster containing } \vec{i})^2 \\ &= \frac{1}{N^2} \sum_{\vec{i}} S(\text{cluster containing } \vec{i}) = \frac{1}{N} \langle S(\text{cluster}) \rangle. \end{aligned} \tag{7.23}$$

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.

7.2.4 The loop algorithm for quantum spins

We will now generalize these cluster algorithms to quantum systems and present the loop algorithm [4]

[4] H. G. Evertz et al., Phys. Rev. Lett. **70**, 875 (1993); B. B. Beard and U.-J. Wiese, Phys. Rev. Lett. **77**, 5130 (1996); B. Ammon, H. G. Evertz, N. Kawashima, M. Troyer and B. Frischmuth, Phys. Rev. B **58**, 4304 (1998).

Table 7.2: The six local configurations for an XXZ model and their weights.

configuration	weight
$S_i(\tau+d\tau)\uparrow$ $S_j(\tau+d\tau)\uparrow$ $S_i(\tau+d\tau)\downarrow$ $S_j(\tau+d\tau)\downarrow$ $S_i(\tau)\uparrow$ $S_j(\tau)\uparrow$, $S_i(\tau)\downarrow$ $S_j(\tau)\downarrow$	$1+\frac{J_z}{4}d\tau$
$S_i(\tau+d\tau)\uparrow$ $S_j(\tau+d\tau)\downarrow$ $S_i(\tau+d\tau)\downarrow$ $S_j(\tau+d\tau)\uparrow$ $S_i(\tau)\uparrow$ $S_j(\tau)\downarrow$, $S_i(\tau)\downarrow$ $S_j(\tau)\uparrow$	$1-\frac{J_z}{4}d\tau$
$S_i(\tau+d\tau)\downarrow$ $S_j(\tau+d\tau)\uparrow$ $S_i(\tau+d\tau)\uparrow$ $S_j(\tau+d\tau)\downarrow$ $S_i(\tau)\uparrow$ $S_j(\tau)\downarrow$, $S_i(\tau)\downarrow$ $S_j(\tau)\uparrow$	$\frac{J_{xy}}{2}d\tau$

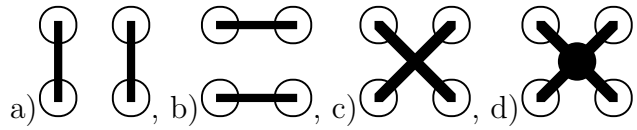


Figure 7.2: The four local graphs: a) vertical, b) horizontal c) crossing and d) freezing (connects all four corners).

The loop algorithm for continuous time world lines

This algorithm is best described by first taking the continuous time limit $M \to \infty$ ($\Delta\tau \to d\tau$) and by working with infinitesimals. Similar to the Ising model we look at two spins on neigboring sites i and j at two neighboring times τ and $\tau + d\tau$, as sketched in Tab. 7.2. There are a total of six possible configurations, having three different probabilities. The total probabilities are the products of all local probabilities, like in the classical case. This is obvious for different time slices. For the same time slice it is also true since, denoting by H_{ij} the term in the Hamiltonian H acting on the bond between sites i and j we have $\prod_{\langle i,j\rangle}(1 - d\tau H_{ij}) = 1 - d\tau \sum_{\langle i,j\rangle} H_{ij} = 1 - d\tau H$. In the following we focus only on such local four-spin plaquettes. Next we again use the Kandel-Domany framework and assign graphs. As the updates are not allowed to break world lines only four graphs, sketched in Fig. 7.2 are allowed. Finally we have to find Δ functions and graph weights that give the correct probabilities. The solution for the XY-model, ferromagnetic and antiferromagnetic Heisenberg model and the Ising model is shown in Tables 7.3 - 7.6.

Let us first look at the special case of the Ising model. As the exchange term is absent in the Ising model all world lines run straight and can be replaced by classical spins. The only non-trivial graph is the "freezing", connecting two neighboring world lines. Integrating the probability that two neighboring sites are *nowhere* connected along the time direction we obtain: times:

$$\prod_{\tau=0}^{\beta}(1 - d\tau J/2) = \lim_{M\to\infty}(1 - \Delta\tau J/2)^M = \exp(-\beta J/2) \tag{7.24}$$

Taking into account that the spin is $S = 1/2$ and the corresponding classical coupling $J_{cl} = S^2 J = J/4$ we find for the probability that two spins are *connected*:

Table 7.3: The graph weights for the quantum-XY model and the Δ function specifying whether the graph is allowed. The dash – denotes a graph that is not possible for a configuration because of spin conservation and has to be zero.

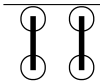
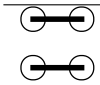
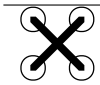
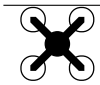
G	$\Delta(\uparrow\uparrow, G) = \Delta(\downarrow\downarrow, G)$	$\Delta(\uparrow\downarrow, G) = \Delta(\downarrow\uparrow, G)$	$\Delta(\downarrow\uparrow, G) = \Delta(\uparrow\downarrow, G)$	graph weight
	1	1	–	$1 - \frac{J_{xy}}{4} d\tau$
	–	1	1	$\frac{J_{xy}}{4} d\tau$
	1	–	1	$\frac{J_{xy}}{4} d\tau$
	0	0	0	0
total weight	1	1	$\frac{J_{xy}}{2} d\tau$	

Table 7.4: The graph weights for the ferromagnetic quantum Heisenberg model and the Δ function specifying whether the graph is allowed. The dash – denotes a graph that is not possible for a configuration because of spin conservation and has to be zero.

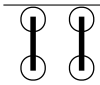
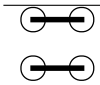
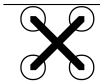
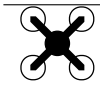
G	$\Delta(\uparrow\uparrow, G) = \Delta(\downarrow\downarrow, G)$	$\Delta(\uparrow\downarrow, G) = \Delta(\downarrow\uparrow, G)$	$\Delta(\downarrow\uparrow, G) = \Delta(\uparrow\downarrow, G)$	graph weight
	1	1	–	$1 - \frac{J}{4} d\tau$
	–	0	0	0
	1	–	1	$\frac{J}{2} d\tau$
	0	0	0	0
total weight	$1 + \frac{J}{4} d\tau$	$1 - \frac{J}{4} d\tau$	$\frac{J}{2} d\tau$	

Table 7.5: The graph weights for the antiferromagnetic quantum Heisenberg model and the Δ function specifying whether the graph is allowed. The dash – denotes a graph that is not possible for a configuration because of spin conservation and has to be zero. To avoid the sign problem (see next subsection) we change the sign of J_{xy}, which is allowed only on bipartite lattices.

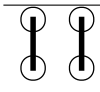
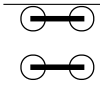
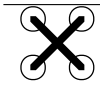
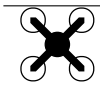
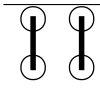
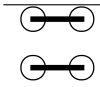
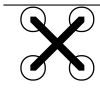
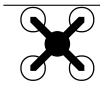
G	$\Delta(\uparrow\uparrow, \uparrow\uparrow, G)$ $= \Delta(\downarrow\downarrow, \downarrow\downarrow, G)$	$\Delta(\uparrow\downarrow, \uparrow\downarrow, G)$ $= \Delta(\downarrow\uparrow, \downarrow\uparrow, G)$	$\Delta(\downarrow\uparrow, \uparrow\downarrow, G)$ $= \Delta(\uparrow\downarrow, \downarrow\uparrow, G)$	graph weight
	1	1	–	$1 - \frac{\lvert J\rvert}{4} d\tau$
	–	1	1	$\frac{\lvert J\rvert}{2} d\tau$
	0	–	0	0
	0	0	0	0
total weight	$1 - \frac{\lvert J\rvert}{4} d\tau$	$1 + \frac{\lvert J\rvert}{4} d\tau$	$\frac{\lvert J\rvert}{2} d\tau$	

Table 7.6: The graph weights for the ferromagnetic Ising model and the Δ function specifying whether the graph is allowed. The dash – denotes a graph that is not possible for a configuration because of spin conservation and has to be zero.

G	$\Delta(\uparrow\uparrow, \uparrow\uparrow, G)$ $= \Delta(\downarrow\downarrow, \downarrow\downarrow, G)$	$\Delta(\uparrow\downarrow, \uparrow\downarrow, G)$ $= \Delta(\downarrow\uparrow, \downarrow\uparrow, G)$	$\Delta(\downarrow\uparrow, \uparrow\downarrow, G)$ $= \Delta(\uparrow\downarrow, \downarrow\uparrow, G)$	graph weight
	1	1	–	$1 - \frac{J_z}{4} d\tau$
	–	0	0	0
	0	–	0	0
	1	0	0	$\frac{J_z}{2} d\tau$
total weight	$1 + \frac{J_z}{4} d\tau$	$1 - \frac{J_z}{4} d\tau$	0	

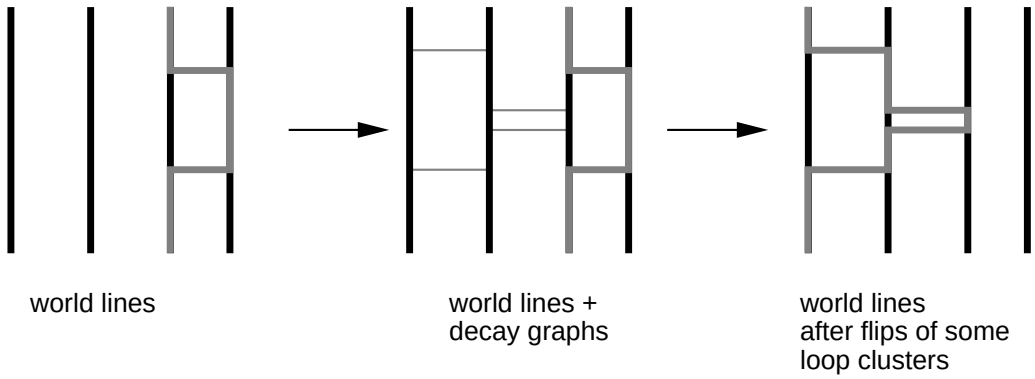


Figure 7.3: Example of a loop update. In a first step decay paths are inserted where possible at positions drawn randomly according to an exponential distribution and graphs are assigned to all exchange terms (hoppings of world lines). In a second stage (not shown) the loop clusters are identified. Finally each loop cluster is flipped with probability 1/2 and one ends up with a new configuration.

$1 - \exp(-2\beta J_{cl})$. We end up exactly with the cluster algorithm for the classical Ising model!

The other cases are special. Here each graph connects two spins. As each of these spins is again connected to only one other, all spins connected by a cluster form a closed loop, hence the name "loop algorithm". Only one issue remains to be explained: how do we assign a horizontal or crossing graph with infinitesimal probability, such as $(J/2)d\tau$. This is easily done by comparing the assignment process with radioactive decay. For each segment the graph runs vertical, except for occasional decay processes occuring with probability $(J/2)d\tau$. Instead of asking at every infinitesimal time step whether a decay occurs we simply calculate an exponentially distributed decay time t using an exponential distribution with decay constant $J/2$. Looking up the equation in the lecture notes of the winter semester we have $t = -(2/J)\ln(1-u)$ where u is a uniformly distributed random number.

The algorithm now proceeds as follows (see Fig. 7.3): for each bond we start at time 0 and calculate a decay time. If the spins at that time are oriented properly and an exchange graph is possible we insert one. Next we advance by another randomly chosen decay time along the same bond and repeat the procedure until we have reached the extent β. This assigns graphs to all infinitesimal time steps where spins do not change. Next we assign a graph to all of the (finite number of) time steps where two spins are exchanged. In the case of the Heisenberg models there is always only one possible graph to assign and this is very easy. In the next step we identify the loop-clusters and then flip them each with probability 1/2. Alternatively a Wolff-like algorithm can be constructed that only builds one loop-cluster.

Improved estimators for measurements can be constructed like in classical models. The derivation is similar to the classical models. I will just mention two simple ones

for the ferromagnetic Heisenberg model. The spin-spin corelation is

$$S_i^z(\tau)S_j^z(\tau') = \begin{cases} 1 & \text{if } (i,\tau) \text{ und } (j,\tau') \text{ are on the same cluster} \\ 0 & \text{otherwise} \end{cases} \tag{7.25}$$

and the uniform susceptibilty is

$$\chi = \frac{1}{N\beta}\sum_c S(c)^2, \tag{7.26}$$

where the sum goes over all loop clusters and $S(c)$ is the length of all the loop segments in the loop cluster c.

The loop algorithm in SSE

We now discuss the loop algorithm in the SSE representation, which we will also use in the exercises. In the SSE representations, the loop algorithm consists of two parts:

1. diagonal updates, replacing identity operators H_0 by diagonal operators H_b, without changing any of the spins.

2. loop updates, flipping spins and changing diagonal into off-diagonal operators.

The first step are the diagonal updates. We attempt to change the expansion order n by inserting and removing the number of unit operators. During this update step, all propagation levels $p = 1, ..., \Lambda$ are traversed in ascending order. If the current operator is an unit operator H_0 it is replaced by a bond Hamiltonian with a certain probability which guarantees detailed balance. The reverse process, i.e. substitution of a bond Hamiltonian by a unit operator is only attempted if the action of the current bond Hamiltonian does not change the propagated state, i.e., if $|\alpha(p)\rangle = |\alpha(p-1)\rangle$, since otherwise the resulting contribution to Eq. (7.8) would vanish.

The acceptance probabilities for both substitutions, as determined from detailed balance, are

$$\begin{aligned} P(H_0 \to H_b) &= \min\left[1, \frac{M\beta\langle\alpha(p)|H_b|\alpha(p-1)\rangle}{\Lambda - n}\right], \\ P(H_b \to H_0) &= \min\left[1, \frac{(\Lambda - n + 1)\delta_{|\alpha(p)\rangle,|\alpha(p-1)\rangle}}{M\beta\langle\alpha(p)|H_b|\alpha(p-1)\rangle}\right]. \end{aligned}$$

At the same time we build up a quadruply-linkes list of of the operators H_p, connecting to the previous and next operators on the two sites of the bond b_p. This list will be needed to construct the loop clusters

In the next step we assign graphs to each of the non-identity operators H_p, and then build clusters. Flipping the clusters will turn diagonal into off-diagonal operators and vice-versa. Since some stochastic choices were already made by replacing identity with diagonal operators, and there are no infinitesimals involved the cluster building rules are actually easier than for continuous time world lines.

For the ferromagnetic and anti-ferromagnetic Heisenberg and Ising models we can see that the graph assignments are unique - we do not even need any random numbers here but can just build the loops and flip them.

Table 7.7: The graph weights for the quantum-XY model in the SSE representation and the Δ function specifying whether the graph is allowed. An offset of $-\frac{J_{xy}}{4}$ has been added to the diagonal terms to give simple cluster rules. The dash – denotes a graph that is not possible for a configuration because of spin conservation and has to be zero.

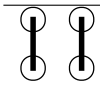
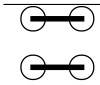
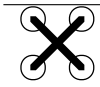
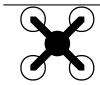
G	$\Delta(\uparrow\uparrow, G)$ $= \Delta(\downarrow\downarrow, G)$	$\Delta(\uparrow\downarrow, G)$ $= \Delta(\downarrow\uparrow, G)$	$\Delta(\downarrow\uparrow, G)$ $= \Delta(\uparrow\downarrow, G)$	graph weight
	0	0	–	0
	–	1	1	$\frac{J_{xy}}{4}$
	1	–	1	$\frac{J_{xy}}{4}$
	0	0	0	0
total weight	$\frac{J_{xy}}{4}$	$\frac{J_{xy}}{4}$	$\frac{J_{xy}}{2}$	

Table 7.8: The graph weights for the ferromagnetic quantum Heisenberg model in the SSE representation and the Δ function specifying whether the graph is allowed. The dash – denotes a graph that is not possible for a configuration because of spin conservation and has to be zero. An offset of $-J/4$ was added to the diagonal terms to keep the weights positive.

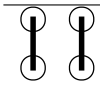
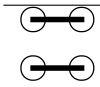
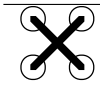
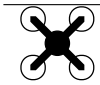
G	$\Delta(\uparrow\uparrow, G)$ $= \Delta(\downarrow\downarrow, G)$	$\Delta(\uparrow\downarrow, G)$ $= \Delta(\downarrow\uparrow, G)$	$\Delta(\downarrow\uparrow, G)$ $= \Delta(\uparrow\downarrow, G)$	graph weight
	0	0	–	0
	–	0	0	0
	1	–	1	$\frac{J}{2}$
	0	0	0	0
total weight	$\frac{J}{2}$	0	$\frac{J}{2}$	

Table 7.9: The graph weights for the antiferromagnetic quantum Heisenberg model in the SSE representation and the Δ function specifying whether the graph is allowed. The dash – denotes a graph that is not possible for a configuration because of spin conservation and has to be zero. An offset of $J/4$ was added to the diagonal terms to keep the weights positive. To avoid the sign problem for off-diagonal terms (see next subsection) we change the sign of J_{xy}, which is allowed only on bipartite lattices.

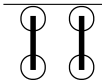
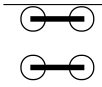
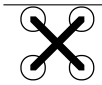
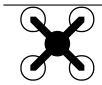
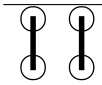
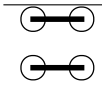
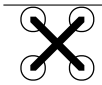
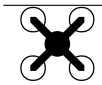
G	$\Delta(\uparrow\uparrow, \uparrow\uparrow, G)$ $= \Delta(\downarrow\downarrow, \downarrow\downarrow, G)$	$\Delta(\uparrow\downarrow, \uparrow\downarrow, G)$ $= \Delta(\downarrow\uparrow, \downarrow\uparrow, G)$	$\Delta(\downarrow\uparrow, \uparrow\downarrow, G)$ $= \Delta(\uparrow\downarrow, \downarrow\uparrow, G)$	graph weight
	0	0	–	0
	–	1	1	$\frac{\lvert J\rvert}{2}$
	0	–	0	0
	0	0	0	0
total weight	0	$\frac{\lvert J\rvert}{2}$	$\frac{\lvert J\rvert}{2}$	

Table 7.10: The graph weights for the ferromagnetic Ising model in the SSE representation and the Δ function specifying whether the graph is allowed. An offset of $-J_z/4$ was added to the diagonal terms to keep the weights positive. The dash – denotes a graph that is not possible for a configuration because of spin conservation and has to be zero.

G	$\Delta(\uparrow\uparrow, \uparrow\uparrow, G)$ $= \Delta(\downarrow\downarrow, \downarrow\downarrow, G)$	$\Delta(\uparrow\downarrow, \uparrow\downarrow, G)$ $= \Delta(\downarrow\uparrow, \downarrow\uparrow, G)$	$\Delta(\downarrow\uparrow, \uparrow\downarrow, G)$ $= \Delta(\uparrow\downarrow, \downarrow\uparrow, G)$	graph weight
	0	0	–	0
	–	0	0	0
	0	–	0	0
	1	0	0	$\frac{J_z}{2}$
total weight	$\frac{J_z}{2}$	0	0	

7.3 The negative sign problem

Now that we have an algorithm with no critical slowing down we could think that we have completely solved the problem of quantum many body problems. However, in this section we will show that the sign problem is **NP**-hard, following the paper M. Troyer and U.J. Wiese, Phys. Rev. Lett. **94**, 170201 (2005).

The difficulties in finding polynomial time solutions to the sign problem are reminiscent of the apparent impossibility to find polynomial time algorithms for nondeterministic polynomial (**NP**)-complete decision problems, which could be solved in polynomial time on a hypothetical non-deterministic machine, but for which no polynomial time algorithm is known for deterministic classical computers. A hypothetical non-deterministic machine can always follow both branches of an if-statement simultaneously, but can never merge the branches again. It can, equivalently, be viewed as having exponentially many processors, but without any communication between them. In addition, it must be possible to check a positive answer to a problem in **NP** on a classical computer in polynomial time.

Many important computational problems in the complexity class **NP**, including the traveling salesman problem and the problem of finding ground states of spin glasses have the additional property of being **NP**-hard, forming the subset of **NP**-complete problems, the hardest problems in **NP**. A problem is called **NP**-hard if any problem in **NP** can be mapped onto it with polynomial complexity. Solving an **NP**-hard problem is thus equivalent to solving any problem in **NP**, and finding a polynomial time solution to any of them would have important consequences for all of computing as well as the security of classical encryption schemes. In that case all problems in **NP** could be solved in polynomial time, and hence **NP**=**P**.

As no polynomial solution to any of the **NP**-complete problems was found despite decades of intensive research, it is generally believed that **NP**≠**P** and no deterministic polynomial time algorithm exists for these problems. The proof of this conjecture remains as one of the unsolved millennium problems of mathematics for which the Clay Mathematics Institute has offered a prize of one million US$. In this section we will show that the sign problem is **NP**-hard, implying that unless the **NP**≠**P** conjecture is disproved there exists no generic solution of the sign problem.

Before presenting the details of our proof, we will give a short introduction to classical and quantum Monte Carlo simulations and the origin of the sign problem. In the calculation of the phase space average of a quantity A, instead of directly evaluating the sum

$$\langle A\rangle = \frac{1}{Z}\sum_{c\in\Omega} A(c)p(c)\ , \quad Z=\sum_{c\in\Omega} p(c), \tag{7.27}$$

over a high-dimensional space Ω of configurations c, a classical Monte Carlo method chooses a set of M configurations $\{c_i\}$ from Ω, according to the distribution $p(c_i)$. The average is then approximated by the sample mean

$$\langle A\rangle \approx \overline{A} = \frac{1}{M}\sum_{i=1}^{M} A(c_i), \tag{7.28}$$

within a statistical error $\Delta A = \sqrt{\mathrm{Var}A(2\tau_A + 1)/M}$, where Var$A$ is the variance of A and the integrated autocorrelation time τ_A is a measure of the autocorrelations of the sequence $\{A(c_i)\}$. In typical statistical physics applications, $p(c) = \exp(-\beta E(c))$ is the Boltzmann weight, $\beta = 1/k_B T$ is the inverse temperature, and $E(c)$ is the energy of the configuration c.

Since the dimension of configuration space Ω grows linearly with the number N of particles, the computational effort for the direct integration Eq. (7.27) scales exponentially with the particle number N. Using the Monte Carlo approach the same average can be estimated to any desired accuracy in *polynomial time*, as long as the autocorrelation time τ_A does not increase faster than polynomially with N.

In a quantum system with Hamilton operator H, instead of an integral like Eq. (7.27), an operator expression

$$\langle A \rangle = \frac{1}{Z} \mathrm{Tr}[A \exp(-\beta H)] \ , \quad Z = \mathrm{Tr} \exp(-\beta H) \tag{7.29}$$

needs to be evaluated in order to calculate the thermal average of the observable A (represented by a self-adjoint operator). Monte Carlo techniques can again be applied to reduce the exponential scaling of the problem, but only after mapping the quantum model to a classical one. One approach to this mapping[5] is a Taylor expansion:

$$\begin{aligned} Z &= \mathrm{Tr} \exp(-\beta H) = \sum_{n=0}^{\infty} \frac{(-\beta)^n}{n!} \mathrm{Tr} H^n \\ &= \sum_{n=0}^{\infty} \sum_{i_1,\ldots,i_n} \frac{(-\beta)^n}{n!} \langle i_1|H|i_2\rangle\langle i_2|H|i_3\rangle \cdots \langle i_n|H|i_1\rangle \\ &\equiv \sum_{n=0}^{\infty} \sum_{i_1,\ldots,i_n} p(i_1, ..., i_n) \equiv \sum_c p(c), \end{aligned} \tag{7.30}$$

where for each order n in the expansion we insert n sums over complete sets of basis states $\{|i\rangle\}$. The "configurations" are sequences $c = (i_1, ..., i_n)$ of n basis states and we define the weight $p(c)$ by the corresponding product of matrix elements of H and the term $(-\beta)^n/n!$. With a similar expansion for $\mathrm{Tr}[A \exp(-\beta H)]$ we obtain an expression reminiscent of classical problems:

$$\langle A \rangle = \frac{1}{Z} \mathrm{Tr}[A \exp(-\beta H)] = \frac{1}{Z} \sum_c A(c) p(c). \tag{7.31}$$

If all the weights $p(c)$ are positive, standard Monte Carlo methods can be applied, as it is the case for non-frustrated quantum magnets and bosonic systems. In fermionic systems negative weights $p(c) < 0$ arise from the Pauli exclusion principle, when along the sequence $|i_1\rangle \to |i_2\rangle \to \cdots \to |i_n\rangle \to |i_1\rangle$ two fermions are exchanged, as shown in Fig. 1.

The standard way of dealing with the negative weights of the fermionic system is to sample with respect to the bosonic system by using the absolute values of the weights

[5]Note that the conclusions are independent of the representation

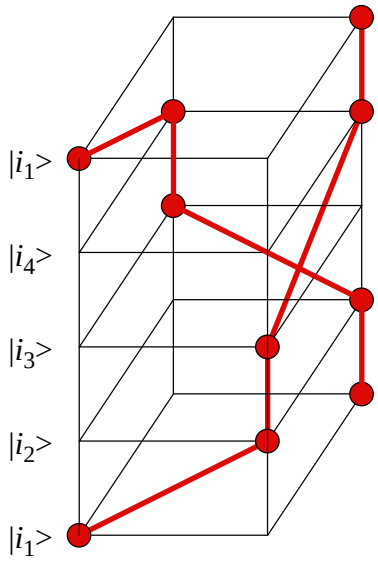


Figure 7.4: A configuration of a fermionic lattice model on a 4-site square. The configuration has negative weight, since two fermions are exchanged in the sequence $|i_1\rangle \to |i_2\rangle \to |i_3\rangle \to |i_4\rangle \to |i_1\rangle$. World lines connecting particles on neighboring slices are drawn as thick lines.

$|p(c)|$ and to assign the sign $s(c) \equiv \operatorname{sign} p(c)$ to the quantity being sampled:

$$\begin{aligned} \langle A \rangle &= \frac{\sum_c A(c)p(c)}{\sum_c p(c)} \\ &= \frac{\sum_c A(c)s(c)|p(c)| \,/ \sum_c |p(c)|}{\sum_c s(c)|p(c)| \,/ \sum_c |p(c)|} \equiv \frac{\langle As \rangle'}{\langle s \rangle'}. \end{aligned} \tag{7.32}$$

While this allows Monte Carlo simulations to be performed, the errors increase exponentially with the particle number N and the inverse temperature β. To see this, consider the mean value of the sign $\langle s \rangle = Z/Z'$, which is just the ratio of the partition functions of the fermionic system $Z = \sum_c p(c)$ with weights $p(c)$ and the bosonic system used for sampling with $Z' = \sum_c |p(c)|$. As the partition functions are exponentials of the corresponding free energies, this ratio is an exponential of the differences Δf in the free energy densities:$\langle s \rangle = Z/Z' = \exp(-\beta N \Delta f)$. As a consequence, the relative error $\Delta s/\langle s \rangle$ increases exponentially with increasing particle number and inverse temperature:

$$\frac{\Delta s}{\langle s \rangle} = \frac{\sqrt{\left(\langle s^2 \rangle - \langle s \rangle^2\right)/M}}{\langle s \rangle} = \frac{\sqrt{1 - \langle s \rangle^2}}{\sqrt{M}\langle s \rangle} \sim \frac{e^{\beta N \Delta f}}{\sqrt{M}}. \tag{7.33}$$

Similarly the error for the numerator in Eq. (7) increases exponentially and the time needed to achieve a given relative error scales exponentially in N and β.

In order to avoid any misconception about what would constitute a "solution" of the sign problem, we start by giving a precise definition:

- A quantum Monte Carlo simulation to calculate a thermal average $\langle A \rangle$ of an observable A in a quantum system with Hamilton operator H is defined to suffer

from a *sign problem* if there occur negative weights $p(c) < 0$ in the classical representation as given by Eq. (7.31).

- The related *bosonic system* of a fermionic quantum system is defined as the system where the weights $p(c)$ are replaced by their absolute values $|p(c)|$, thus ignoring the minus sign coming from fermion exchanges:

$$\langle A\rangle' = \frac{1}{Z'}\sum_c A(c)|p(c)|. \tag{7.34}$$

- An algorithm for the stochastic evaluation of a thermal average such as Eqns. (7.31) or (7.34) is defined to be of *polynomial complexity* if the computational time $t(\epsilon, N, \beta)$ needed to achieve a relative statistical error $\epsilon = \Delta A/\langle A\rangle$ in the evaluation of the average $\langle A\rangle$ scales polynomially with the system size N and inverse temperature β, i.e. if there exist integers n and m and a constant $\kappa < \infty$ such that

$$t(\epsilon, N, \beta) < \kappa\epsilon^{-2}N^n\beta^m. \tag{7.35}$$

- For a quantum system that suffers from a sign problem for an observable A, and for which there exists a polynomial complexity algorithm for the related bosonic system Eq. (7.34), we define a *solution of the sign problem* as an algorithm of polynomial complexity to evaluate the thermal average $\langle A\rangle$.

It is important to note that we only worry about the sign problem if the bosonic problem is easy (of polynomial complexity) but the fermionic problem hard (of exponential complexity) due to the sign problem. If the bosonic problem is already hard, e.g. for spin glasses [6], the sign problem will not increase the complexity of the problem. Also, changing the representation so that the sum in Eq. (7.31) contains only positive terms $p(c) \geq 0$ is not sufficient to solve the sign problem if the scaling remains exponential, since then we just map the sign problem to another exponentially hard problem. Only a polynomial complexity algorithm counts as a solution of the sign problem.

At first sight such a solution seems feasible since the sign problem is not an intrinsic property of the quantum model studied but is representation-dependent: it depends on the choice of basis sets $\{|i\rangle\}$, and in some models it can be solved by a simple local basis change.. Indeed, when using the eigen basis in which the Hamilton operator H is diagonal, there will be no sign problem. This diagonalization of the Hamilton operator is, however, no solution of the sign problem since its complexity is exponential in the number of particles N.

We now construct a quantum mechanical system for which the calculation of a thermal average provides the solution for one and thus all of the **NP**-complete problems. This system exhibits a sign problem, but the related bosonic problem is easy to solve. Since, for this model, a solution of the sign problem would provide us with a polynomial time algorithm for an **NP**-complete problem, the sign problem is **NP**-hard. Of course, it is expected that the corresponding thermal averages cannot be calculated in polynomial time and the sign problem thus cannot be solved. Otherwise we would have found a

[6] F. Barahona, J. Phys. A **15**, 3241 (1982)

polynomial time algorithm for the **NP**-complete problems and would have shown that **NP**=**P**.

The specific **NP**-complete problem we consider is to determine whether a state with energy less than or equal to a bound E_0 exists for a classical three-dimensional Ising spin glass with Hamilton function

$$H = -\sum_{\langle j,k \rangle} J_{jk}\sigma_j\sigma_k. \tag{7.36}$$

Here the spins σ_j take the values ± 1, and the couplings J_{jk} between nearest neighbor lattice points j and k are either 0 or $\pm J$.

This problem is in the complexity class **NP** since the non-deterministic machine can evaluate the energies of all configurations c in polynomial time and test whether there is one with $E(c) \leq E_0$. In addition, the validity of a positive answer (i.e. there is a configuration c) can be tested on a deterministic machine by evaluating the energy of that configuration. The evaluation of the partition function $Z = \sum_c \exp(-\beta E(c))$ is, however, not in **NP** since the non-deterministic machine cannot perform the sum in polynomial time.

This question whether there is a state with energy $E(c) \leq E_0$ can also be answered in a Monte Carlo simulation by calculating the average energy of the spin glass at a large enough inverse temperature β. Since the energy levels are discrete with spacing J it can easily be shown that by choosing an inverse temperature $\beta J \geq N \ln 2 + \ln(12N)$ the thermal average of the energy will be less than $E_0 + J/2$ if at least one configuration with energy E_0 or less exists, and larger than $E_0 + J$ otherwise.

In this classical Monte Carlo simulation, the complex energy landscape, created by the frustration in the spin glass (Fig. 2a), exponentially suppresses the tunneling of the Monte Carlo simulation between local minima at low temperatures. The autocorrelation times and hence the time complexity of this Monte Carlo approach are exponentially large $\tau \propto \exp(aN)$, as expected for this **NP**-complete problem.

We now map this classical system to a quantum system with a sign problem. We do so by replacing the classical Ising spins by quantum spins. Instead of the common choice in which the classical spin configurations are basis states and the spins are represented by diagonal σ_j^z Pauli matrices we choose a representation in which the spins point in the $\pm x$ direction and are represented by σ_j^x Pauli matrices:

$$H = -\sum_{\langle j,k \rangle} J_{jk}\sigma_j^x\sigma_k^x, \tag{7.37}$$

Here the random signs of the couplings are mapped to random signs of the off-diagonal matrix elements which cause a sign problem (see Fig. 2b). The related bosonic model is the ferromagnet with all couplings $J_{jk} \geq 0$ and efficient cluster algorithms with polynomial time complexity are known for this model. Since the bosonic version is easy to simulate, the *sign problem is the origin of the* **NP**-*hardness* of a quantum Monte Carlo simulation of this model. A generic solution of the sign problem would provide a polynomial time solution to this, and thus to all, **NP**-complete problems, and would hence imply that **NP**=**P**. Since it is generally believed that **NP**$\neq$**P**, we expect that such a solution does not exist.

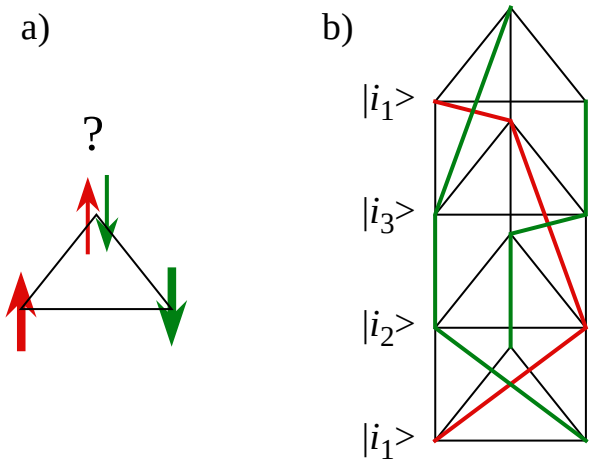


Figure 7.5: a) A classically frustrated spin configuration of three antiferromagnetically coupled spins: no configuration can simultaneously minimize the energy of all three bonds. b) A configuration of a frustrated quantum magnet with negative weights: three antiferromagnetic exchange terms with negative weights are present in the sequence $|i_1\rangle \rightarrow |i_2\rangle \rightarrow |i_3\rangle \rightarrow |i_1\rangle$. Here up-spins with z-component of spin $\sigma_j^z = 1$ and down-spins with $\sigma_j^z = -1$ are connected with differently colored world lines.

By constructing a concrete model we have shown that the sign problem of quantum Monte Carlo simulations is **NP**-hard. This does not exclude that a specific sign problem can be solved for a restricted subclass of quantum systems. This was indeed possible using the meron-cluster algorithm [7] for some particular lattice models. Such a solution must be intimately tied to properties of the physical system and allow an essentially bosonic description of the quantum problem. A generic approach might scale polynomially for some cases but will in general scale exponentially.

In the case of fermions or frustrated quantum magnets, solving the sign problem requires a mapping to a bosonic or non-frustrated system – which is, in general, almost certainly impossible for physical reasons. The origin of the sign problem is, in fact, the distinction between bosonic and fermionic systems. The brute-force approach of taking the absolute values of the probabilities means trying to sample a frustrated or fermionic system by simulating a non-frustrated or bosonic one. As for large system sizes N and low temperatures the relevant configurations for the latter are not the relevant ones for the former, the errors are exponentially large.

Given the **NP**-hardness of the sign problem one promising idea for the simulation of fermionic systems is to use ultra-cold atoms in optical lattices to construct well-controlled and tunable implementations of physical systems, such as the Hubbard model, and to use these "quantum simulators" to study the phase diagrams of correlated quantum systems. But even these quantum simulators are most likely not a generic solution to the sign problem since there exist quantum systems with exponentially diverging time scales and it is at present not clear whether a quantum computer

[7] S. Chandrasekharan and U.-J. Wiese, Phys. Rev. Lett. **83**, 3116 (1999)

could solve the **NP**-complete problems.

7.4 Worm and directed loop updates

The quantum loop algorithm, like the classical cluster algorithms work only for models with spin-inversion symmetry: a configuration of spins and the flipped configuration need to have the same weight. Applying a magnetic field breaks this spin-inversion symmetry, and the loop algorithm cannot be applied. For that case, Nikolay Prokof'ev and coworkers invented the worm algorithm.[8] The worm algorithm is based on a very simple idea: while local updates on world lines are not ergodic (since knots cannot be created or undone), the worm algorithm proceeds by cutting a world line, and then moves the open ends in local updates until they meet again and the world line is glued together once more.

[8] N. V. Prokof'ev, B. V. Svistunov and I. S. Tupitsyn, Sov. Phys - JETP **87**, 310 (1998).

Chapter 8

An Introduction to Quantum Field Theory

8.1 Introduction and motivation

This chapter is a generalization of Sec. 5.1 on Path Integral Monte Carlo. Instead of considering one (or N) quantum-mechanical particles as was done there, the idea is now to consider a quantum *field*, which contains infinitely many degrees of freedom. However, in practice, we are going to simulate a large but finite number of degrees of freedom, and extrapolate at the end. So there really is not much difference with Sec. 5.1.

The formal basis for Quantum Field Theory is an important subject, which will not be covered here. The goal of this chapter is to convey some understanding of *simulations* of quantum field theories, by appealing to intuition rather than rigor.

Let us mention at least three reasons why numerical simulations of Quantum Field Theory are interesting:

• They represent a complementary approach to the traditional, analytic perturbative expansion. A perturbative expansion is useful when the theory possesses a small expansion parameter, which ensures fast convergence of the Taylor series. This is the case for QED (quantum electrodynamics), where the coupling constant characterizing the strength of the interactions is $\alpha_{QED} \approx \frac{1}{137}$. For QCD (quantum chromodynamics) which describes the interactions of quarks, the coupling constant is $\alpha_{QCD} \sim \mathcal{O}(1)$, which makes numerical simulations the method of choice, because they are "non-perturbative" (i.e. not based on a perturbative expansion).

• Quantum field theories are complicated by divergences, infinities which arise when interactions occur at very short distances. These UV divergences cancel out of all measurable quantities. This process of "renormalization" of infinities is traditionally considered in a perturbative expansion framework. The numerical simulations of QFT are performed on a space-time lattice, where the lattice spacing a naturally defines the smallest distance scale, thus providing an elegant formalism to study renormalization as $a \to 0$.

• Solid-state physicists, who deal with crystalline materials, may think that they have no need for quantum field theory which is formulated in continuous, uniform space-

time. This is not true. A topical example is graphene, which is a $2d$ honeycomb crystal. Electrons whose momenta $\vec{p}$ are special, close to two particular values $\vec{p_\pm}$, behave like relativistic particles with momentum $\vec{p}-\vec{p_\pm}$. Their properties for small values of $|\vec{p}-\vec{p_\pm}|$ are well described by an *effective* quantum field theory. Because the momenta considered are small, or equivalently the distances are large, the crystalline nature of graphene is irrelevant. The effective quantum field theory is formulated in the continuum, and can be studied by numerical simulations on, say, a square lattice.

8.2 Recall: Path integral Monte Carlo

The starting point is the Path Integral formalism, introduced in Chapter 5. Let us recall the main idea and some results from Sec. 5.1, specialized to the simplest case of one bosonic particle of mass m moving in one dimension in a potential $V(x)$.

The partition function $Z = \mathrm{Tr}\left(\exp\left(-\beta H\right)\right)$ is the trace of the density matrix $\rho = \exp\left(-\beta H\right)$, with $H = T + V$, $T = -\frac{\hbar^2}{2m}\Delta$, $V = V(x)$. The difficulty in evaluating Z is that the kinetic and potential energy operators T and V are diagonal in momentum space and in coordinate space, respectively, and thus do not commute. To evaluate Z, one divides β into M elementary "time steps" $\hat{\varepsilon} = \beta/M$, and uses the *primitive approximation* for the high-temperature ($T = 1/\hat{\varepsilon}$) density matrix $\exp\left(-\hat{\varepsilon}H\right) \cong \exp\left(-\hat{\varepsilon}T\right)\exp\left(-\hat{\varepsilon}V\right)$, so that $\exp\left(-\beta\left(T+V\right)\right) = \lim_{M\to+\infty}\left[\exp\left(-\hat{\varepsilon}T\right)\exp\left(-\hat{\varepsilon}V\right)\right]^M$. Finally, the matrix elements of $\exp\left(-\hat{\varepsilon}T\right), \hat{\varepsilon} \ll 1$ can be evaluated in coordinate space, yielding

$$\rho\left(x_1, x_{M+1}, \beta\right) = \lim_{M\to+\infty} \int \prod_{j=2}^{M} \mathrm{d}x_j \prod_{j=1}^{M} \left\{\rho^{\mathrm{free}}\left(x_j, x_{j+1}, \hat{\varepsilon}\right)\exp\left[-\hat{\varepsilon}V\left(x_j\right)\right]\right\} \tag{8.1}$$

with $\rho^{\mathrm{free}}\left(x_j, x_{j+1}, \hat{\varepsilon}\right) = \left(2\pi\hbar^2\hat{\varepsilon}/m\right)^{-1/2}\exp\left[-\frac{\left(x_j - x_{j+1}\right)^2}{2\hbar^2\hat{\varepsilon}/m}\right]$. See eq.(5.18).

Here, we want to emphasize that the density matrix ρ is also the Green's function in imaginary time, as explained in Sec. 5.1.1. The solution of the time-dependent Schrödinger equation $i\hbar\frac{\partial}{\partial t}\psi = H\psi$ is

$$\psi(x,t) = \int dx_0\ G(x,t;x_0,0)\ \psi(x_0,0) \tag{8.2}$$

where

$$G(x,t;x_0,0) \equiv \langle x|\exp(-\frac{i}{\hbar}Ht)|x_0\rangle \tag{8.3}$$

is the Green's function, i.e. the solution of the Schrödinger equation for $\psi(x,0) = \delta(x-x_0)$ (also called transition amplitude, or matrix element). If we now introduce an *imaginary* time $\tau \equiv it$, we see that the Green's function $G(x,\tau;x_0,0)$ is *identical* to the element of the density matrix $\rho(x,x_0;\tau/\hbar)$ (Changing time to pure imaginary is also called performing a Wick rotation to Euclidean time).

This equivalence helps understand several aspects of the path integral formalism:

• Propagation over a large imaginary time τ isolates the groundstate: the operator

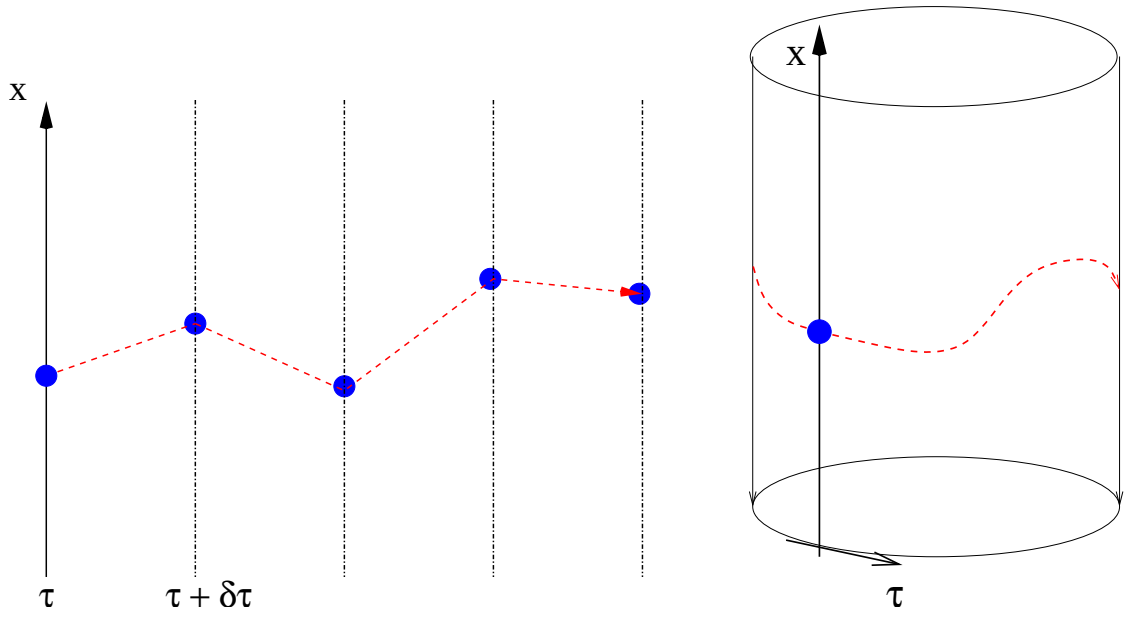


Figure 8.1: After compactification of the Euclidean time direction, the paths become closed loops, and the path integral is identical to the partition function of the quantum mechanical system, where the inverse temperature is (proportional to) the Euclidean time-extent.

$\exp(-\frac{\tau}{\hbar}H)$ is applied to the initial state $|\psi(0)\rangle$, so that

$$|\psi(\tau)\rangle = \exp(-\frac{\tau}{\hbar}H)|\psi(0)\rangle \tag{8.4}$$

If one expands in eigenstates of H: $H|\psi_k\rangle = E_k|\psi_k\rangle, k = 0, 1, .., E_0 \leq E_1 \leq ..$, then

$$|\psi(\tau)\rangle = \sum_k \exp(-\frac{\tau}{\hbar}E_k)\langle\psi_k|\psi(0)\rangle|\psi_k\rangle \quad . \tag{8.5}$$

Unless the initial state $|\psi(0)\rangle$ is accidentally orthogonal to the groundstate $|\psi_0\rangle$, the latter will dominate the sum at large τ, while excited state contributions will be suppressed by factors $\exp(-\frac{\tau}{\hbar}(E_1 - E_0))$.

• The free density matrix $\rho^{\text{free}}(\hat{\varepsilon})$ is simply the operator $\exp(-\frac{\hbar^2\hat{\varepsilon}}{2m}\Delta)$, which describes a free diffusion in imaginary time, and therefore has Gaussian matrix elements as used in eq.(8.1). This equation can then be written (with $\hat{\varepsilon} \to \delta\tau/\hbar$) as

$$\langle x_{M+1}|\exp(-\beta H)|x_1\rangle = \lim_{M\to+\infty} C^{M-1}\int\prod_{j=2}^{M}\mathrm{d}x_j\prod_{j=1}^{M}\exp\left[-\frac{\delta\tau}{\hbar}\left(+\frac{1}{2}m\left(\frac{x_{j+1}-x_j}{\delta\tau}\right)^2 + V(x_j)\right)\right] \tag{8.6}$$

where $C = (2\pi\hbar\delta\tau/m)^{-1/2}$. One can then take the trace by imposing $x_{M+1} = x_1$ and integrating over x_1, to obtain

$$Z = \int_{x(\beta\hbar)=x(0)} \mathcal{D}x(\tau)\ \exp\left[-\frac{1}{\hbar}\int_0^{\beta\hbar} d\tau\left(\frac{1}{2}m(\frac{dx}{d\tau})^2 + V(x)\right)\right] \tag{8.7}$$

up to an infinite, but irrelevant, multiplicative constant $\lim_{M\to+\infty} C^M$ which drops out of all expectation values. Now, the expression "path integral" is clear: Z is an integral over periodic paths $x(\tau)$. Since the integral is over functions of τ, Z is called a functional integral. The exponent $\int d\tau \left(\frac{1}{2}m(\frac{dx}{d\tau})^2 + V(x)\right)$ is called the *Euclidean action* S_E, which depends on the path $x(\tau)$. Each path contributes to Z with a weight $\exp(-\frac{1}{\hbar}S_E)$.

Furthermore, notice the role of $\hbar$ in the exponent of Eq.(8.7): it governs the magnitude of the fluctuations, and therefore plays a role analogous to that of a *temperature*. In the limit $\hbar \to 0$, all fluctuations are suppressed and the only path surviving in the path integral is that which minimizes the exponent: this is the classical limit, where the path from $(x_0, 0)$ to (x, β) is deterministic, and we recover the *minimum action principle*. When $\hbar \neq 0$, fluctuations are allowed around this special path: these fluctuations are *quantum fluctuations*.

Expectation values of observables W, defined by

$$\langle W \rangle = \frac{1}{Z}\mathrm{Tr}\left(W \exp(-\beta H)\right) \tag{8.8}$$

can be measured by forming the average over measurements obtained on paths generated by Monte Carlo sampling of Z. When $\beta \to \infty$, one recovers the groundstate (or "vacuum") expectation value

$$\langle W \rangle \underset{\beta\to\infty}{=} \frac{\langle\psi_0|W|\psi_0\rangle}{\langle\psi_0|\psi_0\rangle}. \tag{8.9}$$

A simple example is the position projector $W = |x_0\rangle\langle x_0| = \delta(x_0)$. Substituting above gives

$$\langle \delta(x_0) \rangle \underset{\beta\to\infty}{=} |\langle\psi_0|x_0\rangle|^2 = |\psi_0(x_0)|^2 \tag{8.10}$$

so that the number of paths going through x_0 (at any Euclidean time) is proportional to the square of the wave-function. An important observable for quantum field theory is the two-point correlator $\delta(x_0, \tau)\delta(x_0, 0)$, which measures the probability for a path to go through the same position x_0 after an Euclidean time τ. Expanding the expectation value over an eigenstate basis of H, one obtains

$$\begin{aligned}
\langle \delta(x_0,\tau)\delta(x_0,0)\rangle &= \frac{\sum_k \langle\psi_k| \exp(-(\beta-\tau)H)\ \delta(x_0)\ \exp(-\tau H)\ \delta(x_0)|\psi_k\rangle}{\sum_k \langle\psi_k| \exp(-\beta H)|\psi_k\rangle} \\
&= \frac{\sum_{k,l} |\langle\psi_k|\delta(x_0)|\psi_l\rangle|^2 \exp(-(\beta-\tau)E_k)\exp(-\tau E_l)}{\sum_k \exp(-\beta E_k)} \\
&\underset{\tau\ll\beta\to\infty}{\approx} |\langle\psi_0|\delta(x_0)|\psi_0\rangle|^2 + |\langle\psi_1|\delta(x_0)|\psi_0\rangle|^2 \exp(-\tau(E_1-E_0)) + \cdots
\end{aligned} \tag{8.11}$$

where a complete set of states $\sum_l |\psi_l\rangle\langle\psi_l|$ was inserted to go from the first to the second line. The final expression shows that the *connected* correlator $\langle\delta(x_0,\tau)\delta(x_0,0)\rangle - \langle\delta(x_0)\rangle^2$ approaches 0 exponentially fast, and that the rate of decay (in Euclidean time τ) gives the *energy gap* $(E_1 - E_0)$ between the groundstate and the first excited state. Contrary

to the groundstate energy E_0, the energy gap is insensitive to an arbitrary shift of the energies, as happens in quantum field theory under renormalization. Therefore, the decay of the connected two-point correlator is the crucial observable with which a physical, experimentally measurable property of the theory can be extracted.

8.3 Path integrals: from classical mechanics to field theory

Consider first the case of a single, classical particle with Hamiltonian $H = \frac{p^2}{2m} + V$. Hamilton's equations describe the time-evolution of this particle:

$$\frac{dq}{dt} = +\frac{\partial H}{\partial p} \quad \longrightarrow \quad \dot{q} = \frac{p}{m} \tag{8.12}$$

$$\frac{dp}{dt} = -\frac{\partial H}{\partial q} \quad \longrightarrow \quad \dot{p} = -\nabla V \tag{8.13}$$

The usual point of view is to start from initial conditions $(q, \dot{q})$ at time $t = 0$, and evolve q and $\dot{q}$ according to the coupled ODEs above. Note, however, that the boundary conditions can instead be split between the beginning and the end of the evolution. In particular, one can specify the beginning and ending coordinates $(q(0), q(t))$. There is a unique path $q(t'), t' \in [0, t]$, which satisfies the above equations, and specifies the initial and final velocities. To find this path, it is convenient to change viewpoint and consider the action $S = \int_0^t dt' \mathcal{L}(q, \dot{q})$, where $\mathcal{L}$ is the Lagrangian $\frac{1}{2}m\dot{q}^2 - V(q)$. One then invokes the principle of stationary (or least) action, from which can be derived the Euler-Lagrange equations

$$\frac{\partial \mathcal{L}}{\partial q_i} = \partial_\mu \frac{\partial \mathcal{L}}{\partial(\partial_\mu q_i)} \quad . \tag{8.14}$$

Note that the notion of action is more general than the notion of Hamiltonian: some systems have an action, but no Hamiltonian. This was in fact the original motivation for Feynman to develop the path integral formalism in his Ph.D. thesis: he was interested in systems having non-local interactions in time (with an interaction term $q(t)q(t')$).

Consider now many particles interacting with each other, with Lagrangian $\mathcal{L} = \sum_i \frac{1}{2}m\dot{q}_i^2 - V(\{q_i\})$ and take the number q_i to represent the *state* of particle i, whose x and y coordinates are fixed on a square grid of spacing a. Furthermore, take the interaction between particles to be of the form $\sum_{\langle ij \rangle}(q_i - q_j)^2$, where $\langle ij \rangle$ stands for nearest-neighbours on the grid, as if springs were connecting i and j. Low-energy configurations will then have almost the same q-value at neighbouring grid points, so that the configuration $\{q_i\}$ will be smooth and look like a "mattress" as in Fig. 8.2.

When the grid spacing a is reduced to zero, the configuration $\{q_i\}$ becomes a *classical field* $q(\vec{x})$ ($\vec{x} \in \mathcal{R}^2$ in this example), with infinitely many degrees of freedom. The action of this field is specified by its Lagrangian density $\mathcal{L} = \frac{1}{2}\partial_\mu q \partial^\mu q - \frac{1}{2}m_0^2 q^2 - V(q)$ where the first term is the continuum version of $(q_i - q_j)^2$ (with $\partial_\mu q \partial^\mu q = \dot{q}^2 - |\vec{\nabla} q|^2$), the second one is a harmonic term corresponding to a mass, and the last term

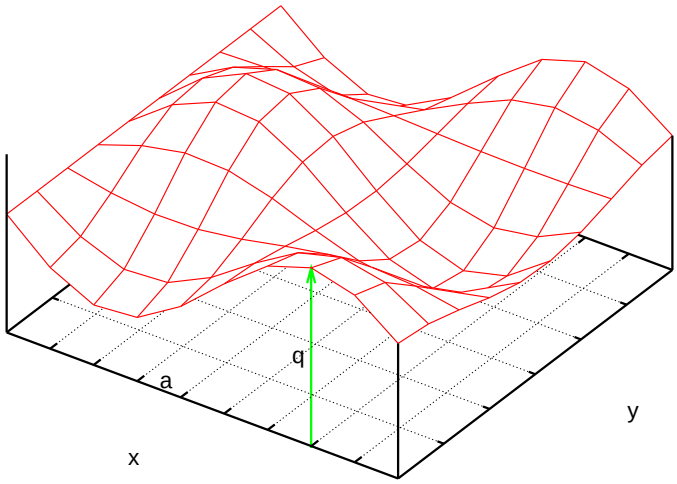


Figure 8.2: A field configuration $q(x, y)$, discretized on a square grid of spacing a.

describes the local (anharmonic) interaction, e.g. $\propto q^4$ [1]. The action is then $S = \int_0^t dt' dx dy \mathcal{L}(q(x, y, t'))$. Note that the Lagrangian density $\mathcal{L}$ satisfies several symmetries: it is invariant under translations and rotations in the (x, y) plane, and under the sign flip $q(\vec{x}) \to -q(\vec{x})\ \forall \vec{x}$, at least for an interaction $\propto q^4$. Each continuous symmetry leads to a conserved quantity: energy-momentum for translations, angular momentum for rotations. We will see the importance of the discrete symmetry $q \leftrightarrow -q$ later.

Now we can consider quantum effects on the above system. As a result of quantum fluctuations, the path from $q(t = 0)$ to $q(t)$ is no longer unique. As in quantum mechanics, all paths contribute with an amplitude $\propto \exp(+\frac{i}{\hbar} \int_0^t dt' \mathcal{L})$, from which it becomes clear that the magnitude of relevant fluctuations in the action is $\hbar$. One can then follow the strategy of Sec. 8.2 and make time purely imaginary, by introducing $\tau = it \in \mathcal{R}$. The immediate result is that idt' above becomes $d\tau'$, so that the amplitude becomes real. The other change is $\dot{q}^2 \to -(\partial_\tau q)^2$, so that an overall minus sign can be taken out, leading to the amplitude

$$\exp(-\frac{1}{\hbar} S_E) \tag{8.15}$$

where $S_E = \int d\tau' d\vec{x} \mathcal{L}_E$ is the Euclidean action, and

$$\mathcal{L}_E = \frac{1}{2}(\partial_\mu \phi)^2 + \frac{1}{2} m_0^2 \phi^2 + V(\phi) \tag{8.16}$$

is the Euclidean Lagrangian density, and the field q is now denoted by ϕ as is customary. The first term $(\partial_\mu \phi)^2 = (\partial_\tau \phi)^2 + |\vec{\nabla} \phi|^2$ is now symmetric between space and time, so that the metric is *Euclidean* in $(d+1)$ dimensions (d spatial dimensions, plus Euclidean time).

It is worth summarizing the sign flips which occurred in the kinetic energy T and the potential energy U during the successive steps we have just taken. We started with

[1] One could think of additional interaction terms, constructed from *higher derivatives* of the field. They are not considered here because they lead to non-renormalizable theories.

the Hamiltonian $H = T + U$, then considered the Lagrangian $L = T - U$. Going to imaginary time changes the sign of T. Finally, we take out an overall minus sign in the definition of L_E, so that paths with the smallest action are the most likely. This leads to the Euclidean Lagrangian density $L_E = T + U$, which is identical to the Hamiltonian we started from, except that the momentum p is replaced by the derivative $\partial_0 \phi$.

It is also useful to perform some elementary dimensional analysis. Since it appears in the exponent of the amplitude Eq.(8.15), the Euclidean action S_E is *dimensionless* (we set $\hbar = 1$). Hence the Euclidean Lagrangian density has mass dimension $(d+1)$, and therefore the field ϕ has mass dimension $\frac{d-1}{2}$. This is interesting, because if we take the "normal" number of spatial dimensions $d = 3$ and the interaction term $V(\phi) = \frac{g_0}{4!}\phi^4$, then g_0 is a dimensionless number. It makes sense then to perform a Taylor expansion of this theory in powers of g_0 about the free case $g_0 = 0$: this is the scope of perturbation theory. Here, we will try to obtain non-perturbative results, by directly simulating the theory at some large value of g_0.

We have so far considered a field $\phi(\vec{x}, \tau)$ which takes values in $\mathcal{R}$. It is easy to generalize the Lagrangian density Eq.(8.16) to cases when ϕ takes values in $\mathcal{C}$, or has several components forming a vector $\vec{\phi} \equiv \begin{pmatrix} \phi_1 \\ .. \\ \phi_N \end{pmatrix}$, perhaps with a constraint $\sum_N \phi_k^2 = 1$, depending on the desired symmetries. Typically, the Euclidean Lagrangian density is the starting, defining point of a quantum field theory.

Finally, we can introduce a finite temperature T, exactly as we did in the quantum-mechanical case (and setting $\hbar = 1$): we make the Euclidean time direction *compact*: $\tau \in [0, \beta = \frac{1}{T}]$, and impose periodic boundary conditions on the field ϕ: $\phi(\vec{x}, \beta) = \phi(\vec{x}, 0)\ \forall \vec{x}$. This works for the same reason as in quantum mechanics: the partition function

$$Z = \int_{periodic} \mathcal{D}\phi \exp(-\int_0^\beta d\tau' d\vec{x} \mathcal{L}_E(\phi)) \tag{8.17}$$

is a weighted sum of eigenstates of the Hamiltonian: $Z = \sum_i \exp(-\beta E_i)$. We will be concerned here with the $T = 0$ situation. In that case, the two-point correlator provides a means to measure the mass gap $(E_1 - E_0)$:

$$\langle \phi(\vec{x}, \tau)\phi(\vec{x}, 0)\rangle - \langle\phi\rangle^2 \underset{\tau\to\infty}{=} c^2 \exp(-(E_1 - E_0)\tau) \tag{8.18}$$

or equivalently the correlation length $\xi = (E_1 - E_0)^{-1}$. The lowest energy state, with energy E_0, is the vacuum, which contains particle-antiparticle pairs because of quantum fluctuations, but whose net particle number is zero. The first excited state, with energy E_1, contains one particle at rest. Call its mass $m_R = E_1 - E_0$. Then this mass can be obtained from the decay of the two-point correlator, as $m_R = 1/\xi$. This is the "true", measurable mass of the theory, and it is not equal to the mass m_0 used in the Lagrangian density. m_R is called the *renormalized* mass, while m_0 is the *bare* mass. Similarly, the "true" strength g_R of the interaction can be measured from 4-correlators of ϕ, and it is not equal to the coupling g_0 used in the Lagrangian density: g_0 is the bare coupling, g_R the renormalized coupling.

8.4 Numerical study of ϕ^4 theory

Here, we show that very important results in Quantum Field Theory can be extracted from simulations of the $4d$ Ising model. Our starting point is the continuum Euclidean action:

$$S_E = \int d\tau d^3x \left[\frac{1}{2}(\partial_\mu \phi_0)^2 + \frac{1}{2} m_0^2 \phi_0^2 + \frac{g_0}{4!} \phi_0^4 \right] \tag{8.19}$$

where the subscript 0 is to emphasize that we are dealing with bare quantities (field, mass and coupling), and the coupling normalization $1/4!$ is conventional. We discretize the theory on a hypercubic ($4d$) lattice with spacing a [2]. After the usual replacements $\int d\tau d^3x \ \rightarrow \ a^4 \sum_{\text{sites } x}$ and $\partial_\mu \phi_0 \ \rightarrow \ \frac{\phi_0(x+\hat{\mu}) - \phi_0(x)}{a}$, we end up with the lattice action

$$S_L = \sum_x \left[-2\kappa \sum_\mu \phi(x)\phi(x+\hat{\mu}) + \phi(x)^2 + \lambda(\phi(x)^2 - 1)^2 - \lambda \right] \tag{8.20}$$

where we use the new variables ϕ, κ and λ defined by

$$a\phi_0 = \sqrt{2\kappa}\ \phi \tag{8.21}$$

$$a^2 m_0^2 = \frac{1-2\lambda}{\kappa} - 8 \tag{8.22}$$

$$g_0 = \frac{6\lambda}{\kappa^2} \tag{8.23}$$

Note in particular the multiplication of ϕ_0 by a to form a dimensionless variable, since ϕ_0 has mass dimension 1. The original formulation had two bare parameters, m_0 and g_0. They have been mapped into two bare parameters, κ and λ. This discretized theory can be simulated by standard Monte Carlo algorithms like Metropolis, on a hypercubic lattice of L sites in each direction. To minimize finite-size effects, periodic boundary conditions are usually imposed in each direction.

The behaviour of our system is easy to understand qualitatively in the two limits $\lambda = 0$ and $\lambda = +\infty$.

• When $\lambda = 0$, the interaction is turned off. This is the free theory, which has two phases depending on the value of κ: a disordered or symmetric phase $\langle\phi\rangle = 0$ when κ is small, and an ordered phase $\langle\phi\rangle \neq 0$ when κ is large. Thus, the symmetry $\phi \leftrightarrow -\phi$ is spontaneously broken when $\kappa > \kappa_c = \frac{1}{8}$, which corresponds to the vanishing of the mass m_0.

• When $\lambda = +\infty$, fluctuations of ϕ away from the values ± 1 cost an infinite amount of energy. Thus, ϕ is restricted to ± 1, and our theory reduces to the $4d$ Ising model with coupling 2κ. As in lower dimensions, the Ising model undergoes a *second-order phase transition* corresponding to the spontaneous breaking of the symmetry $\phi \leftrightarrow -\phi$, for a critical value $\kappa_c \approx 0.075$.

For intermediate values of λ, again a second-order transition takes place, leading to the phase diagram depicted Fig. 8.3.

[2] The lattice spacing is taken to be the same in space and in time for simplicity; one could consider different values a_s and a_τ.

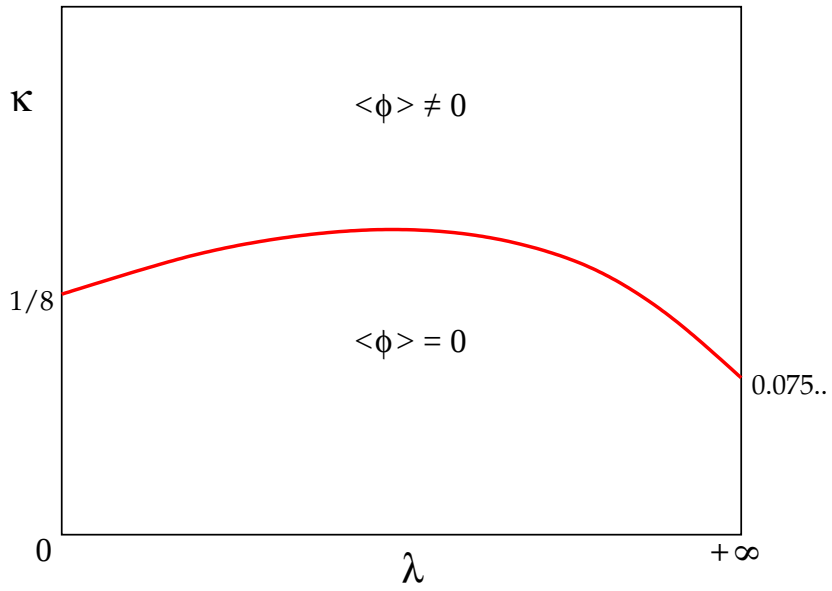


Figure 8.3: Phase diagram of the lattice theory defined by Eq.(8.20). The two phases are separated by a line of second-order phase transitions.

The existence of a second-order phase transition is crucial: it allows us to define a *continuum limit* of our lattice theory. Remember that the "true", renormalized mass m_R can be extracted from the exponential decay of the 2-point correlator

$$\langle\phi(x)\phi(y)\rangle - \langle\phi\rangle^2 \underset{|x-y|\to\infty}{\propto} \exp(-m_R|x-y|) = \exp(-\frac{|x-y|}{\xi}) \tag{8.24}$$

(see Eq.(8.18)). On the lattice, we can only measure the dimensionless combination $am_R = \frac{1}{\xi/a}$, and the separation $|x-y|$ can only be measured in lattice units, i.e. $\frac{|x-y|}{a}$. Taking the continuum limit $a \to 0$ (while the physical mass m_R is fixed) forces the correlation length *measured in lattice units*, i.e. ξ/a, to diverge. This only occurs when the lattice theory has a second-order phase transition (or higher order).

Therefore, the interpretation of a second-order phase transition is different between solid-state physics and lattice field theory. In the first case, the lattice spacing has a physical meaning, like the distance between two ions in a crystal: the lattice is "for real", and the correlation length really diverges at a second-order critical point. In the lattice field theory, the correlation length is "for real", and the lattice spacing a shrinks to zero at the critical point. This is illustrated in Fig. 8.4.

In this latter case, one must be careful that the physical box size (La) also shrinks as $a \to 0$. In order to obtain a controlled continuum limit at constant physical volume, one must increase the number of lattice points L in each direction keeping (La) fixed.

Going back to our ϕ^4 theory, one sees that a continuum limit can be defined for any value of the bare coupling $\lambda \in [0, +\infty]$, by tuning κ to its critical value $\kappa_c(\lambda)$. An interesting question is: what is the value of the "true", renormalized coupling as a function of λ ? The answer is clear when $\lambda = 0$: the theory is free, and the coupling is zero, whether bare or renormalized. To obtain a non-zero answer, a reasonable

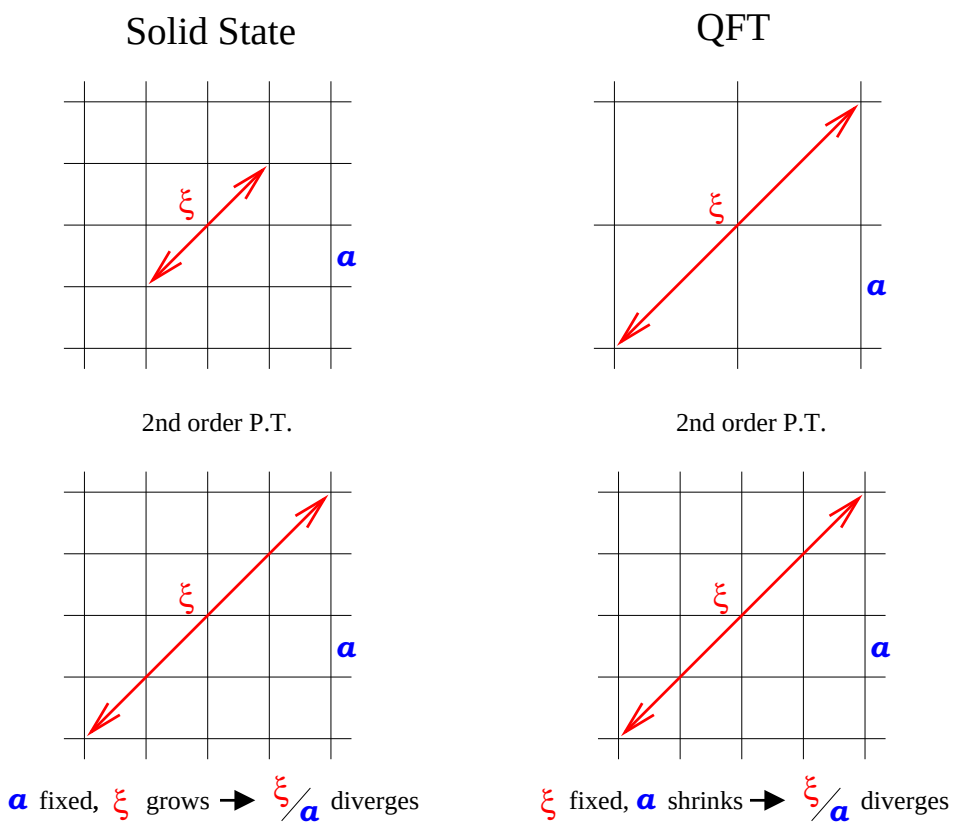


Figure 8.4: Two different viewpoints on a second-order phase transition: in solid-state physics (*left*), the crystal is "real" and the physical correlation length diverges; in quantum field theory (*right*), the correlation length is "real", and the lattice spacing shrinks to zero.

strategy is to maximize the bare coupling, and thus to consider the Ising limit $\lambda = +\infty$. The renormalized coupling is extracted from the strength of the 4-spin correlation, normalized as explained in the Exercises. The rather surprising answer is that the renormalized coupling is zero, just like for $\lambda = 0$. In fact, the renormalized coupling is always zero for any choice of λ. In other words, the renormalized ϕ^4 theory is *free*, no matter the value of the bare coupling! The formal statement is that the ϕ^4 theory is "trivial". Note that this is only true in $(3+1)$ dimensions. In lower dimensions, the renormalized coupling is non-zero.

Now, why is this finding important? The Standard Model of particle interactions contains a Higgs particle, which gives a mass to all other particles by coupling to them. The field theory describing the Higgs particle is very much like the ϕ^4 theory we have just studied, except that the field ϕ is now a complex doublet $\begin{pmatrix} \phi_1 + i\phi_2 \\ \phi_3 + i\phi_4 \end{pmatrix}$. The bare parameters are chosen so that the system is in the broken-symmetry phase, where ϕ has a non-zero expectation value. The masses of all particles are proportional to $\langle\phi\rangle$, therefore it is crucial that $\langle\phi\rangle \neq 0$. In turn, this symmetry breaking is only possible if the coupling λ is non-zero. But, as we have seen, the "true", renormalized value of λ is

zero. Therefore, we have a logical inconsistency. The consequence is that the Standard Model cannot be the final, exact description of particle interactions. Some new physics beyond the Standard Model must exist, and from the numerical study of the lattice theory near $\kappa_c(\lambda)$, one can set the energy scale for this new physics to become visible at around 600-800 GeV. This argument is so powerful that it has been used in the design of the Large Hadron Collider (LHC), recently turned on at CERN, and which will study collisions up to about 1000 GeV only.

8.5 Monte Carlo algorithms for the Ising model

It is most efficient to perform numerical simulations of the ϕ^4 theory in its Ising limit. The partition function to sample is then simply

$$Z = \sum_{\sigma_x = \pm 1} \exp(+J \sum_{\langle xy \rangle} \sigma_x \sigma_y) \tag{8.25}$$

with a homogeneous ferromagnetic interaction $J = 2\kappa > 0$ between pairs of spins σ_x, σ_y at nearest-neighbour sites $\langle xy \rangle$. This partition function can be sampled by the general Metropolis algorithm. However, the autocorrelation (Monte Carlo) time necessary to produce statistically independent configurations grows with the correlation length ξ as ξ^z, where the dynamic critical exponent z is about 2. Simulations near the second-order phase transition, where ξ diverges, become very demanding in computer resources. This phenomenon is called critical slowing down.

Two alternatives to the Metropolis algorithm have been discovered, the cluster algorithm (Swendsen & Wang, 1987) and the worm algorithm (Prokof'ev & Svistunov, 2001), which give $z \sim 0$ to 0.5 (z varies with the dimensionality of the system) for the Ising model. They have revolutionized the numerical study of the systems where they work efficiently. To obtain interesting results on the renormalized coupling of the ϕ^4 theory, either of these algorithms should be used.

8.6 Gauge theories

Of the four forces known in Nature, at least three (the strong, the weak and the electromagnetic forces) are described by gauge theories. In addition to the usual "matter" fields (electrons, quarks), these theories contain "gauge" fields (photons, W and Z bosons, gluons) which "mediate" the interaction: the interaction between, say, two electrons is caused by the exchange of photons between them. This is analogous to the exchange of momentum which occurs when one person throws a ball at another, and the other catches it. In this way, two particles interact when they are far apart, even though the Lagrangian contains only local interactions. Moreover, gauge theories are invariant under a larger class of symmetries, namely *local* (x-dependent) symmetries.

8.6.1 QED

As an example, let us consider here a variant of Quantum ElectroDynamics (QED), called scalar QED, where electrons are turned into bosons. A simple modification of

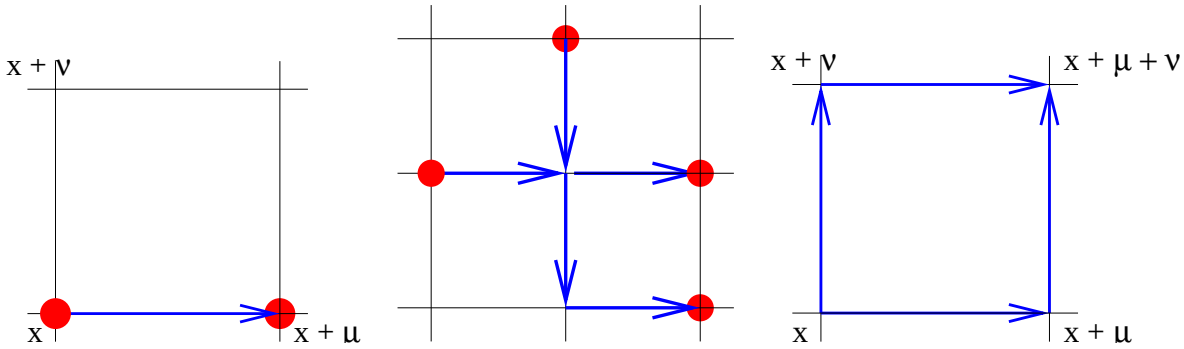


Figure 8.5: Graphical representation (*left*) of the gauge-invariant nearest-neighbour interaction: $\phi^*(x)\phi(x+\hat{\mu})$ becomes $\phi^*(x)U_\mu(x)\phi(x+\hat{\mu})$; (*middle*) an example of a gauge-invariant 4-point correlation; (*right*) the smallest closed loop is the plaquette, with associated matrix $U_\mu(x)U_\nu(x+\hat{\mu})U_\mu^\dagger(x+\hat{\nu})U_\nu^\dagger(x)$.

the previous ϕ^4 theory is required: in order to represent charged bosons, the field ϕ, instead of being real, is made complex, $\phi(x) \in \mathcal{C}$. The continuum Euclidean action becomes

$$S_E = \int d\tau d^3x \left[|\partial_\mu \phi_0|^2 + m_0^2|\phi_0|^2 + \frac{g_0}{4!}|\phi_0|^4\right] \tag{8.26}$$

and, after discretization on the lattice:

$$S_L = \sum_x \left[-\kappa \sum_\mu (\phi^*(x)\phi(x+\hat{\mu}) + \text{h.c.}) + |\phi(x)|^2 + \lambda(|\phi(x)|^2 - 1)^2 - \lambda\right] \tag{8.27}$$

S_L is invariant under the *global* (x-independent) rotation $\phi(x) \to \exp(i\alpha)\phi(x)\ \forall x$. The idea is now to promote this symmetry to a *local* one, where α may depend on x. It is clear that the derivative term $\phi^*(x)\phi(x+\hat{\mu})$ is not invariant under this transformation. Invariance is achieved by introducing new degrees of freedom, namely complex phases (elements of $U(1)$) which live on the links between nearest-neighbours. Call $U_\mu(x) = \exp(i\theta_\mu(x))$ the link variable starting at site x in direction μ, and give it the initial value 1, i.e. $\theta_\mu(x) = 0\ \forall x, \mu$. Modify the derivative term as follows:

$$\phi^*(x)\phi(x+\hat{\mu}) \ \to \ \phi^*(x)U_\mu(x)\phi(x+\hat{\mu}) \tag{8.28}$$

This term is now invariant under a *local* transformation $\phi(x) \to \exp(i\alpha(x))\phi(x)$, with $\alpha(x) \neq \alpha(x+\hat{\mu})$, provided that $U_\mu(x)$ also transforms:

$$\phi(x) \ \to \ \exp(i\alpha x)\phi(x) \tag{8.29}$$

$$U_\mu(x) \ \to \ \exp(i\alpha(x))U_\mu(x)\exp(-i\alpha(x+\hat{\mu})) \tag{8.30}$$

The significance of the new variables $U_\mu(x)$ and of the new expression for the discretized derivative can be elucidated by expressing $\theta_\mu(x) = eaA_\mu(x)$, and considering the continuum limit $a \to 0$. To lowest order in a, the derivative $\partial\mu$ becomes the *covariant derivative* $D_\mu \equiv \partial_\mu + ieA_\mu$, and the transformation eq.(8.30) is a *gauge transformation*

for A_μ: $A_\mu(x) \rightarrow A_\mu(x) - e\partial_\mu\alpha(x)$. Thus, our link variables $U_\mu(x)$ represent the gauge potential $A_\mu(x)$ associated with the electromagnetic field, and eq.(8.28) describes the interaction of our bosonic electrons with the photon. To complete the lattice discretization of QED, what is missing is the energy of the electromagnetic field, namely $\frac{1}{2}\int d\vec{x}d\tau(|\vec{E}|^2(x) + |\vec{B}|^2(x))$. We identify its lattice version below.

It becomes simple to construct n-point correlators of ϕ which are invariant under the local transformation eq.(8.30): all the fields ϕ need to be connected by "strings" of gauge fields, made of products of gauge links U_μ as in Fig. 8.5. Under a local gauge transformation, the phase changes $\alpha(x)$ will always cancel out between ϕ and the attached U, or between two successive U's.

There exists another type of gauge-invariant object. Consider the product of links $\prod_{x\to x} U$ around a closed loop, starting and ending at x. It transforms as

$$\prod_{x\to x} U \;\rightarrow\; \exp(i\alpha(x)) \left(\prod_{x\to x} U\right) \exp(-i\alpha(x)) \tag{8.31}$$

which is invariant since all the U's are complex phases which commute with each other. Thus, another valid term to add to the [real] lattice action is the real part of any closed loop, summed over translations and rotations to preserve the other desired symmetries. The simplest version of such a term is to take elementary square loops of size a, made of 4 links going around a *plaquette*: $P_{\mu\nu}(x) \equiv U_\mu(x)U_\nu(x+\hat{\mu})U_\mu^\dagger(x+\hat{\nu})U_\nu^\dagger(x)$. Thus, the complete action of our scalar QED theory is

$$\sum_x |\phi(x)|^2 - \kappa \sum_x \sum_\mu (\phi^*(x)U_\mu(x)\phi(x+\hat{\mu}) + \text{h.c.}) + \beta \sum_x \sum_{\mu\neq\nu} \text{Re}(P_{\mu\nu}(x)) \tag{8.32}$$

The plaquette term looks geometrically like a curl. Indeed, substituting $U_\mu(x) = \exp(ieaA_\mu(x))$ and expanding to leading-order in a yields

$$\text{Re}(P_{\mu\nu}(x)) \approx 1 - \frac{1}{2}e^2a^4(\partial_\mu A_\nu - \partial_\nu A_\mu)^2 \tag{8.33}$$

so that the last term in eq.(8.32) becomes, up to an irrelevant constant, $-\beta e^2 \frac{1}{2}\int d\vec{x}d\tau(|\vec{E}|^2 + |\vec{B}|^2)$, where one has expressed the electric and magnetic fields $\vec{E}$ and $\vec{B}$ in terms of the gauge potential A_μ. It suffices then to set $\beta = 1/e^2$ to recover the energy of an electro-magnetic field.

Note that it is our demand to preserve invariance under the local transformation eq.(8.30) which has led us to the general form of the action eq.(8.32). We could have considered larger loops instead of plaquettes. But in the continuum limit $a \to 0$, these loops would yield the same continuum action. So the form of the QED action is essentially dictated by the local gauge symmetry.

One can now study the scalar QED theory defined by eq.(8.32) using Monte Carlo simulations, for any value of the bare couplings (κ, β). Contrary to continuum perturbation theory, one is not limited to small values of e (i.e. large β).

8.6.2 QCD

Other gauge theories have basically the *same* discretized action eq.(8.32). What changes is the group to which the link variables $U_\mu(x)$ belong. For QCD, these variables represent

the gluon field, which mediates the interaction between quarks. Quarks come in 3 different colors and are thus represented by a 3-component vector at each site[3]. Hence, the link variables are 3×3 matrices. The gauge-invariant piece associated with a closed loop is the *trace* of the corresponding matrix, thanks to cyclic invariance of the trace in eq.(8.31). No other changes are required to turn lattice QED into lattice QCD!

As emphasized in Sec. 8.4, the Euclidean Lagrangian density defines the lattice theory. The continuum limit can be obtained by approaching a critical point. For QCD, the critical point is $\beta \to +\infty$, i.e. $g_0 = 0$ since $\beta \propto 1/g_0^2$ as in QED. As we have seen, the vanishing of the bare coupling does not imply much about the true, renormalized coupling.

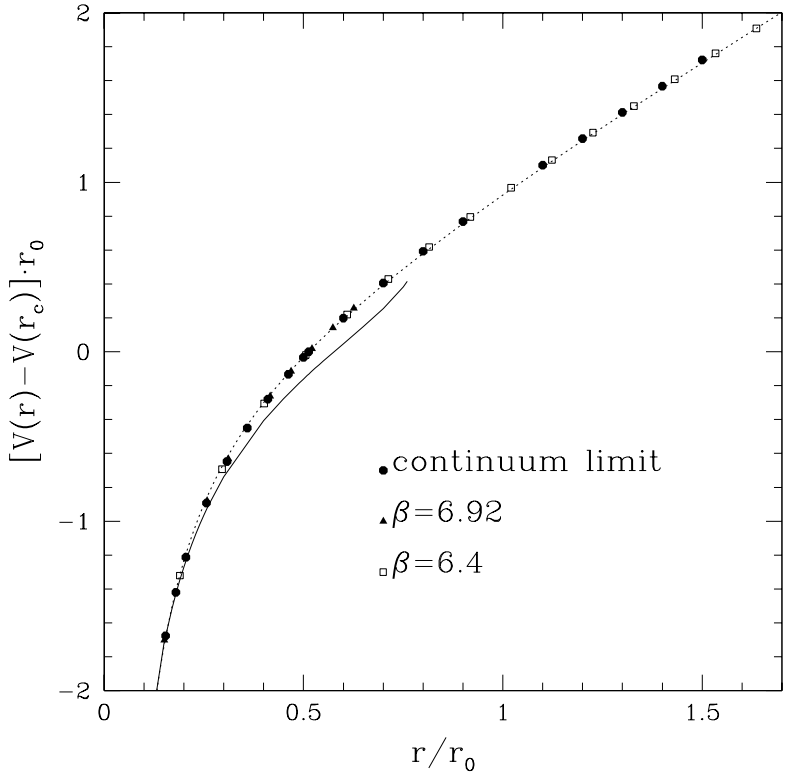


Figure 8.6: Potential $V(r)$ between a static quark and antiquark, as a function of their separation r. Data obtained at 2 values of the lattice spacing (finite values of β) are extrapolated to the continuum limit ($\beta \to \infty$). At short distance, the potential is Coulomb-like because the interaction becomes weak (the solid line shows the prediction of perturbation theory). At large distance, the potential rises linearly, which shows that it takes an infinite energy to separate the two objects: quarks are confined. A simple model of a vibrating string (dotted line) gives a good description, down to remarkably short distances.

[3]This would be the full story if quarks were bosons. Because they are fermions, each color component is in fact a 4-component vector, called Dirac *spinor*.

8.7 Overview

The formulation of lattice QCD is due to K. Wilson (1974). First Monte Carlo simulations were performed by M. Creutz in 1980, on a 4^4 lattice. A goal of early simulations was to check whether quarks were confined. This can be demonstrated by considering the potential $V(r)$ between a quark and an anti-quark separated by distance r. Contrary to the case of QED where the potential $\propto 1/r$ saturates as $r \to \infty$, in QCD the potential keeps rising linearly, $V(r) \sim \sigma r$, so that it takes an infinite amount of energy to completely separate the quark and anti-quark. Equivalently, the force between the two objects goes to a *constant* σ. The energy of the quark-antiquark pair grows as if it was all concentrated in a tube of finite diameter. In fact, describing the quark-antiquark pair as an infinitely thin vibrating string is a very good approximation, as shown in the state-of-the-art Monte Carlo data Fig. 8.6, now performed on 64^4 lattices. To control discretization errors, the lattice spacing must be about $1/10^{\text{th}}$ of the correlation length or less. To control finite-volume effects, the lattice size must be about 3 times the correlation length or more. This implies lattices of minimum size 30^4, which have only become within reach of a reasonable computer budget in recent years.

The above simulations considered only the effect of gluons: since gluons carry a color charge (in contrast to the photon which is electrically neutral), they can lead to complex effects like the confinement of charges introduced in the gluon system. But to study QCD proper, quarks must be simulated also. This is more difficult because quarks are fermions, i.e. non-commuting variables. The strategy is to integrate them out analytically. This integration induces a more complicated interaction among the remaining gluonic link variables. Actually, this interaction is *non-local*, which increases the algorithmic complexity of the Monte Carlo simulation. An efficient, exact simulation algorithm, called Hybrid Monte Carlo, was only discovered in 1987 (see bibliography). Even so, the simulation of so-called "full QCD", on lattices of size 30^4 or larger, requires a computer effort $\mathcal{O}(1)$ Teraflop$\times$ year, which has forced the community to evolve into large collaborations using dedicated computers.

Using these resources, one is now able to reproduce the masses of quark and anti-quark bound states, i.e. mesons and baryons, to a few percent accuracy. The impact of neglecting the effect of quarks or including them is nicely illustrated in Fig. 8.7. Some predictions have also been made for the properties of mesons made of charm or bottom quarks, currently being studied in particle accelerators.

Another essential purpose of QCD simulations is to quantify QCD effects in experimental tests of the electroweak Standard Model. By checking whether experimental results are consistent with the Standard Model, one can expose inconsistencies which would be the signature of new, beyond-the-standard-model physics. To reveal such inconsistencies, one must first determine precisely the predictions of the Standard Model. This entails the determination of QCD effects, which can only be obtained from lattice QCD simulations.

Finally, another direction where QCD simulations have been playing a major role is that of high temperature. The confinement of quarks, which is an experimental fact at normal temperatures, is believed to disappear at very high temperatures $\mathcal{O}(100)$ MeV $\sim \mathcal{O}(10^{12})$ K. This new state of matter, where quarks and gluons form a plasma, is being probed by accelerator experiments which smash heavy ions against each other.

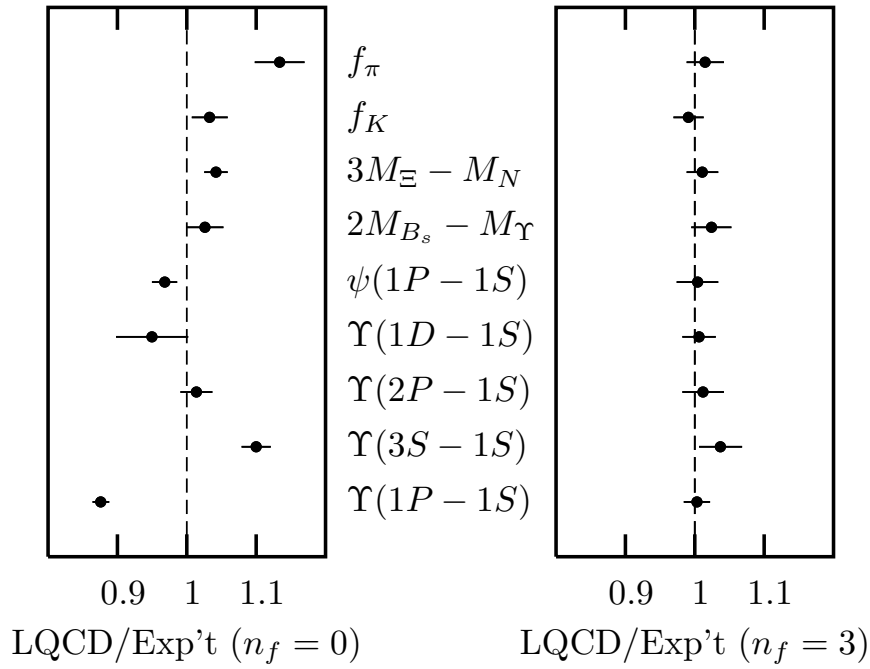


Figure 8.7: Comparison of lattice and experimental measurements of various quantities. The left figure shows the ratios lattice/experiment, for a lattice model which neglects quark effects (the number N_f of quark flavors is set to zero). The right figure shows the same ratios, for a lattice model including 3 quark flavors, all with equal masses.

Lattice simulations provide an essential tool to predict the properties of this plasma.

8.8 Useful references

- *Computational Physics*, by J.M. Thijssen, Cambridge Univ. Press (2007), second edition. See Chapter 15.
- *Quantum Field Theory in a nutshell*, by A. Zee, Princeton Univ. Press (2003). This book reads like a thriller.
- *Hybrid Monte Carlo*, by S. Duane, A. D. Kennedy, B. J. Pendleton and D. Roweth, Phys. Lett. B **195** (1987) 216.

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

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

or in general to a multi-dimensional equation

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

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

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

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

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

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:

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

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

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

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

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

to obtain the **higher dimensional Newton method**

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

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

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

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

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

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

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

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

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

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 minimuim 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

$$u_{n+1} = \frac{Au_n}{||Au_n||}, \tag{A.13}$$

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

Lanczos iterations

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

$$\beta_{n+1} v_{n+1} = A v_n - \alpha_n v_n - \beta_n v_{n-1}, \tag{A.14}$$

where

$$\alpha_n = v_n^\dagger A v_n, \qquad \beta_n = |v_n^\dagger A v_{n-1}|. \tag{A.15}$$

As the orthogonality condition

$$v_i^\dagger v_j = \delta_{ij} \tag{A.16}$$

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

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

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 $\{v_1, v_2, \ldots, v_M\}$. 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 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:

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

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 fromhttp://www.comp-phys.org/software/ietl/ .