Computational Quantum Physics

Giuseppe Carleo[1]

[1] Institute of Physics, Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015, Lausanne, Switzerland

Contents

1 Introduction **3**
1.1 General 3
1.1.1 Exercises 3
1.1.1.1 Programming Languages 4
1.1.1.2 Computer Access 4
1.1.2 Prerequisites 4
1.1.2.1 Computing 4
1.1.2.2 Numerical Analysis 4
1.1.2.3 Quantum Mechanics 5
1.1.3 References 5
1.2 Overview 5

2 Quantum mechanics in one hour **7**
2.1 Introduction 7
2.2 Basis of quantum mechanics 7
2.2.1 Wave functions and Hilbert spaces 7
2.2.2 Mixed states and density matrices 8
2.2.3 Observables 9
2.2.4 The measurement process 9
2.2.5 The uncertainty relation 10
2.2.6 The Schrödinger equation 11
2.2.6.1 The time-dependent Schrödinger equation 11
2.2.6.2 The time-independent Schrödinger equation 11
2.2.6.3 The Schrödinger equation for the density matrix . . . 11
2.2.7 The thermal density matrix 11
2.3 The spin-S problem 12
2.4 A quantum particle in free space 13
2.4.1 The harmonic oscillator 14

3 The quantum one-body problem **17**
3.1 The time-independent 1D Schrödinger equation 17
3.1.1 Discretizing the Hamiltonian 17
3.2 The time-independent Schrödinger equation in higher dimensions . . . 18
3.2.1 Factorization along coordinate axis 19
3.2.2 Potential with spherical symmetry 19
3.2.3 Finite difference methods in higher dimension 19

3.3 The time-dependent Schrödinger equation . . . 20
3.3.1 Spectral methods . . . 20
3.3.2 Direct numerical integration . . . 21
3.3.2.1 Practical implementations of the implicit scheme . . . 22
3.4 Appendix: The split operator method . . . 22

4 Exact diagonalization of many-body problems **25**
4.1 Quantum spin models . . . 25
4.1.1 Hamiltonian Matrix . . . 25
4.1.2 Example: the transverse-field Ising model . . . 26
4.2 Finding Ground States . . . 27
4.2.1 Power Method . . . 27
4.2.2 The Lanczos Method . . . 28
4.2.3 Implementation . . . 29
4.3 Quantum Dynamics . . . 30
4.3.1 Taylor Expansion . . . 30
4.4 The Trotter-Suzuki decomposition . . . 31
4.4.1 Time evolution for the transverse field Ising model . . . 32
4.5 Appendix: The Lanczos algorithm . . . 33
4.5.1 Finding eigenvectors . . . 33
4.5.2 Roundoff errors and ghosts in the Lanczos algorithm . . . 34
4.5.3 Open-source implementations . . . 34

5 Indistinguishable particles: fermions and bosons **37**
5.1 Introduction . . . 37
5.2 The Fock space . . . 38
5.2.1 Fermions . . . 39
5.2.2 Spinful fermions . . . 39
5.2.3 Bosons . . . 40
5.3 Creation and annihilation operators . . . 40
5.3.1 Fermionic operators . . . 40
5.3.1.1 Normal ordering . . . 42
5.3.2 Bosonic operators . . . 42
5.4 Exact diagonalization . . . 44
5.4.1 Bosons . . . 44
5.4.2 Fermions . . . 44
5.4.3 Fixing the number of particles . . . 45
5.5 Example Fermionic models . . . 46
5.5.1 The Hubbard model . . . 46
5.5.2 The t-J model . . . 46

6 Electronic structure of molecules and atoms **47**
6.1 Introduction . . . 47
6.1.1 The electronic structure problem . . . 47
6.2 Hamiltonian in second quantization . . . 48
6.3 Basis functions . . . 48

6.3.1 The electron gas 49
6.3.2 Atoms and molecules 49
6.3.3 Electrons in solids 50
6.3.3.1 Pseudo-potentials 50
6.3.4 Other basis sets 50
6.4 The Hartree Fock method 51
6.4.1 The Hartree-Fock approximation 51
6.4.2 Spinless case 51
6.4.2.1 Expectation value of the energy 51
6.4.3 Spin: Restricted Hartree Fock 53
6.4.4 The Roothaan-Hall Equations 54
6.4.5 Configuration-Interaction 55
6.5 Further Reading 56

7 Density functional theory 57
7.1 The electron density 58
7.2 Variational principle for the density 58
7.3 The Kohn-Sham scheme 59
7.4 Local Density Approximation 61
7.5 Improved approximations 62
7.6 Program packages 62
7.7 Further Reading 62

8 Variational Monte Carlo and Stochastic Optimization 63
8.1 Variational Monte Carlo 63
8.1.1 Stochastic Estimates of Properties 64
8.1.1.1 Energy 65
8.2 Stochastic Variational Optimization 65
8.2.1 Zero-Variance Property 66
8.2.2 Stochastic Gradient Descent 67
8.3 Sampling Methods 68
8.3.1 Markov Chain and Detailed Balance 68
8.3.2 The Metropolis-Hastings Algorithm 69
8.3.3 Estimating Errors and Auto-Correlation Times 70
8.4 Examples of spin wave functions 71
8.4.1 Mean-Field Ansatz 71
8.4.2 Jastrow Ansatz 72

9 Machine Learning Methods for Many-Body Quantum Systems 75
9.1 Artificial Neural networks: the machine 75
9.2 Supervised Learning 76
9.3 Neural-Network quantum states 77
9.3.1 The loss function 77
9.4 Taking Gradients 78
9.4.1 The BackPropagation algorithm 78
9.4.2 Computing gradients of the energy 80

10 Time-Dependent Variational Principle **83**
10.1 Time-Dependent Variational States 83
10.2 Imaginary-time evolution 85
10.2.1 Variational Imaginary Time evolution 85
10.3 The Dirac-Frenkel variational principle 86
10.4 Time-Dependent Variational Monte Carlo 87
10.4.1 Overall Algorithm 87
10.5 Example: dynamics of a mean-field variational state 88

11 Ground-State Quantum Monte Carlo **93**
11.1 Imaginary-Time Evolution 93
11.1.1 Path-Integral Representation 93
11.2 Particles in continuous space 95
11.2.1 The Propagator 95
11.3 Path-Integral-Ground-State Monte Carlo 97
11.3.1 Statistical estimators of quantum expectation values 97
11.3.1.1 Sign Problem 98
11.4 Reptation Quantum Monte Carlo 99
11.4.1 Monte Carlo moves 99
11.4.1.1 Move Right 99
11.4.1.2 Move Left 100
11.4.2 Practical aspects for the implementation 100
11.4.2.1 Data Structure 100
11.4.2.2 Putting everything together 101
11.5 Importance Sampled Propagator in continuous space 102
11.6 Diffusion Quantum Monte Carlo 102

12 Finite-Temperature Quantum Monte Carlo **105**
12.1 Thermal Density Matrix 105
12.1.1 Computing Properties 106
12.2 Continuous-Space particles 107
12.2.1 Bose symmetry 108
12.3 Path-Integral Monte Carlo 109
12.3.1 Moving a single bead 109
12.3.2 Moving multiple beads 110
12.3.3 Displacing entire polymers 112
12.3.4 Sampling permutations 112
12.3.5 Energy expectation value 112

13 Path-Integral Monte Carlo for Lattice Models **117**
13.1 Transverse-Field Ising model 117
13.1.1 Short-Time propagator 117
13.1.2 Path-Integral expression 118
13.1.3 Classical 2D Ising Model 118
13.1.4 Monte Carlo Sampling 120
13.1.5 Energy Estimator 120

13.2 Continuous-Time path integrals 121
13.2.1 Example: single spin Hamiltonian 122
13.2.1.1 Inserting and erasing pair of vertexes 123
13.3 Stochastic Series Expansion 124

14 Quantum computing **127**
14.1 Quantum bits and quantum gates 127
14.1.1 Quantum bits 127
14.1.2 Quantum gates 128
14.1.2.1 Single qubit gates 128
14.1.2.2 Two-qubit gates 129
14.1.2.3 Universal gate sets 130
14.1.3 Measurements 130
14.1.4 Example: Creating entangled quantum states 131
14.2 Simulating the dynamics of quantum systems 132
14.2.1 Time evolution of a quantum spin model 132
14.3 Quantum Phase Estimation 133
14.3.1 Measuring energies from phases 133
14.3.2 Quantum phase estimation algorithm 133
14.3.3 Quantum phase estimation to find the exact eigenvalues 135

15 Variational Quantum Algorithms **139**
15.1 Variational State Preparation 139
15.2 Loss Functions 140
15.2.1 Ground-State Preparation 140
15.2.2 Excited States 141
15.3 Parameterized Quantum Circuits 142
15.3.1 Ground states through adiabatic state preparation 142
15.3.2 Quantum alternating operator 143
15.3.3 Hardware Efficient Ansatz 145
15.4 Optimization Algorithms 145
15.4.1 Parameter Shift Rule 146
15.5 Qubits Encodings 147
15.5.1 Fermions 147

Chapter 1

Introduction

> *The fundamental laws necessary for the mathematical treatment of a large part of physics and the whole of chemistry are thus completely known, and the difficulty lies only in the fact that application of these laws leads to equations that are too complex to be solved.*— Paul Dirac (1929)

1.1 General

As pointed out already by Dirac in the quote above, the fundamental laws governing a large part of physics and the whole chemistry are very well known and experimentally verified. This is at the heart of the great success of quantum mechanics in explaining many phenomena, including the physical and chemical properties of matter.

Despite this success, however *solving* the fundamental equations of quantum mechanics is an intrinsically complex task that is unparalleled in other fields. In the vast majority of cases, only numerical solutions of these equations are feasible and, as we will see during this course, there are many fundamental problems that are fundamentally hard to solve, even with the most powerful computational platforms we have available. In those cases, only approximate solutions are feasible.

The goal of this course is to introduce you to the modern simulation techniques for quantum physics, through lectures and practical programming exercises. We will do a journey through several simulation approaches based both on classical and quantum computers.

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

For students from **other curricula** (for example engineering or computer science) the computational quantum physics courses is strongly suggested for students interested in physics-related applications.

1.1.1 Exercises

The exercises form a very important part of the course, and it's where you will learn how to apply the techniques presented during the theory sessions. It is imperative that you attend the exercise sessions regularly, so to master the necessary practical implementations of the methods.

1.1.1.1 Programming Languages

You are free to choose any programming language you like, we will not impose any specific programming language. It should be noticed however that solutions to the exercises will most exclusively be given as PYTHON scripts. If you decide to use another programming language, it would be advisable to consult with the assistants first, to assess whether they are fluent in the language you choose, in case you need help to debug your codes.

1.1.1.2 Computer Access

For all the exercises we will assume you have a laptop with a working internet connection.

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.

1.1.2.1 Computing

- Basic knowledge of UNIX
- Good working knowledge of at least one common programming language (Python, C, C++, Fortran, Julia...).
- Knowledge of Matlab is typically sufficient, but it is strongly advised to be familiar with Python, since the exercises will be typically presented and discussed in Python.
- Ability to produce graphical plots.

1.1.2.2 Numerical Analysis

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

1.1.2.3 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. However, immediately contact the lecturer if you feel that the material of the first week is too advanced.

1.1.3 References

Here is a list of a few good references for the course, in addition to these lecture notes.

1. Joshua Izaac, and Jingbo Wang, *Computational Quantum Mechanics* , Springer International Publishing (2018) ISBN 9783319999296.
2. J.M. Thijssen, *Computational Physics*, Cambridge University Press (1999) ISBN 0521575885.
3. Federico Becca, and Sandro Sorella, *Quantum Monte Carlo approaches for correlated systems*, Cambridge University Press (2017) ISBN 9781316417041.

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 quantum chemistry. 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
- Quantum algorithms and emulation of quantum computers
- Machine-learning based approaches to quantum many-body problems

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 $\hat{\rho}$,

$$\hat{\rho} = \sum_i P_i|\Psi_i\rangle\langle\Psi_i|, \tag{2.8}$$

describing a mixture of pure states $|\Psi_i\rangle$ each of which carrying a probability P_i. It is easy to check that the density matrix has unit trace:

$$\begin{aligned}\mathrm{Tr}\hat{\rho} &= \sum_x\sum_i P_i\langle x|\Psi_i\rangle\langle\Psi_i|x\rangle && (2.9)\\ &= \sum_i P_i\sum_x|\langle x|\Psi_i\rangle|^2 && (2.10)\\ &= \sum_i P_i && (2.11)\\ &= 1. && (2.12)\end{aligned}$$

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

$$\hat{\rho}_{\mathrm{pure}} = |\Psi\rangle\langle\Psi|. \tag{2.13}$$

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

$$\hat{\rho}_{\rightarrow} = |\rightarrow\rangle\langle\rightarrow| = \begin{pmatrix}1/2 & 1/2\\1/2 & 1/2\end{pmatrix}. \tag{2.14}$$

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

$$\hat{\rho}_{\mathrm{mixed}} = \begin{pmatrix}1/2 & 0\\0 & 1/2\end{pmatrix}. \tag{2.15}$$

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 $\hat{S}_x$, $\hat{S}_y$ and $\hat{S}_z$ of the spin in the x-, y-, and z-directions are represented by the Pauli matrices

$$\hat{S}_x = \frac{\hbar}{2}\hat{\sigma}_x = \frac{\hbar}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{2.16}$$

$$\hat{S}_y = \frac{\hbar}{2}\hat{\sigma}_y = \frac{\hbar}{2}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{2.17}$$

$$\hat{S}_z = \frac{\hbar}{2}\hat{\sigma}_z = \frac{\hbar}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2.18}$$

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

$$\mathbf{e}\cdot\hat{\mathbf{S}} = e_x\hat{S}_x + e_y\hat{S}_y + e_z\hat{S}_z = \frac{\hbar}{2}\begin{pmatrix} e_x & e_x - ie_y \\ e_x + ie_y & -e_z \end{pmatrix} \tag{2.19}$$

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

$$[\hat{S}_x, \hat{S}_y] = \hat{S}_x\hat{S}_y - \hat{S}_y\hat{S}_x = i\hbar\hat{S}_z, \tag{2.20}$$

$$[\hat{S}_y, \hat{S}_z] = i\hbar\hat{S}_x, \tag{2.21}$$

$$[\hat{S}_z, \hat{S}_x] = i\hbar\hat{S}_y, \tag{2.22}$$

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 the associated operator $\hat{A}$ 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

$$|\Psi\rangle = |\uparrow\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{2.23}$$

or alternatively density matrix

$$\hat{\rho}_\uparrow = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \tag{2.24}$$

and we measure the x-component of the spin $\hat{S}_x$, 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

$$\begin{aligned} | \rightarrow\rangle &= \frac{1}{\sqrt{2}}(| \uparrow\rangle + | \downarrow\rangle) = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} && (2.25) \\ | \leftarrow\rangle &= \frac{1}{\sqrt{2}}(| \uparrow\rangle - | \downarrow\rangle) = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix} && (2.26) \end{aligned}$$

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

$$\begin{aligned} p_\rightarrow &= ||\langle\rightarrow |\Psi\rangle||^2 = 1/2 && (2.27) \\ p_\leftarrow &= ||\langle\leftarrow |\Psi\rangle||^2 = 1/2 && (2.28) \end{aligned}$$

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

$$\hat{\rho} = p_\rightarrow| \rightarrow\rangle\langle\rightarrow | + p_\leftarrow| \leftarrow\rangle\langle\leftarrow | = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}. \tag{2.29}$$

which differs from the initial density matrix $\hat{\rho}_\uparrow$.

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

$$\langle A\rangle = \langle\Psi|\hat{A}|\Psi\rangle \tag{2.30}$$

or from a density matrix $\hat{\rho}$ as

$$\langle A\rangle = \mathrm{Tr}(\hat{\rho}\hat{A}). \tag{2.31}$$

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

$$\mathrm{Tr}(\hat{\rho}_\Psi\hat{A}) = \mathrm{Tr}(|\Psi\rangle\langle\Psi|\hat{A}) = \langle\Psi|\hat{A}|\Psi\rangle. \tag{2.32}$$

2.2.5 The uncertainty relation

If two observables A and B do not commute $[\hat{A}, \hat{B}] \neq 0$, 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

$$[\hat{A}, \hat{B}] = i\hbar \tag{2.33}$$

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

$$\Delta A\Delta B \geq \hbar/2 \tag{2.34}$$

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

2.2.6.1 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

$$i\hbar\frac{\partial}{\partial t}|\Psi(t)\rangle = \hat{H}|\Psi(t)\rangle, \tag{2.35}$$

where $\hat{H}$ is the Hamiltonian operator. This is just a first order linear differential equation.

2.2.6.2 The time-independent Schrödinger equation

The so-called time-independent Schrödinger equation is nothing but the eigenvalue problem for the Hamiltonian operator:

$$\hat{H}|\Psi_k\rangle = E_k|\Psi_k\rangle, \tag{2.36}$$

where E_k is the energy associated with the $k-$th eigen-ket $|\Psi_k\rangle$. A great deal of this course will be spent solving "just" this simple eigenvalue problem!

Knowing the eigen-kets of the Hamiltonian is very useful, for example, to solve the time-dependent Schrödinger equation, since

$$\begin{aligned} |\Psi(t)\rangle &= \exp(-i\hat{H}t/\hbar)|\Psi(0)\rangle && (2.37) \\ &= \sum_k c_k \exp(-iE_kt/\hbar)|\Psi_k\rangle && (2.38) \end{aligned}$$

where $c_k = \langle\Psi_k|\Psi(0)\rangle$ are the overlaps of the initial state with eigen-kets of the Hamiltonian.

2.2.6.3 The Schrödinger equation for the density matrix

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

$$i\hbar\frac{\partial}{\partial t}\hat{\rho}(t) = [\hat{H}, \hat{\rho}(t)] \tag{2.39}$$

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 $\beta = 1/k_BT$. In a classical system each microstate k of energy E_k is occupied with a probability given by the Boltzmann distribution

$$p_k = \frac{1}{Z}\exp(-\beta E_k), \tag{2.40}$$

where the partition function

$$Z = \sum_k \exp(-\beta E_k) \tag{2.41}$$

normalizes the probabilities.

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

$$\hat{\rho}_\beta = \frac{1}{Z} \sum_k \exp(-\beta E_k) |\Psi_k\rangle\langle\Psi_k| \tag{2.42}$$

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

$$\hat{\rho}_\beta = \frac{1}{Z} \exp(-\beta \hat{H}) \tag{2.43}$$

where the partition function now is

$$Z = \operatorname{Tr} \exp(-\beta \hat{H}) \tag{2.44}$$

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

$$\langle A \rangle = \operatorname{Tr}(\hat{A}\hat{\rho}_\beta) = \frac{\operatorname{Tr} \hat{A} \exp(-\beta \hat{H})}{\operatorname{Tr} \exp(-\beta \hat{H})}, \tag{2.45}$$

but actually evaluating this expression is a hard problem, as we will see during this course.

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 (2.16)-(2.16) 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 $\hat{S}_z$ operator

$$\hat{S}_z |s\rangle = \hbar s |s\rangle, \tag{2.46}$$

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 $\hat{S}_x$ and $\hat{S}_y$ operators can be constructed from the "ladder operators"

$$\hat{S}_+ |s\rangle = \sqrt{S(S+1) - s(s+1)} |s+1\rangle \tag{2.47}$$

$$\hat{S}_- |s\rangle = \sqrt{S(S+1) - s(s-1)} |s-1\rangle \tag{2.48}$$

which increment or decrement the $\hat{S}_z$ value by 1 through

$$\begin{aligned}\hat{S}_x &= \frac{1}{2}\left(\hat{S}_+ + \hat{S}_-\right) && (2.49)\\ \hat{S}_y &= \frac{1}{2i}\left(\hat{S}_+ - \hat{S}_-\right). && (2.50)\end{aligned}$$

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

$$\hat{H} = -g\mu_B \mathbf{h}\cdot\hat{\mathbf{S}}, \tag{2.51}$$

which introduces nontrivial dynamics since the components of $\hat{\mathbf{S}}$ 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 $\Psi(\mathbf{x})$ in n-dimensional space. In this representation the operator $\hat{\mathbf{x}}$, measuring the position of the particle is simple and diagonal

$$\hat{\mathbf{x}} = \mathbf{x}, \tag{2.52}$$

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

$$\hat{p}_\alpha = -i\hbar\hat{\nabla}_\alpha. \tag{2.53}$$

These two operators do not commute but their commutator is

$$[\hat{x}_\alpha, \hat{p}_\beta] = i\hbar\delta_{\alpha,\beta}. \tag{2.54}$$

The Schrödinger equation of a quantum particle in an external potential $V(\mathbf{x})$ 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

$$H(\mathbf{x},\mathbf{p}) = \frac{|\mathbf{p}|^2}{2m} + V(\mathbf{x}) \tag{2.55}$$

we use the quantum mechanical Hamiltonian operator

$$\hat{H} = \frac{|\hat{\mathbf{p}}|^2}{2m} + V(\hat{x}) = -\frac{\hbar^2}{2m}\nabla^2 + V(\hat{x}), \tag{2.56}$$

which gives the famous form

$$i\hbar\frac{\partial\psi(\mathbf{x},t)}{\partial t} = -\frac{\hbar^2}{2m}\nabla^2\psi(\mathbf{x},t) + V(\mathbf{x})\psi(\mathbf{x},t) \tag{2.57}$$

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 $\hat{x}$ in units where $m = \hbar = K = 1$, the time-independent Schrödinger equation is given by

$$\hat{H}|n\rangle = \frac{1}{2}(\hat{p}^2 + \hat{x}^2)|n\rangle = E_n|n\rangle \tag{2.58}$$

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

$$-\frac{1}{2}\phi_n''(x) + \frac{1}{2}x^2\phi_n(x) = E_n\phi_n(x) \tag{2.59}$$

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

$$\phi_n(x) = \frac{1}{\sqrt{2^n n! \sqrt{\pi}}} \exp\left(-\frac{1}{2}x^2\right) H_n(x), \tag{2.60}$$

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

$$\langle m|\hat{x}|n\rangle \qquad \text{and} \qquad \langle m|\hat{p}|n\rangle \tag{2.61}$$

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

$$\hat{x} = \frac{1}{\sqrt{2}}(\hat{a}^\dagger + \hat{a}) \tag{2.62}$$

$$\hat{p} = \frac{1}{i\sqrt{2}}(\hat{a}^\dagger - \hat{a}), \tag{2.63}$$

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

$$\langle n+1|\hat{a}^\dagger|n\rangle = \langle n|\hat{a}|n+1\rangle = \sqrt{n+1}. \tag{2.64}$$

and commutation relations

$$[\hat{a}, \hat{a}] = [\hat{a}^\dagger, \hat{a}^\dagger] = 0 \tag{2.65}$$

$$[\hat{a}, \hat{a}^\dagger] = 1. \tag{2.66}$$

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

$$\hat{n}|n\rangle = \hat{a}^\dagger\hat{a}|n\rangle = \sqrt{n}\hat{a}^\dagger|n-1\rangle = n||n\rangle. \tag{2.67}$$

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

$$\begin{aligned}H &= \frac{1}{2}(\hat{p}^2+\hat{x}^2)\\ &= \frac{1}{4}\left[-(\hat{a}^\dagger-\hat{a})^2+(\hat{a}^\dagger+\hat{a})^2\right]\\ &= \frac{1}{2}(\hat{a}^\dagger\hat{a}+\hat{a}\hat{a}^\dagger)\\ &= \frac{1}{2}(2\hat{a}^\dagger\hat{a}+1)=\hat{n}+\frac{1}{2}, \qquad (2.68)\end{aligned}$$

which has the correct spectrum. In deriving the last lines we have used the commutation relation (2.66).

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-independent 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

$$-\frac{1}{2}\frac{\partial^2\psi(x)}{\partial x^2}+V(x)\psi(x)=E\psi(x), \tag{3.1}$$

where we have taken units in which $\hbar=m=1$.

In order to efficiently solve this problem on a computer, in the great majority of numerical approaches it is necessary to introduce some *discretization* of the problem. The simplest discretization we can perform consists in discretizing the space. We then consider $x\in[x_0,x_p]$ living in a finite interval, and a discretization of the space into a mesh of uniform spacing δ. The points on the mesh are such that

$$x_n=n\delta+x_0, \tag{3.2}$$

where x_0 is the starting point of the interval and the wave function at these points is denoted by

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

In the following we will assume that the mesh contains a total of $p+1$ points, thus implying that the last point in the interval is

$$x_p=p\delta+x_0. \tag{3.4}$$

3.1.1 Discretizing the Hamiltonian

The previous discussion has introduced a simple strategy to represent the wave function as a finite-size vector of components ψ_n. In order to solve Schroedinger's equation, it is now necessary also to have finite-sized description of the Hamiltonian. This must be a matrix acting on the same vector space of the discretized state ψ_n. The potential energy term in the hamiltonian is diagonal in the position basis and it is easily discretized into a diagonal matrix with diagonal entries V_n:

$$V(x) \quad \rightarrow \quad V_n \equiv V(x_n). \tag{3.5}$$

The kinetic energy term is off-diagonal in the position basis and we can also expect that its matrix representation is off diagonal. In order to find an explicit representation, we can use a finite-difference approximation of the derivative:

$$\frac{\partial^2}{\partial x^2} f(x_n) \simeq \frac{1}{\delta^2} \left(f(x_{n-1}) - 2f(x_n) + f(x_{n+1}) \right) + \mathcal{O}(\delta^2). \tag{3.6}$$

Using this discretization, the action of the hamiltonian on the wave function is written as

$$\langle x_n | \hat{H} | \psi \rangle = -\frac{1}{2\delta^2} \left(\psi_{n-1} - 2\psi_n + \psi_{n+1} \right) + V_n \psi_n + \mathcal{O}(\delta^2). \tag{3.7}$$

The right-hand side of this equation contains only linear combinations of the vector ψ_n and it is thus linear operator that corresponds to the discretization of the original Hamiltonian. It is easy to see that this linear operator is a matrix :

$$\hat{H}_\delta = \begin{pmatrix} \dots & -\frac{1}{2\delta^2} & 0 & 0 & \dots \\ -\frac{1}{2\delta^2} & V_{n-1} + \frac{1}{\delta^2} & -\frac{1}{2\delta^2} & 0 & \dots \\ 0 & -\frac{1}{2\delta^2} & V_n + \frac{1}{\delta^2} & -\frac{1}{2\frac{1}{2\delta^2}} & \dots \\ 0 & 0 & -\frac{1}{2\delta^2} & V_{n+1} + \frac{1}{\delta^2} & \dots \\ \dots & 0 & 0 & \dots & \dots \end{pmatrix} \tag{3.8}$$

Notice that we explicitly omitted, for the moment, the value of the matrix on the boundary. This is because the second-order finite difference scheme we have chosen would act also on $\psi(x_0 - \delta) \equiv \psi_{-1}$ and on $\psi(x_p + \delta) \equiv \psi_{p+1}$, but these are beyond the discretized region we have chosen for the vector ψ_n.

In the following we will concentrate on the conceptually easy (and practically relevant) case in which we have a bound state, thus the wave-function goes to zero at infinity. In this case, we can always choose an interval $[x_0, x_p]$ large enough such that, to good approximation $\psi_{-1} = \psi_{p+1} = 0$. In this case then the matrix above is exactly tridiagonal, since it is safe to just ignore the extra-boundary points ψ_{-1}, ψ_{p+1}.

For bound states then, finding solutions to the time-independent Schroedinger equation is a simple as diagonalizing the finite-dimensional matrix $\hat{H}_\delta$, and find the eigenvectors and eigen-energies

$$\hat{H}_\delta |\psi_k\rangle \quad = \quad E_k |\psi_k\rangle.$$

An interesting property of the matrix $\hat{H}_\delta$ is that it is tridiagonal, and it can be very efficiently diagonalized, as you will see more in detail in the exercises.

3.2 The time-independent Schrödinger equation in higher dimensions

In higher dimensions, in most common cases it is possible to reduce the problem to a one-dimensional problem. This happens if the problem, because of symmetries, 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

$$\psi(\vec{r}) = \psi_x(x)\psi_y(y)\psi_z(z) \tag{3.9}$$

the Schroedinger's equation factorizes into three one-dimensional equations 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

$$\psi_{l,m}(\vec{r}) = \psi_{l,m}(r, \theta, \phi) = \frac{u(r)}{r} Y_{lm}(\theta, \phi) \tag{3.10}$$

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

$$\left[-\frac{\hbar^2}{2\mu}\frac{d^2}{dr^2} + \frac{\hbar^2 l(l+1)}{2\mu r^2} + V(r)\right] u(r) = Eu(r) \tag{3.11}$$

where we have called the particle mass μ (to avoid confusion with magnetic quantum number m in the spherical harmonics). This is again a one-dimensional Schrödinger equation, with a modified effective potential

$$V_l(r) = V(r) + \frac{\hbar^2}{2\mu}\frac{l(l+1)}{r^2} \tag{3.12}$$

and with the radial wave-function defined in the interval $[0, \infty[$. In practice, for regular potentials we always have $u(0) = u(x_i) = 0$ and it is always possible to find a point $x_p \gg 1$ such that, with good approximation, $u(x_p) = 0$.

3.2.3 Finite difference methods in higher dimension

In higher dimension we can still discretize the space on a regular grid (for example, on a square grid in two dimensions). By doing so, we obtain once more a matrix representation of the kinetic energy. The Laplacian in two dimensions for example takes this form

$$\begin{aligned} \nabla^2\psi(x_n, y_n) &= \frac{1}{\delta^2}\left[\psi(x_{n+1}, y_n) - 2\psi(x_n, y_n) + \psi(x_{n-1}, y_n)\right] + \\ &\quad + \frac{1}{\delta^2}\left[\psi(x_n, y_{n+1}) - 2\psi(x_n, y_n) + \psi(x_n, y_{n-1})\right] \end{aligned} \tag{3.13}$$

While the resulting discretized Hamiltonian in general will not be of tridiagonal form as in the 1d case, it is essential to realize that the matrices produced by the discretization of the Schrödinger equation are still very sparse, meaning that only a small fraction of the matrix entries are non zero. For these sparse systems of equations, optimized iterative numerical algorithms exist[1] and are implemented in numerical libraries such as in the EIGEN library (C++) [2] or in SciPy (Python) [3]. 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 (which will be discussed in detail in the following lectures) are best. Again there exist efficient implementations of iterative algorithms for sparse matrices.

3.3 The time-dependent Schrödinger equation

We now move to the problem of solving the time-dependent Schrödinger equation

$$i\frac{\partial}{\partial t}\psi(x,t) = -\frac{1}{2}\frac{\partial^2\psi(x,t)}{\partial x^2} + V(x)\psi(x,t), \tag{3.14}$$

with given initial condition $\psi(x,t_0)$.

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 (time-independent) 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 $\hat{H}|\phi\rangle = E|\phi\rangle$ and calculate the eigenvectors $|\phi_n\rangle$ and eigenvalues ϵ_n. Next we represent the initial wave function $|\psi\rangle$ by a spectral decomposition

$$|\psi(t_0)\rangle = \sum_n c_n|\phi_n\rangle. \tag{3.15}$$

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

$$|\psi(t)\rangle = \sum_n c_n e^{-i\epsilon_n(t-t_0)}|\phi_n\rangle. \tag{3.16}$$

This approach is, however, only useful for very small problems since the effort of diagonalizing the matrix is huge. A better method is direct numerical integration, discussed in the next subsections.

[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] https://eigen.tuxfamily.org/

[3] https://docs.scipy.org/, the relevant routines are contained in *scipy.sparse.linalg*

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, instead of the spectral method it is more convenient 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 $\hat{H}_\delta$ to refer the representation of the Hamiltonian in the chosen finite basis. A simple ODE integration scheme is the forward Euler scheme

$$|\psi(t_{n+1})\rangle = |\psi(t_n)\rangle - i\delta_t \hat{H}_\delta |\psi(t_n)\rangle. \tag{3.17}$$

However, this method is not only numerically unstable, but it also violates the conservation of the norm of the wave function $\langle\psi(t)|\psi(t)\rangle = 1$, since

$$(1 - i\delta_t \hat{H}_\delta)\left(1 - i\delta_t \hat{H}_\delta\right)^\dagger \;=\; 1 + \delta_t^2 \hat{H}^2 \neq 1.$$

The exact quantum evolution

$$|\psi(t+\delta_t)\rangle = e^{-i\hat{H}_\delta \delta_t}|\psi(t)\rangle \tag{3.18}$$

is however clearly unitary and thus conserves the norm, we therefore want to look for a unitary approximation as integrator. Instead of using the forward Euler method (3.17) which is just a first order Taylor expansion of the exact time evolution

$$e^{-iH\delta_t} = 1 - i\delta_t \hat{H}_\delta + \mathrm{O}(\delta_t^2), \tag{3.19}$$

we reformulate the time evolution operator as

$$e^{-iH\delta_t} = \left(e^{iH\delta_t/2}\right)^{-1} e^{-iH\delta_t/2} = \left(1 + \frac{i\delta_t}{2}\hat{H}_\delta\right)^{-1}\left(1 - \frac{i\delta_t}{2}\hat{H}_\delta\right) + \mathrm{O}(\delta_t^3), \tag{3.20}$$

therefore

$$|\psi(t+\delta_t)\rangle = \left(1 + \frac{i\delta_t}{2}\hat{H}_\delta\right)^{-1}\left(1 - \frac{i\delta_t}{2}\hat{H}_\delta\right)|\psi(t)\rangle, \tag{3.21}$$

It is possible to check that the propagation scheme defined above is unitary (for example, showing that the two terms appearing commute with one another), thus we have a small time-step approach that is both unitary and second-order in δ_t. Equivalently, we can write it as

$$\left(1 + \frac{i\delta_t}{2}\hat{H}_\delta\right)|\psi(t+\delta_t)\rangle = \left(1 - \frac{i\delta_t}{2}\hat{H}_\delta\right)|\psi(t)\rangle, \tag{3.22}$$

which shows more explicitly that, unfortunately this is an implicit integrator, since the value of $|\psi(t+\delta_t)\rangle$ is not a simple linear combination of the wave-function values at the previous time-steps.

3.3.2.1 Practical implementations of the implicit scheme

Despite its implicit nature, this integrator can be still be used efficiently. Concentrating again on the 1d case, we have seen previously that if we discretize our problem on a mesh $x_n = n\delta + x_0$, that the Hamiltonian becomes a simple tridiagonal matrix. The implicit equation 3.22 then becomes a linear system of the form $\hat{A}y = b$, where the matrix $\hat{A} = \left(1 + \frac{i\delta_t}{2}\hat{H}_\delta\right)$, the right hand side is $b_n = \left(1 - \frac{i\delta_t}{2}\right)\psi_n(t)$ and the unknown vector $y_n = \psi_n(t + \delta_t)$.

At each time step, one can therefore solve this linear system of equations and find explicitly $\psi_n(t + \delta_t)$. Because of the tridiagonal structure, very efficient 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 EIGEN Library C++, or in SciPy, for a Python version.

3.4 Appendix: The split operator method

An alternative to the unitary, implicit method described in the main text exists, and we discuss it as an optional argument for the interested reader. An explicit and unitary method is possible for a quantum particle in the real space picture with the "standard" Schrödinger equation for non-relativistic particles in continuous space. Writing the Hamiltonian operator as

$$H = \hat{T} + \hat{V} \tag{3.23}$$

with

$$\hat{T} = \frac{1}{2m}\hat{p}^2 \tag{3.24}$$

$$\hat{V} = V(\vec{x}) \tag{3.25}$$

it is easy to see that $\hat{V}$ is diagonal in position space while $\hat{T}$ is diagonal in momentum space.

Indeed if we consider a d-dimensional particle, its wave-function in momentum space is obtained through the Fourier transform:

$$\tilde{\psi}(\vec{k}) = \left(\frac{1}{\sqrt{2\pi}}\right)^d \int_{-\infty}^{\infty} \psi(\vec{x}) e^{-i\vec{k}\cdot x} d\vec{x} \tag{3.26}$$

and the inverse Fourier transform yields

$$\psi(\vec{x}) = \left(\frac{1}{\sqrt{2\pi}}\right)^d \int_{-\infty}^{\infty} \tilde{\psi}(\vec{k}) e^{i\vec{k}\cdot x} d\vec{k}. \tag{3.27}$$

It is then easy to check that $\tilde{\psi}(\vec{k})$ is an eigenstate of the kinetic operator T, and that $T\tilde{\psi}(\vec{k}) = ||\vec{k}||^2\tilde{\psi}(\vec{k})/2$.

If we split the time evolution as

$$e^{-i\Delta_t H/\hbar} = e^{-i\Delta_t \hat{V}/2\hbar} e^{-i\Delta_t \hat{T}/\hbar} e^{-i\Delta_t \hat{V}/2\hbar} + \mathrm{O}(\Delta_t^3) \tag{3.28}$$

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

$$\begin{aligned} \left[e^{-i\Delta_t \hat{V}/2\hbar}|\psi\rangle\right](\vec{x}) &= e^{-i\Delta_t V(\vec{x})/2\hbar}\psi(\vec{x}) & (3.29) \\ \left[e^{-i\Delta_t \hat{T}/\hbar}|\psi\rangle\right](\vec{k}) &= e^{-i\Delta_t \hbar ||\vec{k}||^2/2m}\psi(\vec{k}) & (3.30) \end{aligned}$$

in real space for the first term and momentum space for the second term.

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

$$\begin{aligned} e^{-iHt/\hbar} &= \left(e^{-i\Delta_t \hat{V}/2\hbar} e^{-i\Delta_t \hat{T}/\hbar} e^{-i\Delta_t \hat{V}/2\hbar}\right)^N + \mathrm{O}(\Delta_t^2) \\ &= e^{-i\Delta_t \hat{V}/2\hbar}\left[e^{-i\Delta_t \hat{T}/\hbar} e^{-i\Delta_t \hat{V}/\hbar}\right]^{N-1} e^{-i\Delta_t \hat{T}/\hbar} e^{-i\Delta_t \hat{V}/2\hbar} & (3.31) \end{aligned}$$

In practice, in order to obtain efficient representations of the wave-functions both in real and momentum space we still need to discretize the real space with a suitable mesh of size Δx, for a total of P points per spatial direction. As a consequence of this discretization, the continuous Fourier transform becomes a discrete Fourier transform defined on the discrete set of wave-vectors $k_n = \frac{2\pi}{n}P$, for each spatial direction, with $n = 0, 1, \ldots P-1$. Changing from real space to momentum space then requires the application of the discrete Fourier transform and of its inverse when going back from momentum space to real space. This can be efficiently accomplished numerically thanks to the Fast Fourier Transform (FFT) algorithm, which performs the discrete Fourier transform in only $\mathcal{O}(P\log(P))$ operations.

The discretized algorithm then starts as

$$\begin{aligned} \psi_1(\vec{x}) &= e^{-i\Delta_t V(\vec{x})/2\hbar}\psi_0(\vec{x}) & (3.32) \\ \psi_1(\vec{k}) &= \mathcal{F}\psi_1(\vec{x}) & (3.33) \end{aligned}$$

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:

$$\begin{aligned} \psi_{2n}(\vec{k}) &= e^{-i\Delta_t \hbar ||\vec{k}||^2/2m}\psi_{2n-1}(\vec{k}) & (3.34) \\ \psi_{2n}(\vec{x}) &= \mathcal{F}^{-1}\psi_{2n}(\vec{k}) & (3.35) \\ \psi_{2n+1}(\vec{x}) &= e^{-i\Delta_t V(\vec{x})/\hbar}\psi_{2n}(\vec{x}) & (3.36) \\ \psi_{2n+1}(\vec{k}) &= \mathcal{F}\psi_{2n+1}(\vec{x}) & (3.37) \end{aligned}$$

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

$$\psi_{2N+1}(\vec{x}) = e^{-i\Delta_t V(\vec{x})/2\hbar}\psi_{2N}(\vec{x}) \tag{3.38}$$

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 (3.28) by a fifth-order version involving five instead of three terms.

Chapter 4

Exact diagonalization of many-body problems

4.1 Quantum spin models

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 $\text{dim}\mathcal{H} = d$ 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.

In this Chapter we will specifically focus on quantum spin-1/2 particles. 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. This exponential scaling of the Hilbert space dimension with the number of particles is a big challenge. The basis for $N = 30$ spins is already of size $2^{30} \approx 10^9$. A single complex vector needs 16 GByte of memory and may just barely fit into the memory of your personal computer.

For small and moderately sized systems of up to about 30 spin-1/2 we can calculate exactly the ground state, low-lying spectrum, and time evolution by direct calculations. For more than 30 spins we cannot apply exact diagonalization techniques anymore, and this will be the subject of several methods we will study in the next chapters.

4.1.1 Hamiltonian Matrix

As we have seen already in the previous Chapter, to perform exact diagonalization to find eigenstates of a given Hamiltonian, $\hat{H}$, or study its dynamics, it is important to come up with a concrete representation of the Hamiltonian matrix that can be efficiently manipulated on a computer. One common feature of many-body quantum models is that the matrix representation of their hamiltonian is *sparse*. For example, taking again the case of quantum spins, one can see that the total number of non-zero elements in the matrix representation of the Hamiltonian is at most $k \times 2^N$, where k is typically a small value (in most cases, $k \sim N$). This is to be contrasted to a general, full matrix, that instead contains $\mathcal{O}(2^N \times 2^N)$ elements. Sparsity–a generalization of the simple pattern of non-zero elements seen in tridiagonal matrices in the previous Chapter– can be exploited by exact diagonalization methods in different ways, both to find eigenvalues and eigenvectors of the Hamiltonian and to study its dynamics. Before seeing how sparsity can be exploited, we will first analyze a few prototypical spin models, in order to better understand where the sparse nature of the Hamiltonian matrix comes from. In all cases we will analyze in this Chapter we will consider the simple, and widely adopted, basis of eigenstates of $\hat{\sigma}^z$. Specifically, each many-spin state is written as a linear combination of 2^N basis states:

$$|\Psi\rangle = \sum_{s_1 s_2 \dots s_N} c_{s_1 s_2 \dots s_N} |s_1 s_2 \dots s_N\rangle, \tag{4.2}$$

where

$$|s_1 s_2 \dots s_N\rangle = |s_1\rangle \otimes |s_2\rangle \otimes \dots |s_N\rangle \tag{4.3}$$

are eigen-kets of the $\hat{\sigma}^z$ Pauli matrix:

$$\hat{\sigma}_i^z |s_1 s_2 \dots s_N\rangle = s_i |s_1 s_2 \dots s_N\rangle, \tag{4.4}$$

for $s_i = \pm 1$.

4.1.2 Example: the transverse-field Ising model

The simplest quantum spin model is probably the quantum transverse field Ising model (TFIM), which adds a magnetic field in the x direction to a lattice of spin-1/2 particles coupled by an Ising interaction:

$$\hat{H} = \sum_{\langle i,j \rangle} J_{ij} \hat{\sigma}_i^z \hat{\sigma}_j^z - \Gamma \sum_i \hat{\sigma}_i^x. \tag{4.5}$$

Here the symbol $\langle i, j\rangle$ denotes a sum over all bonds in the lattice. In the absence of the second term, the first term is nothing but a classical Ising model and can be solved by your favorite method of simulating the Ising model. The second term does not exist in classical Ising models, where the spins point only in the z direction. Considering that the Pauli $\hat{\sigma}^x$ matrix is

$$\hat{\sigma}^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{4.6}$$

we see that this term flips an ↑ spin to a ↓ spin, and thus introduces quantum fluctuations to the classical Ising model.

The way of writing the hamiltonian as above is nothing but a short-hand for the more laborious (but more precise) notation with tensor products, that in this case would imply for example that a spin operator in the direction $\alpha = (x, y, z)$ and acting on spin i is in reality the following $2^N \times 2^N$ matrix:

$$\hat{\sigma}_i^\alpha \equiv \underbrace{\hat{I} \otimes \hat{I} \otimes \dots \hat{I}}_{i-1\,\text{times}} \otimes \hat{\sigma}^\alpha \otimes \underbrace{\hat{I} \otimes \dots \otimes \hat{I}}_{N-i\,\text{times}} \tag{4.7}$$

$$= \hat{I}(2^{i-1}) \otimes \hat{\sigma}^\alpha \otimes \hat{I}(2^{N-i}), \tag{4.8}$$

where $\hat{I}(n)$ are identity matrices of dimension n, and the $\otimes$ product here denotes Kronecker product between matrices.

We can readily verify that this Hamiltonian is sparse. For example, let's start computing the diagonal matrix elements in the basis specified above:

$$\langle s_1 s_2 \dots s_N | \hat{H} | s_1 s_2 \dots s_N \rangle \;=\; \sum_{\langle i,j \rangle} J_{ij} s_i s_j, \tag{4.9}$$

which is the familiar classical Ising interaction term. Thus we have found the first 2^N (in general) non-zero matrix elements, corresponding to the diagonal of $\hat{H}$. The off-diagonal terms can be readily found noticing that the action of the $\hat{\sigma}_i^x$ operator is just to flip a spin:

$$\hat{\sigma}_i^x | s_1 \dots s_i \dots s_N \rangle \;=\; | s_1 \dots - s_i \dots s_N \rangle, \tag{4.10}$$

thus there is only one non-zero matrix element per $\hat{\sigma}_i^x$ term:

$$\langle s_1' s_2' \dots s_N' | \hat{\sigma}_i^x | s_1 s_2 \dots s_N \rangle \;=\; \delta_{s_1', s_1} \dots \delta_{s_i' - s_i} \dots \delta_{s_N', s_N} \tag{4.11}$$

implying that, at fixed $|s_1 s_2 \dots s_N\rangle$, there is a total of N non zero matrix elements for the Hamiltonian. In total, therefore, we have that the TFI Hamiltonian contains “only” $(N+1) \times 2^N$ non-zero elements.

4.2 Finding Ground States

We start with the problem of finding the lowest eigenvector (and its energy) of the Hamiltonian, the so-called ground state. This task is realized by using an iterative matrix eigensolver. These solvers exploit the fact that computing the product of the Hamiltonian matrix with an arbitrary vector can be done efficiently. While for a generic matrix of size $M \times M$ a product $\hat{A}|v\rangle$ can be computed in $\mathcal{O}(M^2)$ time, for a matrix $M \times M$, in the case of Hamiltonians we are considering here this product is computable in only $\mathcal{O}(M) = \mathcal{O}(N^\alpha \times 2^N)$, where α is in general a small exponent ($\alpha = 1$, for the TFIM, as seen before).

4.2.1 Power Method

The power method is the simplest iterative solver we can use to find ground-states of many-body Hamiltonians that exploits sparseness. This method generates a sequence of P vectors $k = 1, \dots P$ by repeated application of the Hamiltonian:

$$|u_{k+1}\rangle \;=\; \left(\Lambda \hat{I} - \hat{H}\right) |u_k\rangle, \tag{4.12}$$

where Λ is a suitable constant, and the initial state $|u_0\rangle$ is given as starting condition for the algorithm. This sequence of vectors converges to the ground-state of the Hamiltonian under reasonable assumptions. To see this, let us formally expand the initial vector in terms of the eigen-states of the Hamiltonian:

$$|u_0\rangle \;=\; \sum_l c_l |E_l\rangle, \tag{4.13}$$

with $E_0 \leq E_1 \leq \ldots E_M$, thus the last state is

$$
\begin{aligned}
|u_P\rangle &= \left(\Lambda\hat{I} - \hat{H}\right)^P |u_0\rangle, && (4.14)\\
&= \sum_l \left(\Lambda - E_l\right)^P c_l |E_l\rangle, && (4.15)
\end{aligned}
$$

and the overlap with the ground state is

$$
\langle E_0 | u_P\rangle = \left(\Lambda - E_0\right)^P c_0. \qquad (4.16)
$$

We notice however that the state $|u_P\rangle$ is not normalized in general, thus the probability amplitude of being in the ground-state after k iterations is

$$
\begin{aligned}
\frac{|\langle E_0|u_P\rangle|^2}{\langle u_P|u_P\rangle} &= \frac{\left(\Lambda - E_0\right)^{2P}|c_0|^2}{\left(\Lambda - E_0\right)^{2P}|c_0|^2 + \left(\Lambda - E_1\right)^{2P}|c_1|^2 + \ldots} && (4.17)\\
&= \frac{1}{1 + \frac{(\Lambda-E_1)^{2P}}{(\Lambda-E_0)^{2P}}\frac{|c_1|^2}{|c_0|^2} + \ldots}. && (4.18)
\end{aligned}
$$

From this expression we can see that a suitable choice of Λ can force the state $|u_P\rangle$ have a probability amplitude of being in the ground state that is *exponentially* close to 1. Specifically, if we impose $\Lambda > E_M$, we have that

$$
\lim_{P\to\infty} \frac{\left(\Lambda - E_l\right)^{2P}}{\left(\Lambda - E_0\right)^{2P}} = 0, \qquad (4.19)
$$

for any excited state l such that $E_l > E_0$. In the limit of large P we therefore have that

$$
\frac{|\langle E_0|u_P\rangle|^2}{\langle u_P|u_P\rangle} \simeq 1 - \frac{\left(\Lambda - E_1\right)^{2P}}{\left(\Lambda - E_0\right)^{2P}}\frac{|c_1|^2}{|c_0|^2}, \qquad (4.20)
$$

and the correction can be made arbitrarily (and exponentially) small by increasing the number of steps P. We also see that for the exponential convergence to be true we need to have that the initial state has some finite overlap with the exact ground state, namely $|c_0|^2 \neq 0$. This can be achieved, for example, starting the iterations from a random vector.

The power method is therefore a very simple, yet exponentially converging method, to find the ground state of the Hamiltonian. If, for example, the Hamiltonian is stored in memory as a sparse matrix, then by simple iterative applications one can find the ground state. In practice, it is convenient to keep the state $|u_k\rangle$ normalized at each step, to avoid an exponential increase of the coefficients appearing in the vector $|u_k\rangle$.

4.2.2 The Lanczos Method

The Lanczos algorithm is an important improvement over the power method, that allows to reconstruct not only the ground state wave function, but also excited states. The Lanczos algorithm builds an orthogonal basis $\{v_1, v_2, \ldots, v_P\}$ for the Krylov-subspace

$K_P = \mathrm{span}\{u_1, u_2, \dots, u_P\}$, which is constructed by P iterations of the power method. This is achieved by the following iterations:

$$\beta_{n+1}|v_{n+1}\rangle = \hat{H}|v_n\rangle - \alpha_n|v_n\rangle - \beta_n|v_{n-1}\rangle, \tag{4.21}$$

where

$$\alpha_n = \langle v_n|\hat{H}|v_n\rangle, \qquad \beta_n = |\langle v_n|\hat{H}|v_{n-1}\rangle|. \tag{4.22}$$

Since the orthogonality condition

$$\langle v_i|v_j\rangle = \delta_{ij} \tag{4.23}$$

does not determine the phases of the basis vectors, the β_i can be chosen to be real and positive. It can be shown that we only need to keep three vectors of size M in memory, which makes the Lanczos algorithm very efficient, when compared to dense matrix eigen-solvers which require storage of order M^2 (see Table 4.1 for a summary of the complexity of matrix operations).

In the Krylov basis the matrix $\hat{H}$ is approximated by the following tridiagonal matrix

$$\hat{T}^{(n)} \doteq \begin{bmatrix} \alpha_1 & \beta_2 & 0 & \cdots & 0 \\ \beta_2 & \alpha_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \beta_n \\ 0 & \cdots & 0 & \beta_n & \alpha_n \end{bmatrix}, \tag{4.24}$$

and it can also been shown that the eigenvalues $\{\tau_1, \dots, \tau_M\}$ of $\hat{T}$ are good approximations of the eigenvalues of $\hat{H}$. Moreover, the extreme eigenvalues converge very fast. Thus $P \ll M$ iterations are sufficient to obtain the extreme eigenvalues. Since the Lanczos matrix is tridiagonal, we can use all the efficient computational approaches discussed in the previous Chapter to find both its eigenvalues and eigenvectors.

In practice, the Lanczos method can be already found implemented in all linear algebra solvers for sparse matrices, for example in scipy. For Python users, we strongly suggest to use SciPy (in particular scipy.sparse.linalg) which performs Lanczos/Arnoldi calling an efficient, C-coded backend. These routines allow to diagonalize directly sparse matrices defined within scipy. Alternatively, and in order to avoid storing the sparse matrix, one can also define its own Matrix-Vector multiplication using scipy.sparse.linalg.LinearOperator, and then obtain the eigenvalues and eigenvectors with a call to scipy.sparse.linalg.eigsh.

A more detailed discussion of the Lanczos algorithm and the issue of ghost eigenvalues can be found in Appendix 4.5.

4.2.3 Implementation

From the practical implementation point of view, the main requirement to use the simple power method or the more refined Lanczos algorithm is to provide a function that computes the product of the Hamiltonian with an arbitrary vector $|v\rangle$:

$$\hat{H}|v\rangle \;=\; |v'\rangle. \tag{4.25}$$

There are two main approaches to implement this efficiently. One one hand, we can form and store the hamiltonian $\hat{H}$ as a sparse matrix. This approach is very elegant and can be readily implemented, for example, in Python with scipy. The only requirement, for spin hamiltonians, is to explicitly use and form the Kronecker products for spin operators, as seen before:

$$\hat{\sigma}_i^{\alpha} \;=\; \hat{I}(2^{i-1}) \otimes \hat{\sigma}^{\alpha} \otimes \hat{I}(2^{N-i}), \tag{4.26}$$

and then construct interactions terms as simple products of these matrices. For example, spin-spin interaction terms $\hat{\sigma}_i^{\alpha}\hat{\sigma}_j^{\beta}$ can be readily obtained as a sparse matrix-matrix multiplication.

In addition of being very elegant and compactly implemented, this approach has also the advantage that computing products of sparse matrices with vectors is a typically highly optimized operation in dedicated software libraries, thus the resulting scheme will be automatically highly efficient. The main drawback however is the memory requirements, since we need to store all the non-zero matrix elements of the Hamiltonian, and there are at least as many as $N \times 2^N$ of them, as we have seen before. This memory requirement is added to the requirements due to the necessity of storing (at least) the vectors $|v\rangle$ and $|v'\rangle$, yielding an additional 2×2^N coefficients to be stored.

The main alternative approach is to never store the matrix $\hat{H}$ and provide instead a function that computes the matrix-vector product "on the fly". This allows to drastically reduce the memory consumption to the bare minimum, namely to 2×2^N, at the expenses of, typically, a larger computational time. The latter approach is especially suited for applications where reaching to the largest possible value of N is crucial, and need specialized low-level implementations. In the exercises we will mostly focus on the first approach.

4.3 Quantum Dynamics

In the previous discussion we have seen how to explicitly construct sparse representations of the Hamiltonian of quantum spin systems, and how to use them to obtain the ground-state wave-function. We now focus on the problem of solving the time-dependent Schrödinger equation for the many-spin system, a task which requires specific techniques. For simplicity, we will analyze here the specific case of time-independent Hamiltonians, and leave the straightforward extension to time-dependent Hamiltonians as an exercise.

4.3.1 Taylor Expansion

To implement the time evolution we have to devise an efficient way to numerically compute the matrix exponential $\exp(-i\hat{H}t)$, since for a static Hamiltonian the time-evolved state that satisfies Schrödinger equation reads :

$$|\Psi(t)\rangle = e^{-i\hat{H}t} |\Psi(0)\rangle \,. \tag{4.27}$$

The most straightforward way to do so is to take a small time step Δ_t and consider a truncated Taylor expansion of the exponential, such that

$$|\Psi(t+\Delta_t)\rangle = \left(1 - i\Delta_t \hat{H} - \frac{\Delta_t^2}{2}\hat{H}^2 - i\frac{\Delta_t^3}{6}\hat{H}^3 + \dots\right)|\Psi(t)\rangle\,. \tag{4.28}$$

Taking the first s orders in the Taylor expansion guarantees a scheme locally of order $\mathcal{O}(\Delta_t^s)$. This scheme can be efficiently implemented recalling that the Hamiltonian $\hat{H}$ is sparse, and that we can efficiently compute products of $\hat{H}$ with a given vector:

$$|\Psi'\rangle = \hat{H}\,|\Psi\rangle\,. \tag{4.29}$$

A simple iterative scheme that realizes the Taylor expansion numerically is given by the following recursion formula:

$$\begin{aligned} |\Gamma_k\rangle &= \frac{-i\Delta_t}{k}\hat{H}\,|\Gamma_{k-1}\rangle && (4.30)\\ |\Delta_k\rangle &= |\Delta_{k-1}\rangle + |\Gamma_k\rangle\,, && (4.31)\end{aligned}$$

for $k = 1, 2, \dots s$, up to the maximum truncation order chosen, and with zero-order conditions $|\Gamma_0\rangle = |\Delta_0\rangle = |\Psi(t)\rangle$. Then we have

$$|\Psi(t+\Delta_t)\rangle = |\Delta_s\rangle. \tag{4.32}$$

This scheme is particularly memory friendly, because it needs to store at most two vectors: $|\Delta_k\rangle$ and $|\Gamma_k\rangle$.

4.4 The Trotter-Suzuki decomposition

We now present an alternative numerical scheme which, at variance with the previous Taylor series, explicitly preserves the unitary character of the Hamiltonian evolution. To derive this scheme, we introduce one of the most important tools in computational quantum physics: the Trotter-Suzuki decomposition.[1]

To do this we split the Hamiltonian into a sum of K non-commuting terms $\hat{H} = \sum_{k=1}^{K}\hat{h}_k$. The splitting is done in such a way that the exponential of the individual terms, $e^{-i\hat{h}_k\Delta_t}$, can be easily computed. The time evolution operator $\exp(-i\hat{H}\Delta_t)$ for a small time step Δ_t is then decomposed into multiple products of the non-commuting terms in the Hamiltonian. To first order, the Trotter-Suzuki decomposition for a small time step Δ_t is

$$\exp(-i\hat{H}\Delta_t) = e^{-i\hat{h}_1\Delta_t}\dots e^{-i\hat{h}_K\Delta_t} + \mathcal{O}(\Delta_t^2). \tag{4.33}$$

The second order version of this formula reads

$$\exp(-i\hat{H}\Delta_t) = e^{-i\hat{h}_1\frac{\Delta_t}{2}}\dots e^{-i\hat{h}_K\frac{\Delta_t}{2}}e^{-i\hat{h}_K\frac{\Delta_t}{2}}\dots e^{-i\hat{h}_1\frac{\Delta_t}{2}} + \mathcal{O}(\Delta_t^3). \tag{4.34}$$

For the special case with $K = 2$ terms this expression simplifies to

$$\exp(-i\hat{H}\Delta_t) = e^{-i\hat{h}_1\Delta_t/2}e^{-i\hat{h}_2\Delta_t}e^{-i\hat{h}_1\Delta_t/2} \tag{4.35}$$

[1] H. F. Trotter, *On the product of semi-groups of operators*, Proc. Amer. Math. Soc. **10**, 545 (1959); M. Suzuki, *Generalized Trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems*, Commun. Math. Phys. **51**, 183 (1976).

by combining the two terms $e^{-i\hat{h}_2\Delta_t/2}e^{-i\hat{h}_2\Delta_t/2}$ into $e^{-i\hat{h}_2\Delta_t}$. By similarly combining the terms $e^{-i\hat{h}_1\Delta_t/2}e^{-i\hat{h}_1\Delta_t/2}$ arising from two adjacent time steps into $e^{-i\hat{h}_1\Delta_t}$ one ultimately needs only one single additional terms for the full time evolution, when compared to the first order approximation. At second order, the full time evolution for $K = 2$ indeed reads

$$\begin{aligned} \exp(-i\hat{H}t) &\simeq e^{-i\hat{h}_1\Delta_t/2}e^{-i\hat{h}_2\Delta_t}e^{-i\hat{h}_1\Delta_t/2} \times e^{-i\hat{h}_1\Delta_t/2}e^{-i\hat{h}_2\Delta_t}e^{-i\hat{h}_1\Delta_t/2}\ldots \quad (4.36)\\ &\simeq e^{-i\hat{h}_1\Delta_t/2}\ldots e^{-i\hat{h}_2\Delta_t}e^{-i\hat{h}_1\Delta_t}\ldots e^{-i\hat{h}_2\Delta_t}e^{-i\hat{h}_1\Delta_t/2}. \quad (4.37) \end{aligned}$$

4.4.1 Time evolution for the transverse field Ising model

To implement time evolution in the transverse field Ising model we split the Hamiltonian into $K = 2$ non-commuting terms. The first one is the the transverse field term

$$\hat{H}_x = -\Gamma \sum_l \hat{\sigma}_l^x, \quad (4.38)$$

and the second one is the Ising term

$$\hat{H}_z = \sum_{\langle l,m \rangle} J_{lm}\hat{\sigma}_l^z\hat{\sigma}_m^z. \quad (4.39)$$

We will now see that each of these terms can be easily exponentiated.

The transverse field term splits into N commuting terms for each of the spins:

$$e^{-i\hat{H}_x\Delta_t} = e^{i\Delta_t\Gamma\sum_l \hat{\sigma}_l^x} = \prod_l e^{i\Delta_t\Gamma\hat{\sigma}_l^x}. \quad (4.40)$$

Each of the terms in the product above can be written explicitly in Kronecker product form:

$$e^{i\Delta_t\Gamma\hat{\sigma}_l^x} = \hat{I}(2^{l-1}) \otimes \left(\cos(\Delta_t\Gamma)\hat{I}(2) + i\sin(\Delta_t\Gamma)\hat{\sigma}^x\right) \otimes \hat{I}(2^{N-l}). \quad (4.41)$$

The Ising term instead is diagonal, and the exponentiation is particularly simple, yielding a diagonal matrix:

$$e^{-i\hat{H}_z\Delta_t}|s_1s_2\ldots s_N\rangle = \prod_{\langle l,m \rangle} e^{-i\Delta_t J_{lm}s_ls_m}|s_1s_2\ldots s_N\rangle. \quad (4.42)$$

We can further write this as a sum of Kronecker products, noticing that (the proof is left as an exercise)

$$e^{-i\theta\hat{\sigma}_l^z\hat{\sigma}_m^z} = \cos\theta\hat{I}(2^N) - i\sin\theta\hat{\sigma}_l^z\hat{\sigma}_m^z, \quad (4.43)$$

thus each term in the product $\prod_{\langle l,m \rangle} e^{-i\Delta_t J_{lm}s_ls_m}$ can be easily applied recalling the explicit Kronecker product form of $\hat{\sigma}_l^z$ and $\hat{\sigma}_m^z$.

Overall, then both the diagonal and the off diagonal terms can easily be applied to a wave function in a similar way as we did for the multiplication with the Hamiltonian $\hat{H}$. We will implement time evolution for the TFIM in the exercises.

Table 4.1: Time and memory complexity for operations on sparse and dense $M \times M$ matrices

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

4.5 Appendix: The Lanczos algorithm

Sparse matrices with only $\mathcal{O}(M)$ non-zero elements are very common in scientific simulations. We have seen in this Chapter that many-body quantum Hamiltonians belong to the class of sparse matrices, and that for typical spin models one has $M \sim 2^N N^\alpha$, for some small power α.

The importance of sparsity becomes obvious when considering the cost of matrix operations as listed in table 4.1. For large M 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.

4.5.1 Finding eigenvectors

While finding the eigenvectors of the tridiagonal Lanczos matrix $\hat{T}$ is a relatively easy computational task, however these are not directly the eigenvectors of the original matrix $\hat{H}$, since they are given in the (much smaller) Krylov basis $\{v_1, v_2, \ldots, v_P\}$. 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.

4.5.2 Roundoff errors and ghosts in the Lanczos algorithm

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

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

- Re-orthogonalization 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 $\hat{T}$ contains extra spurious eigenvalues, called "ghosts". These ghosts are not real eigenvalues of $\hat{A}$. However they converge towards real eigenvalues of $\hat{A}$ 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 $\hat{Q}^{(n)}$, which can be obtained from $\hat{T}$ by deleting the first row and column:

$$\hat{Q}^{(n)} := \begin{bmatrix} \alpha_2 & \beta_3 & 0 & \cdots & 0 \\ \beta_3 & \alpha_3 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \beta_n \\ 0 & \cdots & 0 & \beta_n & \alpha_n \end{bmatrix}. \tag{4.44}$$

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.

4.5.3 Open-source implementations

Numerically stable and efficient implementations of the Lanczos algorithm can be obtained as part of open-source packages.

For Python users, we strongly suggest to use SciPy (in particular scipy.sparse.linalg) which performs Lanczos/Arnoldi calling an efficient, C-coded backend. These routines allow to diagonalize directly sparse matrices defined within scipy. Alternatively, and in order to avoid storing the sparse matrix, one can also define its own Matrix-Vector multiplication using scipy.sparse.linalg.LinearOperator, and then obtain the eigenvalues and eigenvectors with a call to scipy.sparse.linalg.eigsh.

For C++ users, we strongly suggest the use of the EIGEN library[2] in conjunction with SPECTRA[3]. Both libraries are header-only, require almost no installation effort

[2] http://eigen.tuxfamily.org/

[3] http://yixuan.cos.name/spectra/

(apart from downloading it), and are very efficient. SPECTRA handles the Lanczos/Arnoldi algorithm, and just needs the user to implement a function implementing the Matrix-Vector multiplication, with minor modifications with respect to the one previously discussed in the Lecture.

Chapter 5

Indistinguishable particles: fermions and bosons

5.1 Introduction

We have seen at the beginning of this course that the complete quanto-mechanical description of a single particle is given by its wave function $\langle \vec{q}\,|\psi\rangle \equiv \psi(\vec{q})$, where $\vec{q}$ is a chosen coordinate (say the position or the momentum of the particle, for example). Single-particle wave functions live in the function space L^2, which, among others, ensures that wave-functions are normalizable. Without loss of generality, we can pick from this space a set of orthonormal functions $\phi_a(\vec{q})$, satisfying

$$\begin{aligned} \int d\vec{q}\phi_a^\star(\vec{q})\phi_b(\vec{q}) &= \delta_{ab} && (5.1) \\ \sum_a \phi_a^\star(\vec{q})\phi_a(\vec{q}\,') &= \delta(\vec{q}-\vec{q}\,'), && (5.2) \end{aligned}$$

and such that any $\psi(\vec{q})$ can be written as a linear combination of those basis functions.

What happens now if we want to describe a many-particle system? We have seen in the case of many-spins systems that we should start by considering the tensor product of the individual Hilbert spaces. For example, a two-particle system would live in the space $L^2 \otimes L^2$, and its wave function

$$\psi(\vec{q}_1, \vec{q}_2), \tag{5.3}$$

can be written as a linear combination of an (infinite) set of basis functions of the form

$$\begin{aligned} \phi_1^{(2)}(\vec{q}_1, \vec{q}_2) &= \phi_1(\vec{q}_1)\phi_1(\vec{q}_2) && (5.4) \\ \phi_2^{(2)}(\vec{q}_1, \vec{q}_2) &= \phi_1(\vec{q}_1)\phi_2(\vec{q}_2) \\ \phi_3^{(2)}(\vec{q}_1, \vec{q}_2) &= \phi_2(\vec{q}_1)\phi_1(\vec{q}_2) \\ \ldots &= \ldots . \end{aligned}$$

This basis set, albeit completely acceptable, is not the most practical one to describe physical systems of many-particles. The reason is that in most situations we deal with an *indistinguishable* set of particles[1]. For example, electrons or photons are indistinguishable: there is no serial number painted on the electrons that would allow us to distinguish two electrons. As a consequence, the only physical basis states are only those that have the correct particle-exchange symmetries. If we exchange the label of two particles, the system must be the same as before.

For our two-body wave-function example, this means that

$$\psi(\vec{q}_2, \vec{q}_1) = e^{i\phi}\psi(\vec{q}_1, \vec{q}_2), \tag{5.5}$$

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[2]

$$\psi(\vec{q}_1, \vec{q}_2) = e^{i\phi}\psi(\vec{q}_2, \vec{q}_1) = e^{2i\phi}\psi(\vec{q}_1, \vec{q}_2), \tag{5.6}$$

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

$$\psi(\vec{q}_2, \vec{q}_1) = \pm\psi(\vec{q}_1, \vec{q}_2) \tag{5.7}$$

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.

One of the major consequences of the exchange symmetry is that no two fermions can be in the same state as a wave function. Consider for example the state

$$\phi_1^{(2)}(\vec{q}_1, \vec{q}_2) = \phi_1(\vec{q}_1)\phi_1(\vec{q}_2), \tag{5.8}$$

which is completely symmetric with respect to particle exchange $\vec{q}_1 \leftrightarrow \vec{q}_2$, therefore any attempt to anti-symmetrize leads to a vanishing state:

$$\text{Antisymm}[\phi_1^{(2)}(\vec{q}_1, \vec{q}_2)] = \phi_1(\vec{q}_1)\phi_1(\vec{q}_2) - \phi_1(\vec{q}_2)\phi_1(\vec{q}_1) = 0. \tag{5.9}$$

This is known as Pauli principle.

5.2 The Fock space

Generic elements in the tensor-product space spanned by the functions in (5.4) do not satisfy the physical requirements of exchange symmetry. For example, the state

$$\phi_2^{(2)}(\vec{q}_1, \vec{q}_2) \quad \neq \quad \pm\phi_2^{(2)}(\vec{q}_2, \vec{q}_1), \tag{5.10}$$

and it is not physically allowed.

[1] There are fundamental reasons why classical and quantum particles are taken to be indistinguishable. One of the most striking arguments comes from the Gibbs paradox in thermodynamics: the entropy of an ideal gas would not be extensive if particles were distinguishable.

[2] As a side remark we want to mention that in two dimensions the first homotopy group is $\mathbb{Z}$ and nontrivial: 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.

It is therefore more convenient to work in a space where symmetrization properties are taken into account from the very beginning. The Hilbert space describing a quantum many-body system with $N = 0, 1, \ldots, \infty$ identical particles is called the Fock space. It is the direct sum of the appropriately symmetrized N-particle Hilbert spaces (made up from single-particle Hilbert spaces) $\mathcal{H}$:

$$\bigoplus_{N=0}^{\infty} S_{\pm}\mathcal{H}^{\otimes N} \tag{5.11}$$

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

5.2.1 Fermions

For Fermions the basis wave functions have to be antisymmetric under exchange of any two particles. If we have N fermions and L states, a basis state for the Fock space is fully specified by the ket of occupation numbers $|n_1 \ldots n_L\rangle$ of the single-particle states. Since no two fermions cannot occupy the same state the occupation numbers are restricted to $n_i = 0$ or 1. Moreover, since the total number of particles is conserved we must have $\sum_i^L n_i = N$.

The wave function of the state $|n_1, \ldots, n_L\rangle$ is given by the appropriately anti-symmetrized and normalized product of the single particle wave functions.

For example, the two-particle basis state $|1, 1\rangle$ has wave function

$$\langle \vec{q}_1, \vec{q}_2 | 1, 1\rangle = \frac{1}{\sqrt{2}} \left[\phi_1(\vec{q}_1)\phi_2(\vec{q}_2) - \phi_1(\vec{q}_2)\phi_2(\vec{q}_1)\right]. \tag{5.12}$$

In general, a Fermionic basis state in Fock space takes the form of a Slater determinant

$$\langle \vec{q}_1, \ldots \vec{q}_N | n_1, \ldots n_L\rangle \quad = \quad \frac{1}{\sqrt{N!}} \begin{vmatrix} \phi_{a_1}(\vec{q}_1) & \ldots & \phi_{a_1}(\vec{q}_N) \\ \phi_{a_2}(\vec{q}_1) & \ldots & \phi_{a_2}(\vec{q}_N) \\ \vdots & \vdots\vdots\vdots & \vdots \\ \phi_{a_N}(\vec{q}_1) & \ldots & \phi_{a_N}(\vec{q}_N) \end{vmatrix}, \tag{5.13}$$

where the occupied single-particle states (also called *orbitals*) are $1 \le a_1 < a_2 < \cdots < a_N \le L$, such that $n_{a_i} \neq 0$. The states $|n_1, \ldots n_L\rangle$ are trivially anti-symmetric under exchange of two particles, because of the property of the determinant (i.e., exchanging two columns of the matrix leads to a factor -1).

5.2.2 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 $\vec{q} = (\vec{x}, \sigma)$ including both position $\vec{x}$ and spin σ.

5.2.3 Bosons

For Bosons instead a general basis function in Fock space needs to be symmetric under permutations. If we have N Bosons and L states, a basis state for the Fock space is again fully specified by the ket of occupation numbers $|n_1 \dots n_L\rangle$ of the single-particle states. At variance with Fermions, however, the occupation numbers can take any arbitrary value $n_i = 0, \dots N$, provided that the total number of particles is conserved, i.e. $\sum_i^L n_i = N$.

The wave function of the state $|n_1, \dots, n_L\rangle$ is given by the appropriately symmetrized and normalized product of the single particle wave functions.

For example, the two-particle basis state $|1, 1\rangle$ has wave function

$$\langle \vec{q}_1, \vec{q}_2 | 1, 1\rangle = \frac{1}{\sqrt{2}} \left[\phi_1(\vec{q}_1)\phi_2(\vec{q}_2) + \phi_1(\vec{q}_2)\phi_2(\vec{q}_1) \right]. \tag{5.14}$$

In general, a Fermionic basis state in Fock space takes the form of a matrix permanent

$$\langle \vec{q}_1, \dots \vec{q}_N | n_1, \dots n_L \rangle \;=\; \sqrt{\frac{\prod_j n_j}{N!}} \mathrm{perm} \begin{pmatrix} \phi_{a_1}(\vec{q}_1) & \dots & \phi_{a_1}(\vec{q}_N) \\ \phi_{a_2}(\vec{q}_1) & \dots & \phi_{a_2}(\vec{q}_N) \\ \vdots & ::: & \vdots \\ \phi_{a_N}(\vec{q}_1) & \dots & \phi_{a_N}(\vec{q}_N) \end{pmatrix}, \tag{5.15}$$

where the occupied single-particle states are1 $\leq a_1 \leq a_2 \leq \dots \leq a_N \leq L$, such that $n_{a_i} \neq 0$ and the same state can be occupied by more than a particle.

5.3 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 which directly operate in Fock space.

5.3.1 Fermionic operators

Let us start introducing the creation, $\hat{c}_i{}^\dagger$, and annihilation, $\hat{c}_i$, operators which add or remove, respectively, a fermion in state i.

The hermitian operator $\hat{c}_i{}^\dagger \hat{c}_i$ first annihilates then creates a particle in state i, and, analogously to the harmonic oscillator case we have already seen, it counts how many fermions occupy a given state, i.e.

$$\hat{c}_i{}^\dagger \hat{c}_i |n_i\rangle \;=\; n_i |n_i\rangle. \tag{5.16}$$

We therefore have

$$\hat{c}_i{}^\dagger \hat{c}_i |n_i = 0\rangle \;=\; 0 \tag{5.17}$$

$$\hat{c}_i{}^\dagger \hat{c}_i |n_i = 1\rangle \;=\; |1\rangle. \tag{5.18}$$

On the other hand, the operator $\hat{c}_i\hat{c}_i{}^\dagger$ first creates than destroys a particle in state i, and we have

$$\begin{aligned} \hat{c}_i\hat{c}_i{}^\dagger|n_i = 0\rangle &= |n_i = 0\rangle && (5.19)\\ \hat{c}_i\hat{c}_i{}^\dagger|n_i = 1\rangle &= 0, && (5.20) \end{aligned}$$

where the last relation comes from the Pauli principle. Summing the two cases, we obtain

$$(\hat{c}_i\hat{c}_i{}^\dagger + \hat{c}_i{}^\dagger\hat{c}_i)|n_i = 0,1\rangle = |n_i = 0,1\rangle, \qquad (5.21)$$

which leads to the identity

$$\{\hat{c}_i, \hat{c}_i{}^\dagger\} = 1, \qquad (5.22)$$

where $\{\hat{a}, \hat{b}\} \equiv \hat{a}\hat{b} + \hat{b}\hat{a}$, denotes the anti-commutator. We have thus found that the fermionic operators anti-commute. In addition, since we cannot create nor destroy two fermions in the same state, we have

$$\begin{aligned} \{\hat{c}_i, \hat{c}_i\} &= 0 && (5.23)\\ \left\{\hat{c}_i{}^\dagger, \hat{c}_i{}^\dagger\right\} &= 0. && (5.24) \end{aligned}$$

When acting on a Fock state, say of two fermions, we have

$$\begin{aligned} \hat{c}_i{}^\dagger c_j^\dagger|0\rangle &= |n_i = 1, n_j = 1\rangle && (5.25)\\ c_j^\dagger\hat{c}_i{}^\dagger|0\rangle &= |n_j = 1, n_i = 1\rangle\,, && (5.26) \end{aligned}$$

however we know that if we exchange labels to the orbitals in the Slater determinant the state picks a minus sign, therefore

$$|n_j = 1, n_i = 1\rangle = -\,|n_i = 1, n_j = 1\rangle\,, \qquad (5.27)$$

and we can conclude that it must be

$$\hat{c}_i{}^\dagger\hat{c}_j{}^\dagger = -\hat{c}_j{}^\dagger\hat{c}_i{}^\dagger, \qquad (5.28)$$

or equivalently

$$\{\hat{c}_i{}^\dagger, \hat{c}_j{}^\dagger\} = 0. \qquad (5.29)$$

Moreover,

$$\begin{aligned} \hat{c}_i\hat{c}_j{}^\dagger|n_i = 1\rangle &= \hat{c}_i|n_j = 1, n_i = 1\rangle && (5.30)\\ &= -\hat{c}_i|n_i = 1, n_j = 1\rangle && (5.31)\\ &= -|n_j = 1\rangle, && (5.32) \end{aligned}$$

and

$$\begin{aligned} \hat{c}_j{}^\dagger\hat{c}_i|n_i = 1\rangle &= \hat{c}_j{}^\dagger|0\rangle && (5.33)\\ &= |n_j = 1\rangle, && (5.34) \end{aligned}$$

which again summing up the two equations leads to

$$\{\hat{c}_i, \hat{c}_j{}^\dagger\} \quad = \quad 0. \tag{5.35}$$

All these commutation relations can be recast in the compact form

$$\begin{aligned} \{\hat{c}_i, \hat{c}_j{}^\dagger\} &= \delta_{i,j}, & (5.36)\\ \{\hat{c}_i, \hat{c}_j\} &= 0, & (5.37)\\ \{\hat{c}_i{}^\dagger, \hat{c}_j{}^\dagger\} &= 0. & (5.38) \end{aligned}$$

5.3.1.1 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:

$$|n_1, \ldots, n_L\rangle \quad = \quad \prod_{i=1}^{L} (\hat{c}_i{}^\dagger)^{n_i}|0\rangle = (\hat{c}_1{}^\dagger)^{n_1}(\hat{c}_2{}^\dagger)^{n_2}\cdots(\hat{c_L}{}^\dagger)^{n_L}|0\rangle \tag{5.39}$$

The order in which we apply the creation operators is extremely important, since the fermionic creation operators anti-commute and, for example, $\hat{c}_1{}^\dagger\hat{c}_2{}^\dagger|0\rangle = -\hat{c}_2{}^\dagger\hat{c}_1{}^\dagger|0\rangle$. We thus need to agree on a specific ordering of the creation operators to define what we mean by the state $|n_1, \ldots, n_L\rangle$. The choice of ordering does not matter but we have to stay consistent and use e.g. the convention in equation (5.39).

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

$$\begin{aligned} \hat{c}_i|n_1, \ldots, n_i, \ldots, n_L\rangle &= \delta_{n_i,1}(-1)^{\sum_{j=1}^{i-1} n_j}|n_1, \ldots, n_i - 1, \ldots, n_L\rangle & (5.40)\\ \hat{c}_i{}^\dagger|n_1, \ldots, n_i, \ldots, n_L\rangle &= \delta_{n_i,0}(-1)^{\sum_{j=1}^{i-1} n_j}|n_1, \ldots, n_i + 1, \ldots, n_L\rangle, & (5.41) \end{aligned}$$

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

5.3.2 Bosonic operators

The same procedure can be carried on for Bosons, introducing creation $\hat{b}_i^\dagger$ and annihilation operators $\hat{b}_i$. The most notable difference is that in this case the Pauli principle does not hold, and different commutation relations are found. In particular, one can show that

$$\begin{aligned} [\hat{b}_i, \hat{b}_j^\dagger] &= \delta_{ij}, & (5.42)\\ [\hat{b}_i, \hat{b}_j] &= 0, & (5.43)\\ [\hat{b}_i^\dagger, \hat{b}_j^\dagger] &= 0, & (5.44) \end{aligned}$$

and that the bosonic operators act on Fock states as

$$\begin{aligned} \hat{b}_i|n_1, \ldots n_i, \ldots . n_L\rangle &= \sqrt{n_i}|n_1, \ldots n_i - 1, \ldots . n_L\rangle, & (5.45)\\ \hat{b}_i^\dagger|n_1, \ldots n_i, \ldots . n_L\rangle &= \sqrt{n_i + 1}|n_1, \ldots n_i + 1, \ldots . n_L\rangle. & (5.46) \end{aligned}$$

Notice that at variance with fermions in the previous expression we have extra factors to take into account, to guarantee the normalization of the Fock states.

A generic Fock state can be then written as

$$|n_1, \ldots, n_L\rangle = \prod_{i=1}^{L} \frac{(\hat{b}_i^\dagger)^{n_i}}{\sqrt{n_i!}} |0\rangle = \frac{(\hat{b}_1^\dagger)^{n_1}}{\sqrt{n_1!}} \frac{(\hat{b}_2^\dagger)^{n_2}}{\sqrt{n_2!}} \cdots \frac{(\hat{b}_L^\dagger)^{n_L}}{\sqrt{n_L!}} |0\rangle \qquad , \tag{5.47}$$

and since creation operators on different states commute, in this case we do not need to stick to any specific normal ordering.

5.4 Exact diagonalization

Exact diagonalization for identical particles is very similar to what we have already done for spins. In the following we will see specific examples for fermions, which however already contains all the necessary ingredients.

5.4.1 Bosons

Bosons are conceptually identical to spins, in the sense that in practical applications we can think of truncating the local Hilbert space and set up a maximum occupation number, d such that $0 \leq n_i < d$. Then, in this representation the bosonic operators have a very simple explicit form, for example:

$$\hat{b}_i = \underbrace{\hat{I}(d) \otimes \ldots \hat{I}(d)}_{i-1 \text{ terms}} \otimes \hat{b} \otimes \underbrace{\hat{I}(d) \otimes \ldots \hat{I}(d)}_{L-i \text{ terms}}, \tag{5.48}$$

with, for example, the destruction operator a $d \times d$ matrix:

$$\hat{b} = \begin{pmatrix} 0 & \sqrt{1} & 0 & \cdots & 0 \\ 0 & 0 & \sqrt{2} & \cdots & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & \sqrt{d} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{5.49}$$

and $\hat{b}_i^\dagger$ is instead just the transpose of $\hat{b}_i$.

With this explicit writing of bosonic operators in terms of Kronecker products, we can then write any bosonic Hamiltonian using the same strategy adopted for spins in the previous Chapter.

5.4.2 Fermions

For Fermions, the situation is slightly more complicated, because we have to take into account the normal ordering of the operators, as discussed previously. However, Jordan and Wigner realized that fermionic operators can be mapped onto spin operators, with a relatively simple trick. The main idea of these mappings is to identify the local occupation numbers $n_i = (0,1)$ with the local spin numbers $s_i = (1,-1)$ in such a way that the number operator maps onto the $\hat{\sigma}^z$ operator:

$$\hat{n}_i |n_1, \ldots, n_i, \ldots, n_L\rangle \rightarrow \frac{\left(\hat{I} - \hat{\sigma}_i^z\right)}{2} |s_1, \ldots, s_i, \ldots, s_L\rangle, \tag{5.50}$$

then we have that

$$\begin{aligned} \hat{c}_i |n_1, \ldots, n_i, \ldots, n_L\rangle &= \delta_{n_i,1} (-1)^{n_1} \ldots (-1)^{n_{i-1}} |n_1, \ldots, n_i - 1, \ldots, n_L\rangle & (5.51) \\ &\rightarrow \underbrace{\hat{\sigma}^z \otimes \ldots \hat{\sigma}^z}_{i-1 \text{ terms}} \otimes \hat{\sigma}^+ \otimes \underbrace{\hat{I} \otimes \cdots \otimes \hat{I}}_{L-i \text{ terms}} |s_1, \ldots, s_i, \ldots, s_L\rangle & (5.52) \end{aligned}$$

where we have introduced the usual raising and lowering spin operators:

$$\hat{\sigma}^{\pm} = \frac{\hat{\sigma}^x \pm i\hat{\sigma}^y}{2}. \tag{5.53}$$

With the prescription

$$\hat{c}_i = \underbrace{\hat{\sigma}^z \otimes \hat{\sigma}^z \ldots \hat{\sigma}^z}_{i-1 \text{ terms}} \otimes \hat{\sigma}^+ \otimes \underbrace{\hat{I} \otimes \hat{I} \cdots \otimes \hat{I}}_{L-i \text{ terms}}, \tag{5.54}$$

and analogously for its conjugate operator $\hat{c}_i^\dagger$, we can therefore write an arbitrary fermionic Hamiltonian just using Kronecker products of spin operators, and again we can use all the techniques for sparse matrices described in the previous Chapter.

The mapping derived above is for spinless fermions, however we can also readily generalize the discussion to the case when spin degrees of freedom are present, most notably for operators obeying the commutation relations

$$\{\hat{c}_{i,\sigma}^\dagger, \hat{c}_{j,\sigma'}\} = \delta_{i,j}\delta_{\sigma,\sigma'}. \tag{5.55}$$

We can readily use the Kronecker product expressions found for the spinless case just increasing the total number of local states:

$$n_{i,\sigma} \rightarrow n'_k, \tag{5.56}$$

with $k \in [1, L \times d]$, with the d the dimension of the local spin space. For example in the case of spin 1/2 fermions we have $\sigma =\uparrow,\downarrow$, $d = 2$ and we can work with the following occupation numbers as a basis:

$$|n_{1\uparrow}, n_{1\downarrow} \ldots, n_{L,\uparrow}, n_{L,\downarrow}\rangle \rightarrow |n'_1, n'_2 \ldots, n'_{2L-1}, n'_{2L}\rangle, \tag{5.57}$$

and

$$\hat{c}_{i,\uparrow} \rightarrow \hat{c}'_{2i} \tag{5.58}$$

$$\hat{c}_{i,\downarrow} \rightarrow \hat{c}'_{2i+1}. \tag{5.59}$$

5.4.3 Fixing the number of particles

In most cases of interest the Hamiltonians we work with commute with the total particle number operator,

$$\hat{N} = \sum_i \hat{n}_i, \tag{5.60}$$

this means that $\hat{H}$ is block-diagonal in the sectors with fixed number of particles. We can for example exploit this symmetry of the problem to reduce the dimensionality of the Hilbert space and diagonalize a smaller matrix. We will not describe in detail strategies to do that in this course. A simple strategy to diagonalize the Hamiltonian in a fixed symmetry sector consists instead in wisely choosing the initial states for our iterative diagonalization approaches. For example, in the power method we can choose an initial state $|u_0\rangle$ such that it is an eigenket of $\hat{N}$ with the desired number of particles. For example, we could take any simple product state:

$$|u_0\rangle \;\; = \;\; |n_1\rangle \otimes |n_2\rangle \otimes \ldots |n_L\rangle, \tag{5.61}$$

with random n_i and such that $\sum_i n_i = N$. Then, the power method will yield the lowest eigenstate of $\hat{H}$ non-orthogonal to $|u_0\rangle$ (remember that we asked the condition $|\langle E_0|u_0\rangle| \neq 0$). Since $|u_0\rangle$ however is orthogonal to all eigenstates of the Hamiltonian that have number of particles different from N, it follows that we will converge to the ground state with the given fixed number of particles.

5.5 Example Fermionic models

Here we give a few examples of model fermionic Hamiltonian. The simplest case is certainly the "tight binding" model for spinless fermions

$$\hat{H} = \sum_{i,j} t_{ij} \hat{c}_i{}^{\dagger} \hat{c}_j, \tag{5.62}$$

where $\hat{c}_i$ obey Fermi statistics and t_{ij} are some arbitrary coefficients describing transitions between state i and state j. Notice that for the Hamiltonian to be hermitian, we must have $t_{ij} = t^{\star}_{ji}$, which in particular implies that the diagonal term t_{ii} is real. This model is easily solvable by Fourier transforming it, as there are no interactions.

5.5.1 The Hubbard model

To include effects of electron correlations, the Hubbard model includes only the often dominant on-site repulsion:

$$H = \sum_{\langle i,j\rangle,\sigma} (t_{ij} \hat{c}^{\dagger}_{i,\sigma} \hat{c}_{j,\sigma} + \text{H.c.}) + \sum_i U_i \hat{n}_{i,\uparrow} \hat{n}_{i,\downarrow}. \tag{5.63}$$

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.

5.5.2 The t-J model

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

$$H = \sum_{\langle i,j\rangle,\sigma} \left[(1 - \hat{n}_{i,-\sigma}) t_{ij} \hat{c}^{\dagger}_{i,\sigma} \hat{c}_{j,\sigma} (1 - \hat{n}_{j,-\sigma}) + \text{H.c.}\right] + \sum_{\langle i,j\rangle} J_{ij} (\vec{S}_i \vec{S}_j - \hat{n}_i \hat{n}_j / 4). \tag{5.64}$$

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.

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 to predict almost all properties of matter at human scale. With $\mathrm{O}(10^{23})$ atoms in a typical piece of matter, the exponential scaling of the Hilbert space dimension with the number of particles is a nightmare, and the exact diagonalization schemes discussed previously cannot be applied. In this Chapter we will set aside for a moment exact solutions, and introduce instead approximate methods that reduce the problem to a polynomial one, typically scaling like $\mathcal{O}(N^4)$ and even $\mathrm{O}(N)$ 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. Being able to numerically implement approximate methods is however extremely important, since it is often the case that a fast, qualitatively (but not quantitatively) accurate prediction is desirable.

6.1.1 The electronic structure problem

For many atoms (with the notable exception of Hydrogen and Helium which are so light that quantum effects are important in daily life), the nuclei of atoms are so much heavier than the electrons that we can view them as classical particles and can consider them as stationary for the purpose of calculating the properties of the electrons. This assumption, that neglects the kinetic energy of the nuclei, is known as the Born-Oppenheimer approximation, leading to the following Hamiltonian operator for the electrons

$$\hat{H} = \sum_{i=1}^{N} \left(-\frac{\hbar^2}{2m} \nabla^2_{\mathbf{r}_i} + V_{\mathrm{en}}(\mathbf{r}_i) \right) + \sum_{i<j} \frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|} \tag{6.1}$$

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

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

In the following we will also adopt the following notation for the one-body and two-body parts entering the Hamiltonian:

$$\begin{aligned} \hat{v}_1(\mathbf{r}) &= -\frac{\hbar^2}{2m}\nabla^2_{\mathbf{r}} + V_{\text{en}}(\mathbf{r}), && (6.3)\\ \hat{v}_2(\mathbf{r},\mathbf{r}') &= \frac{e^2}{|\mathbf{r}-\mathbf{r}'|}. && (6.4) \end{aligned}$$

6.2 Hamiltonian in second quantization

We have seen in the previous lecture that the natural space for treating identical particles (like the electrons) is the Fock space. In order to operate efficiently in this space, we need to be able to write the Hamiltonian operator (6.1) in terms of creation and annihilation operators. Using a basis set of L orbital wave functions $\{\phi_i\}$, the matrix elements of the Hamiltonian are

$$\begin{aligned} t_{ij} &= \int d\mathbf{r}\phi_i^*(\mathbf{r})\hat{v}_1(\mathbf{r})\phi_j(\mathbf{r}) && (6.5)\\ V_{ijkl} &= \int d\mathbf{r} \int d\mathbf{r}'\phi_i^*(\mathbf{r})\phi_j^\star(\mathbf{r}')\hat{v}_2(\mathbf{r},\mathbf{r}')\phi_k(\mathbf{r})\phi_l(\mathbf{r}'). && (6.6) \end{aligned}$$

In second quantized notation, the Hamiltonian then reads

$$\hat{H} = \sum_{ij\sigma} t_{ij}\hat{c}^\dagger_{i\sigma}\hat{c}_{j\sigma} + \frac{1}{2}\sum_{ijkl\sigma\sigma'} V_{ijkl}\hat{c}^\dagger_{i\sigma}\hat{c}^\dagger_{j\sigma'}\hat{c}_{l\sigma'}\hat{c}_{k\sigma}. \tag{6.7}$$

Notice that, mostly as a matter of convention (we like to write interactions with a + sign in front) the order of the indices in the interaction terms (i, j, l, k) does not follow the same order of the indices of interaction matrix (i, j, k, l). The explicit derivation of the form (6.5) is straightforward yet rather tedious. We have omitted here, but the interested reader can find it in full detail in any many-body theory book.

6.3 Basis functions

Before attempting to solve the many body problem we will discuss suitable basis sets for single particle wave functions. In realistic applications, using the naive discretization of space introduced in the first lectures will not be an efficient solution, thus we need some physical or chemical intuition to guide us in the best choice for the basis sets. This is also why the basis functions are so tightly connected to the notion of *orbital* in chemistry, since a correct choice of the single particle basis directly affects (at least in simple molecules) the understanding of the electronic structure.

6.3.1 The electron gas

For the free electron gas ($V_{\text{en}} = 0$) with Hamilton operator

$$\hat{H} = -\sum_{i=1}^{N} \frac{\hbar^2}{2m}\nabla^2 + \sum_{i<j} \frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|} \tag{6.8}$$

one of the most commonly adopted basis functions are plane waves

$$\psi_{\mathbf{k}}(\mathbf{r}) = \exp(-i\mathbf{k}\cdot\mathbf{r}). \tag{6.9}$$

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):

$$f_{inlm}(r,\theta,\phi) \propto r^{n-1}e^{-\zeta_i r}Y_{lm}(\theta,\phi). \tag{6.10}$$

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 (6.6) for basis functions centered around two different nuclei at position $\mathbf{R}_A$ and $\mathbf{R}_B$. There we have to evaluate integrals containing terms like

$$\frac{1}{|\mathbf{r}-\mathbf{r}'|}e^{-\zeta_i|\mathbf{r}-\mathbf{R}_A|}e^{-\zeta_j|\mathbf{r}-\mathbf{R}_B|} \tag{6.11}$$

which cannot be solved in any closed form.

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

$$f_{ilmn}(\mathbf{r}) \propto x^l y^m z^n e^{-\zeta_i r^2} \tag{6.12}$$

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:

$$e^{-\zeta_i|\mathbf{r}-\mathbf{R}_A|^2}e^{-\zeta_j|\mathbf{r}-\mathbf{R}_B|^2} = Ke^{-\zeta|\mathbf{r}-\mathbf{R}|^2} \tag{6.13}$$

with

$$K = e^{-\frac{\zeta_i\zeta_j}{\zeta_i+\zeta_j}|\mathbf{R}_A-\mathbf{R}_B|^2} \tag{6.14}$$

$$\zeta = \zeta_i + \zeta_j \tag{6.15}$$

$$\mathbf{R} = \frac{\zeta_i\mathbf{R}_A + \zeta_j\mathbf{R}_B}{\zeta_i+\zeta_j} \tag{6.16}$$

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

$$\frac{1}{|\mathbf{r}-\mathbf{r}'|} = \frac{2}{\sqrt{\pi}} \int_0^\infty dt e^{-t^2(\mathbf{r}-\mathbf{r}')^2}. \tag{6.17}$$

and thus all the integrals (6.6) reduce to purely Gaussian integrals which can be performed analytically.

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

6.3.3 Electrons in solids

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

6.3.3.1 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.

6.3.4 Other basis sets

There is ongoing development of new basis sets, such as finite element or wavelet based approaches. One key problem for the simulation of large molecules is that since there are $\mathcal{O}(L^4)$ integrals of the type (6.6), quantum chemistry calculations typically scale as $\mathcal{O}(N^4)$. A big goal is thus to find smart basis sets and truncation schemes to reduce the effort to an approximately $\mathcal{O}(N)$ method, since the overlap of basis functions at large distances becomes negligibly small.

6.4 The Hartree Fock method

6.4.1 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:

$$\Phi(\mathbf{r}_1,\sigma_1;\ldots;\mathbf{r}_N,\sigma_N) = \frac{1}{\sqrt{N}} \begin{vmatrix} \phi_1(\mathbf{r}_1,\sigma_1) & \cdots & \phi_N(\mathbf{r}_1,\sigma_1) \\ \vdots & & \vdots \\ \phi_1(\mathbf{r}_N,\sigma_N) & \cdots & \phi_N(\mathbf{r}_N,\sigma_N) \end{vmatrix}. \tag{6.18}$$

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

6.4.2 Spinless case

To derive the Hartree-Fock equations it will be easiest to work in a second quantized notation. Also, for simplicity we will consider at first the spinless case, i.e.

$$\Phi(\mathbf{r}_1;\ldots;\mathbf{r}_N) = \frac{1}{\sqrt{N}} \begin{vmatrix} \phi_1(\mathbf{r}_1) & \cdots & \phi_N(\mathbf{r}_1) \\ \vdots & & \vdots \\ \phi_1(\mathbf{r}_N) & \cdots & \phi_N(\mathbf{r}_N) \end{vmatrix}, \tag{6.19}$$

and then the spinful one.

6.4.2.1 Expectation value of the energy

The first goal is to compute the expectation value of the Hamiltonian operator over the Slater determinant (6.19):

$$E \;=\; \langle\Phi|\hat{H}|\Phi\rangle, \tag{6.20}$$

using the fact that the state has the simple expression

$$|\Phi\rangle \;=\; \hat{c}^\dagger_{a_1}\hat{c}^\dagger_{a_2}\ldots\hat{c}^\dagger_{a_N}|0\rangle, \tag{6.21}$$

where in analogy with the previous lecture, the set $a_1 < a_2 \cdots < a_N$ refer to the indexes of the occupied orbitals, i.e. such that $n_{a_i} \neq 0$.

The second-quantized Hamiltonian (6.7) contains the one-body term:

$$\hat{H}_1 = \sum_{ij} t_{ij}\hat{c}^\dagger_i\hat{c}_j, \tag{6.22}$$

whose expectation value is

$$\langle\Phi|\hat{H}_1|\Phi\rangle \;=\; \sum_{ij} t_{ij}\langle 0|\hat{c}_{a_N}\ldots\hat{c}_{a_1}\hat{c}^\dagger_i\hat{c}_j\hat{c}^\dagger_{a_1}\ldots\hat{c}^\dagger_{a_N}|0\rangle, \tag{6.23}$$

from which we immediately see that we must have $i = j = a_f$, where a_f is one of the occupied orbitals. Therefore

$$\langle\Phi|\hat{H}_1|\Phi\rangle \ = \ \sum_f t_{a_f a_f}, \tag{6.24}$$

The two-body interaction term is:

$$\hat{H}_2 \ = \ \frac{1}{2}\sum_{ijkl} V_{ijkl}\hat{c}_i^\dagger \hat{c}_j^\dagger \hat{c}_l \hat{c}_k, \tag{6.25}$$

and the expectation value over the Slater determinant reads :

$$\langle\Phi|\hat{H}_2|\Phi\rangle = \frac{1}{2}\sum_{ijkl} V_{ijkl}\langle 0|\hat{c}_{a_N}\ldots\hat{c}_{a_1}\hat{c}_i^\dagger \hat{c}_j^\dagger \hat{c}_l \hat{c}_k \hat{c}_{a_1}^\dagger \ldots \hat{c}_{a_N}^\dagger|0\rangle. \tag{6.26}$$

In this expression, we see again that we must have $(l, k) = (a_f, a_g)$, where a_f and a_g are two occupied orbitals with $f \neq g$. There are now two possibilities:

1. $i = k = a_f$, $j = l = a_g$. This gives rise to the so-called *direct term :*

$$\begin{aligned}
&\frac{1}{2}\sum_{fg} V_{a_f a_g a_f a_g}\langle 0|\hat{c}_{a_N}\ldots\hat{c}_{a_1}\hat{c}_{a_f}^\dagger \hat{c}_{a_g}^\dagger \hat{c}_{a_g} \hat{c}_{a_f} \hat{c}_{a_1}^\dagger \ldots \hat{c}_{a_N}^\dagger|0\rangle = \\
&\qquad = -\frac{1}{2}\sum_{f\neq g} V_{a_f a_g a_f a_g}\langle 0|\hat{c}_{a_N}\ldots\hat{c}_{a_1}\hat{c}_{a_f}^\dagger \hat{c}_{a_g}^\dagger \hat{c}_{a_f} \hat{c}_{a_g} \hat{c}_{a_1}^\dagger \ldots \hat{c}_{a_N}^\dagger|0\rangle = \\
&\qquad\qquad = \frac{1}{2}\sum_{f\neq g} V_{a_f a_g a_f a_g}.
\end{aligned} \tag{6.27}$$

2. $i = l = a_f$ and $j = k = a_g$. This gives rise to the so-called *exchange term* :

$$-\frac{1}{2}\sum_{f\neq g} V_{a_f a_g a_g a_f}, \tag{6.28}$$

   where we have used the fact that $\hat{c}_i^\dagger \hat{c}_j{}^\dagger \hat{c}_i \hat{c}_j = -\hat{c}_i^\dagger \hat{c}_i \hat{c}_j{}^\dagger \hat{c}_j = -\hat{n}_i \hat{n}_j$.

The total energy therefore simply reads :

$$E \ = \ \sum_f t_{a_f a_f} + \frac{1}{2}\sum_{fg}\left(V_{a_f a_g a_f a_g} - V_{a_f a_g a_g a_f}\right), \tag{6.29}$$

where we have also eliminated the restriction $f \neq g$, since in that case the direct and exchange terms sum to zero. Notice that the second quantization formalism has allowed us to compute in a rather straightforward way this expectation value, without the painful complications that would emerge if we had attacked the wave-function starting from the definition of the determinant.

We can also try to grasp the physical meaning of the terms entering (6.29). Apart from the trivial kinetic energy term $\sum_f t_{a_f a_f}$, we have that the interactions lead to the direct (Hartree) term:

$$\begin{aligned} E_H &= \frac{1}{2}\sum_{fg} V_{a_f a_g a_f a_g} && (6.30)\\ &= \frac{1}{2}\sum_{fg}\int d\mathbf{r}d\mathbf{r}'\, |\phi_{a_f}(\mathbf{r})|^2 \frac{e^2}{|\mathbf{r}-\mathbf{r}'|} |\phi_{a_g}(\mathbf{r}')|^2 && (6.31)\\ &= \frac{1}{2}\int d\mathbf{r}d\mathbf{r}'\rho(\mathbf{r})\frac{e^2}{|\mathbf{r}-\mathbf{r}'|}\rho(\mathbf{r}'), && (6.32)\end{aligned}$$

where $\rho(\mathbf{r})$ is the total electron density in a given point $\mathbf{r}$. This term is just the classical electrostatic repulsion energy for the charge distribution $\rho(\mathbf{r})$. On the other hand, the exchange term leads to a purely quantum term, which does not have a classical analogy, and it is exclusively due to the anti-symmetrization properties of the fermionic wave-function.

6.4.3 Spin: Restricted Hartree Fock

We now relax the assumption that we have only spinless fermions. To simplify the discussion we assume the so-called "closed-shell" conditions, where each orbital is occupied by both an electron with spin ↑ and one with spin ↓ as well as the *restricted* Hartree Fock form for the spin-orbitals:

$$\phi_k(\mathbf{r},\sigma) = \phi_k(\mathbf{r}) \times \begin{cases}\alpha(\uparrow) \\ \beta(\downarrow)\end{cases}, \qquad (6.33)$$

i.e. the radial part of the orbital is independent of the spin value. The closed-shell condition also implies that each radial orbital $\phi_k(\mathbf{r})$ is occupied exactly by one spin up and one spin down. These conditions allow to enormously simplify the expression for the energy, which becomes:

$$E_{\text{RHF}} = \sum_f 2t_{a_f a_f} + \frac{1}{2}\sum_{fg}\left(2V_{a_f a_g a_f a_g} - V_{a_f a_g a_g a_f}\right)$$

The main difference with respect to the spinless case is that the both the kinetic term and direct term have an extra factor of 2. This can be understood from the fact that in the expectation value of the interaction energy:

$$\langle\Phi|\hat{H}_2|\Phi\rangle = \frac{1}{2}\sum_{ijkl\sigma\sigma'} V_{ijkl}\langle 0|\hat{c}_{a_N\downarrow}\hat{c}_{a_N\uparrow}\dots\hat{c}_{a_1\downarrow}\hat{c}_{a_1\uparrow}\hat{c}^\dagger_{i\sigma}\hat{c}^\dagger_{j\sigma'}\hat{c}_{l\sigma'}\hat{c}_{k\sigma}\hat{c}^\dagger_{a_1\uparrow}\hat{c}^\dagger_{a_1\downarrow}\dots\hat{c}^\dagger_{a_N\uparrow}\hat{c}^\dagger_{a_N\downarrow}|0\rangle, \qquad (6.34)$$

the exchange term is obtained setting the restriction $\sigma = \sigma'$, whereas the direct term does not have this restriction and thus pick an extra factor of 2.

6.4.4 The Roothaan-Hall Equations

We have now found a general expression for the HF energy. However, in order to concretely solve this equation on a computer, we need to express the orbitals in terms of some known (and fixed) basis set. In particular we concentrate again on the Restricted Hartree Fock case with closed shells, for which the only unknowns are the spatial parts of the orbitals. We write those as linear combinations of some other basis orbitals we have chosen:

$$|\phi_k\rangle = \sum_\beta C_{\beta k}|f_\beta\rangle. \tag{6.35}$$

We can then express the total energy in terms of the given orbitals $|f_\beta\rangle$, for example the one-body term reads :

$$\begin{aligned} 2\sum_i t_{ii} &= 2\sum_i \langle\phi_i|\hat{v}_1|\phi_i\rangle & (6.36)\\ &= 2\sum_i\sum_{\alpha\beta} C^\star_{\alpha i}C_{\beta i}\langle f_\alpha|\hat{v}_1|f_\beta\rangle & (6.37)\\ &= \sum_{\alpha\beta} P_{\alpha\beta}\bar{t}_{\alpha\beta} & (6.38)\end{aligned}$$

where we have introduced the so-called "density matrix"

$$P_{\alpha\beta} = 2\sum_i C^*_{\alpha i}C_{\beta i}, \tag{6.39}$$

and the matrix elements of the one-body term in the chosen fixed basis $\bar{t}_{\alpha\beta} = \langle f_\alpha|\hat{v}_1|f_\beta\rangle$.

A similar calculation for the direct and exchange terms can be carried out (left as an Exercise) and the energy reads

$$\begin{aligned} E_0 &= \sum_{\alpha\beta} P_{\alpha\beta}\bar{t}_{\alpha\beta} + \frac{1}{2}\sum_{\alpha\beta\gamma\delta} P_{\alpha\beta}P_{\gamma\delta}\left(\bar{V}_{\alpha\gamma\beta\delta} - \frac{1}{2}\bar{V}_{\alpha\gamma\delta\beta}\right) & (6.40)\\ &= \frac{1}{2}\sum_{\alpha\beta}\left(\bar{t}_{\alpha\beta} + F_{\alpha\beta}\right)P_{\alpha\beta}. & (6.41)\end{aligned}$$

Where we have introduced the so-called Fock matrix:

$$F_{\alpha\beta} = \bar{t}_{\alpha\beta} + \sum_{\gamma\delta} P_{\gamma\delta}\left(\bar{V}_{\alpha\gamma\beta\delta} - \frac{1}{2}\bar{V}_{\alpha\gamma\delta\beta}\right). \tag{6.42}$$

Since the energy is now an explicit function of the coefficients $C_{\alpha\beta}$ we can minimize it in order to find the best possible variational approximation to the ground state wave function. The only complication here is that we also need to impose that resulting orbitals are orthonormal

$$\begin{aligned} \langle\phi_k|\phi_l\rangle &= \sum_{\alpha\beta} C^\star_{\alpha k}C_{\beta l}\langle f_\alpha|f_\beta\rangle & (6.43)\\ &= \delta_{kl}, & (6.44)\end{aligned}$$

thus the energy needs to be minimized by imposing such constraints using a Lagrange multiplier approach. By performing the minimization explicitly (a straightforward yet tedious exercise) one finds that the matrix of coefficients have to satisfy the Equation

$$\sum_{\beta}(F_{\alpha\beta} - \epsilon_k S_{\alpha\beta})C_{\beta k} = 0, \tag{6.45}$$

first derived by Roothaan and Hall in the 1950s. In this Equation we have introduced

$$S_{\alpha\beta} \equiv \langle f_\alpha | f_\beta \rangle, \tag{6.46}$$

$$= \int d\mathbf{r} f^\star_\alpha(\mathbf{r}) f_\beta(\mathbf{r}), \tag{6.47}$$

the positive-definite overlap (or Gramian) matrix, which for an orthogonal basis would simply be the identity, as well as the coefficients ϵ_k, which are the Lagrange multipliers resulting from imposing the orthogonality condition. A full derivation of this Equation can be found in quantum chemistry textbooks. Here we just notice that this equation takes the form of a generalized eigenvalue problem: $\hat{A}|x\rangle = \lambda_x \hat{B}|x\rangle$.

Since the matrix $F_{\alpha,\beta}$ depends itself on the coefficients C (the generalized eigenvectors), this equation is however intrinsically nonlinear, and cannot be solved "in one shot". The equation is instead typically solved iteratively, using the previous value of the coefficients C in defining the matrix F, solving the generalized eigenvalue problem for new values of C etc, until convergence to a fixed point is achieved.

6.4.5 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

$$|\Phi\rangle = \prod_{\mu=1}^{N} \hat{c}^\dagger_\mu |0\rangle \tag{6.48}$$

one or two of the $\hat{c}^\dagger_\mu$ are replaced by other (unoccupied) orbitals $\hat{c}^\dagger_i$:

$$|\psi_0\rangle = \left(1 + \sum_{i,\mu} \alpha^i_\mu \hat{c}^\dagger_i \hat{c}_\mu + \sum_{i<j,\mu<\nu} \alpha^{ij}_{\mu\nu} \hat{c}^\dagger_i \hat{c}^\dagger_j \hat{c}_\mu \hat{c}_\nu \right) |\Phi\rangle. \tag{6.49}$$

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.

6.5 Further Reading

This lecture is meant to give you an overview of the main techniques adopted (mostly) in Chemistry to find the electronic structure of molecules. The topics we can cover however are rather limited, and are typically the subject of extensive Quantum Chemistry courses, that are beyond the scope of our syllabus. The interested reader can find most of the technical details we have omitted in this book:

- "Modern Quantum Chemistry" by Attila Szabo and Neil S. Ostlund (especially chapter 2.4, using the second quantization formalism)

Chapter 7

Density functional theory

In the previous lecture we have introduced the Electronic Structure problem, and discussed the Hartree-Fock approximation to solve it.

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 (DFT) 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.

The starting point of our analysis is again the "standard Model" in Condensed Matter Physics, namely the Hamiltonian

$$\hat{H} = \sum_{i=1}^{N} \left(-\frac{\hbar^2}{2m} \nabla^2_{\mathbf{r}_i} + V_{en}(\mathbf{r}_i) \right) + \sum_{i<j} \frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|}, \tag{7.1}$$

where we have once more made use of the Born-Oppenheimer approximation, and considered the interaction with the nuclei as an *external potential* seen by the electrons, and depending parametrically on the nuclei positions,

$$V_{en}(\mathbf{r}) = -e^2 \sum_{i=1}^{M} \frac{Z_i}{|\mathbf{R}_i - \mathbf{r}|}. \tag{7.2}$$

This external potential is what determines basically all the chemical and physical properties of molecules and materials, and it can be seen as a "non-universal" addition to the electron gas Hamiltonian

$$\begin{aligned} \hat{H}_{eg} &= -\frac{\hbar^2}{2m} \sum_{i=1}^{N} \nabla^2_{\mathbf{r}_i} + \sum_{i<j} \frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|} \qquad (7.3) \\ &= \hat{T}_e + \hat{W}_{ee}. \end{aligned}$$

Goal of the DFT is, in a nutshell, to use some universal features of the electron gas in order to solve the full Electronic Structure problem.

7.1 The electron density

DFT is heavily based on the electron density, defined as the expectation value of the density operator

$$\hat{\rho}(\mathbf{r}) \;=\; \sum_{i=1}^{N} \delta(\mathbf{r}-\mathbf{r}_i)$$

over a given fermionic state Ψ:

$$\begin{aligned}\hat{\rho}(\mathbf{r}) &= \int d\mathbf{r}_1 d\mathbf{r}_2 \ldots d\mathbf{r}_N \left|\Psi(\mathbf{r}_1,\mathbf{r}_2,\ldots,\mathbf{r}_N)\right|^2 \sum_{i=1}^{N} \delta(\mathbf{r}-\mathbf{r}_i) \\ &= N\int d\mathbf{r}_2 \ldots d\mathbf{r}_N \left|\Psi(\mathbf{r},\mathbf{r}_2,\ldots,\mathbf{r}_N)\right|^2,\end{aligned}$$

where we have used the permutation symmetry of the wave-function, and ignored the spin degrees of freedom (as for the Hartree-Fock, for simplicity we will derive all the necessary equations in the spinless case). Notice also that these expressions are in first quantization, and that the density is normalized such that $\int d\mathbf{r}\rho(\mathbf{r}) = N$.

7.2 Variational principle for the density

The standard variational principle states that given some state Ψ and a Hamiltonian $\hat{H}$,

$$\langle\Psi|\,\hat{H}\,|\Psi\rangle \;\geq\; E_0,$$

where E_0 is the exact ground-state energy of H, and the equal sign is obtained whenever $\Psi = \Psi_0$, the exact ground-state wave function. We can also state then that the wave-function satisfies the following minimization problem:

$$E_0 \;=\; \min_{\Psi} \langle\Psi|\,\hat{H}\,|\Psi\rangle\,,$$

where we search for the optimal Ψ that minimizes the functional $E[\Psi] = \langle\Psi|\,\hat{H}\,|\Psi\rangle$ among the (huge) space of all possible $N-$body normalized wave-functions. Kohn first realized that the variational principle can be written in terms of an optimization procedure for the density itself, instead of the wave-function. This idea can be formulated following Levy's constrained-search approach.

We start writing

$$\begin{aligned}E_0 &= \min_{\Psi} \langle\Psi|\,\hat{H}\,|\Psi\rangle \\ &= \min_{\rho}\min_{\Psi[\rho]} \left[\langle\Psi|\,\hat{T}_e + \hat{W}_{ee} + \sum_i V_{en}(\mathbf{r}_i)\,|\Psi\rangle\right],\end{aligned}$$

where we have separated the minimization over Ψ in a two-step process: first we fix the density $\rho(\mathbf{r})$, and minimize over all possible $N-$body states $\Psi[\rho]$ which have the density

ρ. Then, we do a further minimization over the density. It is clear that this procedure is completely equivalent to the original formulation of the optimization procedure. Then, we notice that

$$\langle\Psi|\sum_i V_{en}(\mathbf{r}_i)|\Psi\rangle = \int d\mathbf{r}\rho(\mathbf{r})V_{en}(\mathbf{r}), \tag{7.4}$$

$$\equiv \bar{V}_{en}[\rho] \tag{7.5}$$

which follows directly from the previous definition of the density operator, and in the last line we have made explicit that the expectation value of the one-body potential is a functional of the density. This term indeed does not depend explicitly on $\Psi[\rho]$ (since all wave-functions $\Psi[\rho]$ give the same density) and we are left only with the minimization over the density. Putting the terms together we have

$$E_0 = \min_\rho \left[F[\rho] + \bar{V}_{en}[\rho]\right], \tag{7.6}$$

where we have introduced the *density functional*

$$F[\rho] = \min_{\Psi[\rho]}\langle\Psi[\rho]|\hat{T}_e + \hat{W}_{ee}|\Psi[\rho]\rangle.$$

This is the central object of DFT, and it is an universal quantity in the sense that it does not depend on the external potential $V_{en}(\mathbf{r})$.

The formulation of the variational principle (7.6) makes our life very easy: instead of minimizing with respect to the many-body wave functions, we only have to minimize with respect to the density (a function of just 3 spatial variables) and we obtain both the ground state energy and the electron density in the ground state – and everything is exact. This is a tremendous simplification with respect to the original problem of solving for the ground-state wave-function, which lives in a much higher dimensional space.

However, the problem is that, while the functional F is universal, it is also unknown! Thus for the DFT to be useful, we need to find good approximations for this universal functional.

7.3 The Kohn-Sham scheme

Kohn and Sham proposed to decompose the density functional as

$$F[\rho] = E_k[\rho] + E_h[\rho] + E_{xc}[\rho]. \tag{7.7}$$

The first term reads:

$$E_k[\rho] = \min_{\Phi[\rho]}\langle\Phi[\rho]|\hat{T}_e|\Phi[\rho]\rangle,$$

where the minimization is done not all over possible many-body states $\Psi[\rho]$ at the given density, but instead on the smaller set of single-particle (Slater determinants) wave-functions $\Phi[\rho]$ at given density. The Hartree-term $E_h[\rho]$ has an explicit expression and it is given by the Coulomb repulsion between two electrons:

$$E_h[\rho] = \frac{1}{2}\int d\mathbf{r}d\mathbf{r}'\rho(\mathbf{r})\frac{e^2}{|\mathbf{r}-\mathbf{r}'|}\rho(\mathbf{r}'), \tag{7.8}$$

as we have also seen in the previous lecture. The exchange- and correlation term $E_{xc}[\rho]$ contains instead the remaining unknown contribution, and it is implicitly defined by Equation (7.7). We will discuss how to approximate it a bit later.

A consequence of this decomposition of the density functional is that we can write the variational minimization as:

$$\begin{aligned} E_0 &= \min_\rho \left[F[\rho] + \bar{V}_{en}[\rho]\right] \\ &= \min_\rho \left[\min_{\Phi[\rho]}\langle\Phi|\hat{T}_e|\Phi\rangle + \bar{V}_{en}[\rho] + E_h[\rho] + E_{xc}[\rho]\right] \\ &= \min_\rho \left[\min_{\Phi[\rho]}\left[\langle\Phi|\hat{T}_e|\Phi\rangle + \bar{V}_{en}[\rho_\Phi] + E_h[\rho_\Phi] + E_{xc}[\rho_\Phi]\right]\right], \end{aligned} \tag{7.9}$$

in other words the minimization is done now on all the single-particle wave-functions which have some density ρ_Φ. Notice that here we have implicitly assumed that single-particle wave-functions are able to represent all possible ground-state densities profiles $\rho(\mathbf{r})$ generated by the many-body wave-functions. This assumption is reasonable but hard to prove rigorously.

To calculate the ground state energy (and other properties) we have to minimize the energy functional in (7.9) with respect to the density, solving the variational problem

$$0 = \delta E[\rho] = \int d\mathbf{r}\delta\rho(\mathbf{r})\left(V_{en}(\mathbf{r}) + e^2\int d\mathbf{r}'\frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} + \frac{\delta E_k[\rho]}{\delta\rho(\mathbf{r})} + \frac{\delta E_{xc}[\rho]}{\delta\rho(\mathbf{r})}\right) \tag{7.10}$$

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

$$\int d\mathbf{r}\delta\rho(\mathbf{r}) = 0. \tag{7.11}$$

Since the minimization is carried over densities coming from single-particle Slater determinant wave-functions, this variational problem is equivalent to solving the Schroedinger equation for a non-interacting system in an external potential:

$$\left(-\frac{1}{2m}\nabla^2 + V_{\text{KS}}(\mathbf{r})\right)\phi_\mu(\mathbf{r}) = \epsilon_\mu\phi_\mu(\mathbf{r}), \tag{7.12}$$

where the Kohn-Sham potential of the non-interacting system is

$$V_{\text{KS}}(\mathbf{r}) = V_{en}(\mathbf{r}) + e^2\int d\mathbf{r}'\frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} + V_{\text{xc}}(\mathbf{r}), \tag{7.13}$$

and the exchange-correlation potential is defined by

$$V_{\text{xc}}(\mathbf{r}) = \frac{\delta E_{\text{xc}}[\rho]}{\delta\rho(\mathbf{r})}. \tag{7.14}$$

The Equations (7.12) are know as Kohn-Sham equations, and effectively reduce the many-body problem to the solution of a much simpler one-body problem with an effective potential $V_{\mathrm{KS}}(\mathbf{r}) = V_{\mathrm{en}}(\mathbf{r}) + \mathcal{J}(\mathbf{r}) + V_{\mathrm{xc}}(\mathbf{r})$. This should be contrasted with the Hartree-Fock approximation, which we have seen gives rise as well to a one-body problem but with an effective potential $V_{HF}(\mathbf{r}) = V_{en}(\mathbf{r}) + \mathcal{J}(\mathbf{r}) - \mathcal{K}(\mathbf{r})$. Comparing the two expressions we therefore see that $V_{xc}(\mathbf{r})$ contains both the exchange part (as per the term $\mathcal{K}(\mathbf{r})$), but also all the other corrections due to the correlations between the electrons, which are not captured by the Hartree Fock mean-field treatment.

For some given $V_{xc}(\mathbf{r})$, the non-linear equations (7.12) are solved in the same way we solved the Hartree-Fock equations. In particular, they are solved iteratively, using some finite basis to express the unknown Kohn-Sham orbitals $\phi_\mu(\mathbf{r})$.

7.4 Local Density Approximation

The Kohn-Sham equations discussed so-far are in principle exact. However, the functional $E_{xc}[\rho]$ and thus the potential $V_{xc}(\mathbf{r})$ is not known, and it was dubbed the *stupidity functional* by Richard Feynman, in the sense that we are just moving our original ignorance of the many-body system to the ignorance of this functional. To proceed further we need to introduce approximations for this functional.

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 $E[\rho](\mathbf{r})$ which could be a function of $\rho(\mathbf{r}), \nabla\rho(\mathbf{r}), \nabla\nabla\rho(\mathbf{r}), \dots$ we ignore all the gradients and just take the local density

$$E_{xc}^{LDA}[\rho] = \int d\mathbf{r}\rho(\mathbf{r})E_{xc}^{\mathrm{eg}}(\rho(\mathbf{r})), \tag{7.15}$$

where $E_{xc}^{\mathrm{eg}}(\rho)$ is the exact exchange-correlation energy per particle of the *homogeneous electron gas,* (7.3). This approximation is clearly exact for the uniform electron gas (i.e. in the absence of any external potential) for which the density is constant, since the total Hamiltonian is translational invariant. In the case in which we have an external potential, this approximation is expected to work well when the electron density does not have strong spatial variations (i.e. corrections coming from the gradients of the density can be neglected).

To proceed further, we need an expression for the exchange-correlation energy $E_{xc}^{\mathrm{eg}}(\rho)$ in the uniform electron gas. Recalling the definition (7.7), we immediately see that

$$\begin{aligned} E^{\mathrm{eg}}[\rho] &= F^{\mathrm{eg}}[\rho] \\ &= E_k[\rho] + E_h^{\mathrm{eg}}[\rho] + E_{xc}^{\mathrm{eg}}[\rho], \end{aligned}$$

therefore if the exact energy of the electron gas is known, $E^{\mathrm{eg}}(\rho)$, $E_{xc}^{\mathrm{eg}}(\rho)$ can be found just subtracting the (known) Hartree term and the kinetic energy of the non-interacting electrons. In practice, very accurate energies $E^{\mathrm{eg}}(\rho)$ (at various values of the density ρ) can be obtained from quantum Monte Carlo calculations, and explicit expressions for $E_{xc}^{\mathrm{eg}}(\rho)$ are available.

The exchange correlation potential is then found doing the functional derivative of (7.15)

$$\begin{aligned} V_{xc}^{LDA}(\mathbf{r}) &= \frac{\delta E_{xc}[\rho]}{\delta \rho(\mathbf{r})} \\ &= E_{xc}^{\text{eg}}(\rho(\mathbf{r})) + \rho(\mathbf{r}) \nabla_\rho E_{xc}^{\text{eg}}(\rho(\mathbf{r})). \end{aligned}$$

Putting all together, one arrives at the following explicit expression for the LDA exchange and correlation potential:

$$r_s^{-1} = a_B \left(\frac{4\pi}{3} \rho \right)^{1/3} \tag{7.16}$$

$$V_{xc}^{LDA} = -\frac{e^2}{a_B} \left(\frac{3}{2\pi} \right)^{2/3} \frac{1}{r_s} \left[1 + 0.0545 r_s \ln(1 + 11.4/r_s) \right], \tag{7.17}$$

where the numerical factors come from fitting the quantum Monte Carlo data for $E^{\text{eg}}(\rho)$ to a semi-empirical expression based on many-body perturbation theory.

7.5 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 ↑ and ↓. The "generalized gradient approximation" (GGA) and its variants use functionals depending not only on the density, but also on its derivatives.

7.6 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.

7.7 Further Reading

Levy's constrained search is introduced in:

- *M. Levy*, Proc. Natl. Acad. Sci. USA 76, 6062 (1979)
- *M. Levy,* Phys. Rev. A 26, 1200 (1982)
- *E. H. Lieb,* Int. J. Quantum Chem. 24, 24 (1983).

Chapter 8

Variational Monte Carlo and Stochastic Optimization

In the previous lectures we have extensively made use of the variational principle. This principle states that

$$E(\theta_1, \dots \theta_M) = \langle \Psi(\theta_1, \dots \theta_M) | \hat{H} | \Psi(\theta_1, \dots \theta_M) \rangle \geq E_0, \tag{8.1}$$

where $\Psi(\theta_1 \dots \theta_M)$ is some *ansatz* wave-function depending on a set of M parameters, and E_0 is the exact ground-state energy of the Hamiltonian $\hat{H}$.

Apart from the very special case of mean-field-like ansatz wave-functions (like the single determinant used in the Hartree-Fock method), it is seldom possible to compute analytically $E(\theta_1, \dots \theta_M)$ for a generic variational state Ψ. The goal of this lecture is to introduce a series of stochastic techniques that allow to obtain accurate estimates of the variational energies for a given set of parameters $\mathbf{p}$, as well as approaches to optimize those parameters in order to obtain the best possible ground-state energy within the given *ansatz* form. Albeit this approach is principle approximate (since the chosen form of Ψ might not contain the exact ground-state wave-function) however modern optimization techniques, in conjunction with modern many-body variational states encompassing thousands or more variational parameters, effectively allow to solve for the ground-state properties with very high accuracy.

8.1 Variational Monte Carlo

The Variational Monte Carlo method is rooted into the observation that expectation values like (8.1) can be written as statistical averages over a suitable probability distribution. Let us assume that our Hilbert space is spanned by the many-body kets $|\mathbf{x}\rangle$. These in practice depend on the system in exam. For example in the case of spins 1/2 we have seen that one would typically have $|\mathbf{x}\rangle = |s_1 s_2 \dots s_N\rangle$, whereas for second-quantized fermions $|\mathbf{x}\rangle = |n_1 n_2 \dots n_N\rangle$, for particles in continuous space $|\mathbf{x}\rangle = |\mathbf{r}_1 \mathbf{r}_2 \dots \mathbf{r}_N\rangle$, etc. The only difference is of course that in the first two cases one has a discrete set of quantum numbers, whereas in the latter case the degrees of freedom are continuos. In all cases we will denote sums over the Hilbert space with discrete sums, although one should always bear in mind that in the case of continuous variables these sums must be interpreted as integrals. In particular we will use the closure relation $\sum_{\mathbf{x}} |\mathbf{x}\rangle\langle\mathbf{x}| = \hat{I}$.

8.1.1 Stochastic Estimates of Properties

Using the closure relation, we can rewrite a generic quantum expectation value of some operator $\hat{O}$ as

$$\frac{\langle\Psi|\hat{O}|\Psi\rangle}{\langle\Psi|\Psi\rangle} = \frac{\sum_{\mathbf{x},\mathbf{x}'}\langle\Psi|\mathbf{x}\rangle\langle\mathbf{x}|\hat{O}|\mathbf{x}'\rangle\langle\mathbf{x}'|\Psi\rangle}{\sum_{\mathbf{x}}\langle\Psi|\mathbf{x}\rangle\langle\mathbf{x}|\Psi\rangle} \tag{8.2}$$

$$= \frac{\sum_{\mathbf{x},\mathbf{x}'}\Psi^{\star}(\mathbf{x})O_{\mathbf{x}\mathbf{x}'}\Psi(\mathbf{x}')}{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2}. \tag{8.3}$$

There can be, in general, two cases:

1. The operator $\hat{O}$ is diagonal in the computational basis, i.e. $O_{\mathbf{x}\mathbf{x}'} = \delta_{\mathbf{x}\mathbf{x}'}O(\mathbf{x})$. Then

$$\frac{\langle\Psi|\hat{O}|\Psi\rangle}{\langle\Psi|\Psi\rangle} = \frac{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2 O(\mathbf{x})}{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2} \tag{8.4}$$

$$\equiv \mathbb{E}_{\Pi}[O(\mathbf{x})], \tag{8.5}$$

   where $\mathbb{E}_{\Pi}[\dots]$ denote *statistical* expectation values over the probability distribution $\Pi(\mathbf{x}) = |\Psi(\mathbf{x})|^2 / \sum_{\mathbf{x}'}|\Psi(\mathbf{x}')|^2$. In other words, in this case quantum expectation values are completely equivalent to averaging over Hilbert-space states sampled according to the square-modulus of the wave-function.

2. The operator $\hat{O}$ is off-diagonal in the computational basis. Then, we can define an auxiliary diagonal operator (often called, in a somehow misleading fashion, *local* operator or estimator)

$$O_{\mathrm{loc}}(\mathbf{x}) = \sum_{\mathbf{x}'} O_{\mathbf{x}\mathbf{x}'}\frac{\Psi(\mathbf{x}')}{\Psi(\mathbf{x})}, \tag{8.6}$$

   such that it is easily seen that

$$\frac{\langle\Psi|\hat{O}|\Psi\rangle}{\langle\Psi|\Psi\rangle} = \frac{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2 O_{\mathrm{loc}}(\mathbf{x})}{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2} \tag{8.7}$$

$$\equiv \mathbb{E}_{\Pi}[O_{\mathrm{loc}}(\mathbf{x})]. \tag{8.8}$$

For any observable, then, we can always compute expectation values over arbitrary wave-functions as statistical averages. In the case of off-diagonal operators, it should be noticed that the sum $\sum_{\mathbf{x}'} O_{\mathbf{x}\mathbf{x}'}\frac{\Psi(\mathbf{x}')}{\Psi(\mathbf{x})}$, is extended to the tiny portion of the Hilbert space for which $\mathbf{x}'$ is such that $|O_{\mathbf{x}\mathbf{x}'}| \neq 0$. As we have already seen when performing exact diagonalization, matrix elements of physical operators are typically such that they are row/column sparse, meaning that for fixed $\mathbf{x}$, the number of elements $\mathbf{x}'$ such that $|O_{\mathbf{x}\mathbf{x}'}| \neq 0$ is polynomial in the system size. In turn this implies that, for given $\mathbf{x}$, the local estimator $O_{\mathrm{loc}}(\mathbf{x})$ can always be computed in polynomial time, provided also that computing the amplitudes $\langle \mathbf{x}|\Psi\rangle$ can be done in polynomial time. In the rest of this and of the following lectures, we will always restrict our attentions to variational wave functions that have this property.

While the local estimator can be computed efficiently, however the summations in $\sum_{\mathbf{x}} |\Psi(\mathbf{x})|^2$ are typically not exactly doable, since one has an exponentially large number of possible values of $\mathbf{x}$ on which to perform the summation, and therefore cannot be done by brute-force. Think for example to the case in which you have $|\mathbf{x}\rangle = |s_1 s_2 \dots s_N\rangle$, a system of N spins 1/2. In that case, the summation over $\mathbf{x}$ implies summing over 2^N possible values, which soon becomes unfeasible as N grows. Similarly, this exponential grows applies to all the other many-body systems of interest.

The powerful idea of the Variational Monte Carlo (VMC) is exactly to replace these sums over exponentially many states, with a statistical average over a large but finite set of states sampled according to the probability distribution $\Pi(\mathbf{x})$. We therefore have a way to compute, stochastically, the expectation value of *all* the properties of interest, provided we have a way to perform statistical averages efficiently. For example, this strategy will allow us to compute the expectation value of $\hat{\sigma}_i^x$ for a spin system, the expectation value of $\hat{c}_i^\dagger \hat{c}_j$ for fermions, or even the expectation value of the interaction energy $W_{ee}(\vec{r}_1 \dots \vec{r}_N)$ for our electronic structure problems.

8.1.1.1 Energy

An immediate corollary of the previously presented scheme, is that also the expectation value of the Hamiltonian $\hat{H}$ (which is itself a generic off-diagonal operator) can be computed using the estimator (8.8). Historically, the local estimator associated to the Hamiltonian is called "local energy":

$$E_{\mathrm{loc}}(\mathbf{x}) \quad = \quad \sum_{\mathbf{x}'} H_{\mathbf{x}\mathbf{x}'} \frac{\Psi(\mathbf{x}')}{\Psi(\mathbf{x})}. \tag{8.9}$$

Notice that this expression (and the equivalent above for local estimators) is strictly valid only for discrete Hilbert spaces. A more general form valid also for continuous Hilbert spaces is

$$E_{\mathrm{loc}}(\mathbf{x}) \quad = \quad \frac{\langle \mathbf{x}|\hat{H}|\Psi\rangle}{\langle \mathbf{x}|\Psi\rangle}. \tag{8.10}$$

8.2 Stochastic Variational Optimization

The final goal we want to achieve here is to optimize the variational energy with respect to the parameters $\boldsymbol{\theta} = \theta_1, \dots \theta_M$. We have seen that the expectation value of the energy can be written as a statistical average of the form

$$\langle \hat{H} \rangle \quad \simeq \quad \mathbb{E}_{\Pi}[E_{\mathrm{loc}}(\mathbf{x})]. \tag{8.11}$$

It is easy to show that also the gradient of the energy can be written under to form of the expectation value of some stochastic variable. In particular, define

$$D_k(\mathbf{x}) \quad = \quad \frac{\partial_{\theta_k} \Psi(\mathbf{x})}{\Psi(\mathbf{x})}, \tag{8.12}$$

then

$$
\begin{aligned}
\partial_{\theta_k}\langle\hat{H}\rangle &= \partial_{\theta_k}\frac{\sum_{\mathbf{x},\mathbf{x}'}\Psi^\star(\mathbf{x})H_{\mathbf{x}\mathbf{x}'}\Psi(\mathbf{x}')}{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2} \\
&= \frac{\sum_{\mathbf{x},\mathbf{x}'}\Psi^\star(\mathbf{x})H_{\mathbf{x}\mathbf{x}'}D_k(\mathbf{x}')\Psi(\mathbf{x}')}{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2}+\frac{\sum_{\mathbf{x},\mathbf{x}'}\Psi^\star(\mathbf{x})D_k^\star(\mathbf{x})H_{\mathbf{x}\mathbf{x}'}\Psi(\mathbf{x}')}{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2} \\
&\quad -\frac{\sum_{\mathbf{x},\mathbf{x}'}\Psi^\star(\mathbf{x})H_{\mathbf{x}\mathbf{x}'}\Psi(\mathbf{x}')}{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2}\frac{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2\left(D_k(\mathbf{x})+D_k^\star(\mathbf{x})\right)}{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2} \\
&= \frac{\sum_{\mathbf{x},\mathbf{x}'}\frac{\Psi^\star(\mathbf{x})}{\Psi^\star(\mathbf{x}')}H_{\mathbf{x}\mathbf{x}'}D_k(\mathbf{x}')|\Psi(\mathbf{x}')|^2+\sum_{\mathbf{x},\mathbf{x}'}|\Psi(\mathbf{x})|^2H_{\mathbf{x}\mathbf{x}'}D_k^\star(\mathbf{x}')\frac{\Psi(\mathbf{x}')}{\Psi(\mathbf{x})}}{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2}+ \\
&\quad -\langle H\rangle\frac{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2\left(D_k(\mathbf{x})+D_k^\star(\mathbf{x})\right)}{\sum_{\mathbf{x}}|\Psi(\mathbf{x})|^2} \\
&= \mathbb{E}_\Pi[E_{\mathrm{loc}}(\mathbf{x})D_k^\star(\mathbf{x})]-\mathbb{E}_\Pi[E_{\mathrm{loc}}(\mathbf{x})]\mathbb{E}_\Pi[D_k^\star(\mathbf{x})]+\mathrm{cc}. \qquad (8.13)
\end{aligned}
$$

We can therefore compactly write $\partial_{\theta_k}\langle\hat{H}\rangle = \mathbb{E}_\Pi[G_k(\mathbf{x})]$, with the gradient estimator being

$$
G_k(\mathbf{x}) = 2\mathrm{Re}\left[\left(E_{\mathrm{loc}}(\mathbf{x})-\mathbb{E}_\Pi[E_{\mathrm{loc}}(\mathbf{x})]\right)D_k^\star(\mathbf{x})\right]. \qquad (8.14)
$$

8.2.1 Zero-Variance Property

One of the most interesting feature of the energy and energy-gradient estimators so-far presented is that they have the so-called zero-variance property: their statistical fluctuations are exactly zero when sampling from the exact ground-state wave-function. Let us consider for example

$$
\begin{aligned}
\mathrm{Var}_\Pi[E_{\mathrm{loc}}(\mathbf{x})] &= \mathbb{E}_\Pi[E_{\mathrm{loc}}(\mathbf{x})^2]-\mathbb{E}_\Pi[E_{\mathrm{loc}}(\mathbf{x})]^2 \\
&= \sum_{\mathbf{x}}\Psi(\mathbf{x})^2E_{\mathrm{loc}}(\mathbf{x})^2-\langle H\rangle^2 \\
&= \sum_{\mathbf{x}}\sum_{\mathbf{x_1}}H_{\mathbf{x},\mathbf{x}_1}\Psi(\mathbf{x}_1)\sum_{\mathbf{x_2}}H_{\mathbf{x},\mathbf{x}_2}\Psi(\mathbf{x}_2)-\langle H\rangle^2 \\
&= \sum_{\mathbf{x_1}}\Psi(\mathbf{x}_1)\sum_{\mathbf{x}}H_{\mathbf{x},\mathbf{x}_1}\sum_{\mathbf{x_2}}H_{\mathbf{x},\mathbf{x}_2}\Psi(\mathbf{x}_2)-\langle H\rangle^2 \\
&= \langle\hat{H}^2\rangle-\langle\hat{H}\rangle^2, \qquad (8.15)
\end{aligned}
$$

where we have assumed for simplicity that the wave-function is real. Therefore the variance of the local energy is an important physical quantity: the energy variance. It is easy to see that if Ψ is an eigenstate of H then $\langle\hat{H}^2\rangle = \langle\hat{H}\rangle^2 = E_0^2$, and $\mathrm{Var}_\Pi[E_{\mathrm{loc}}(\mathbf{x})] = 0$, i.e. the statistical fluctuations completely vanish. This property is very important since it also implies that, in a sense to be specified below, the closer we get to the ground-state, the less fluctuations we have on the quantity we want to minimize, the energy.

8.2.2 Stochastic Gradient Descent

The gradient descent method is the simplest optimization scheme, where at each iteration i the variational parameters are modified according to

$$\theta_k^{i+1} = \theta_k^i - \eta \partial_{\theta_k} \langle \hat{H} \rangle, \tag{8.16}$$

where η is a (small) parameter called the "learning rate" in the machine learning community. An important difference with respect to the non-stochastic (deterministic) gradient descent approach, is that now we only have stochastic averages of the gradient which is therefore subjected to noise. Let us assume for simplicity that all the components of the gradient are subjected to the same amount of gaussian noise with standard deviation σ, i.e.

$$\partial_{\theta_k} \langle \hat{H} \rangle = \mathbb{E}_{\Pi}[G_k(\mathbf{x})] + \mathrm{Normal}\left(0, \sigma\right), \tag{8.17}$$

where the variance is due to the fact that we are estimating the gradient using a finite number of samples, thus for the central limit theorem it will be equal to

$$\sigma^2 = \frac{\mathrm{Var}_{\Pi}[G_k(\mathbf{x})]}{N_s}. \tag{8.18}$$

We can then compare Eq. 67 to the discretized Langevin equation (for example as found in Brownian motion):

$$r_k^{i+1} = r_k^i - \delta_t \partial_{r_k} E(\mathbf{r}) + \mathrm{Normal}\left(0, \sqrt{2\delta_t T}\right), \tag{8.19}$$

where δ_t is a small time step, such that the particle positions $\mathbf{r}(t = \delta_t i) \equiv \mathbf{r}^i$ evolve in time under the action of a conservative force (the first term) and a stochastic force, the second term. It can be shown that the stationary distribution of the Langevin process is the Boltzmann distribution

$$\Pi_B(\mathbf{r}) = \frac{e^{-\frac{E(\mathbf{r})}{T}}}{Z(T)}, \tag{8.20}$$

which in the limit $T \to 0$ would converge to the ground-state of the energy potential, i.e. to $\min_{\mathbf{r}} E(\mathbf{r})$.

To complete the analogy between stochastic gradient descend and Langevin dynamics, we can identify the parameters and the particle positions, such that

$$E(\mathbf{r}) \rightarrow E(\theta) \tag{8.21}$$

$$\partial_{r_k} E(\mathbf{r}) \rightarrow \mathbb{E}_{\Pi}[G_k(\mathbf{x})] \tag{8.22}$$

$$\delta_t \rightarrow \eta \tag{8.23}$$

$$\sqrt{2\delta_t T} \rightarrow \eta \sqrt{\frac{\mathrm{Var}_{\Pi}[G_k(\mathbf{x})]}{N_s}} \tag{8.24}$$

We therefore see that the variance of the gradient corresponds to the effective temperature

$$T_{\text{eff}} \propto \frac{\eta \mathrm{Var}_{\Pi}[G_k(\mathbf{x})]}{N_{\mathrm{s}}}. \tag{8.25}$$

Since we want to find the variational ground state (that corresponds to the state with $T_{\text{eff}} = 0$ in this analogy), we should have a scheme in which the effective temperature is gradually decreased at each optimization step, i.e. $T_1 > T_2 > T_3 \dots$, as in the simulated annealing optimization protocol. Given the form of the effective temperature, convenient ways to reduce the temperature are either to reduce the learning rate: $\eta(i) = \eta_0/\sqrt{i+1}$ or to increase the number of samples N_s with the iteration count.

During the optimization however it often happens that if we are close enough to the ground-state solution $\text{Var}_\Pi[G_k(\mathbf{x})] \to 0$. Indeed, we have shown before that for an exact eigenstate the statistical fluctuations of the gradient are exactly vanishing, i.e. $\text{Var}_\Pi[G_k] = 0$. In practice then, even a constant number of samples and a fixed (small) η are sufficient to converge to the ground-state, provided that one checks during the optimization that the value of the effective temperature (8.25) is actually going to zero as expected.

8.3 Sampling Methods

Once established this fundamental connection between variational methods, optimization of wave functions and statistical sampling, we need an efficient way of sampling from the probability distribution $\Pi(\mathbf{x}) = |\Psi(\mathbf{x})|^2$, and compute the required expectation values. In particular the goal is to generate N_s samples $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots \mathbf{x}^{(N_s)}$ such that we can estimate expectation values as averages over those samples, for example for the local energy:

$$\mathbb{E}_\Pi[O_{\text{loc}}(\mathbf{x})] \simeq \frac{1}{N_s}\sum_i^{N_s} O_{\text{loc}}(\mathbf{x}^{(i)}). \tag{8.26}$$

8.3.1 Markov Chain and Detailed Balance

Devising strategies to sample from a given probability distribution is, in general, a complex computational task. This task can be simplified if the probability to be sampled from has a special structure, however in general there is a family of sampling techniques that are rather universal, and known as Markov-Chain Monte Carlo (MCMC). Here we review these algorithms, that you have already covered in previous courses.

A Markov chain is completely specified by the transition probability $\mathcal{T}(\mathbf{x}^{(i)} \to \mathbf{x}^{(i+1)})$, i.e. given a sample $\mathbf{x}^{(i)}$, we transition to the next element of the chain with probability T. The transition probability (as all well-defined probabilities) must always be normalized: $\sum_{\mathbf{x}'} \mathcal{T}(\mathbf{x} \to \mathbf{x}') = 1$. We would like to devise a Markov chain process such that $\Pi^{\text{mc}}(\mathbf{x}) = \Pi(\mathbf{x})$, i.e. that the probability with which a given state $\mathbf{x}$ appears in the chain is equal to desired probability we want to sample from.

An important condition for this to happen is that the probability distribution $\Pi^{\text{mc}}(\mathbf{x})$ is *stationary*, i.e. all states along the chain should be distributed according to the same probability, and this should not change along the chain. A sufficient condition for this to happen is that

$$\Pi(\mathbf{x})\mathcal{T}(\mathbf{x} \to \mathbf{x}') = \Pi(\mathbf{x}')\mathcal{T}(\mathbf{x}' \to \mathbf{x}), \tag{8.27}$$

which is called *detailed balance* equation. This condition basically enforces stationarity (also called micro-reversibility) in the chain: the probability of being in a given state $\mathbf{x}$ and of doing a transition to another state $\mathbf{x}'$ must be equal to the reverse process, starting from $\mathbf{x}'$ and transitioning to $\mathbf{x}$.

8.3.2 The Metropolis-Hastings Algorithm

There exist many possible transition probabilities that satisfy the detailed balance condition (8.27), however the most famous choice is certainly the Metropolis-Hastings prescription. In this case, we separate the transition process into two steps:

$$\mathcal{T}(\mathbf{x} \to \mathbf{x}') \quad = \quad T(\mathbf{x} \to \mathbf{x}')A(\mathbf{x} \to \mathbf{x}'), \tag{8.28}$$

i.e. we first propose a state with some (simple) probability distribution $T(\mathbf{x} \to \mathbf{x}')$ we can easily sample from, and then accept or reject the new state $\mathbf{x}'$ as the next element of the chain with probability $A(\mathbf{x} \to \mathbf{x}')$.

Using the detailed balance condition, we see that the acceptance probability must satisfy:

$$\frac{A(\mathbf{x} \to \mathbf{x}')}{A(\mathbf{x}' \to \mathbf{x})} \quad = \quad \frac{\Pi(\mathbf{x}')}{\Pi(\mathbf{x})} \times \frac{T(\mathbf{x}' \to \mathbf{x})}{T(\mathbf{x} \to \mathbf{x}')}. \tag{8.29}$$

A possible acceptance that satisfies this condition is:

$$A(\mathbf{x} \to \mathbf{x}') \quad = \quad \min\left(1, \frac{\Pi(\mathbf{x}')}{\Pi(\mathbf{x})} \times \frac{T(\mathbf{x}' \to \mathbf{x})}{T(\mathbf{x} \to \mathbf{x}')}\right). \tag{8.30}$$

Notice that this acceptance probability satisfies (8.29), since if $\frac{\Pi(\mathbf{x}')}{\Pi(\mathbf{x})} \times \frac{T(\mathbf{x}' \to \mathbf{x})}{T(\mathbf{x} \to \mathbf{x}')} < 1$ then $\frac{\Pi(\mathbf{x})}{\Pi(\mathbf{x}')} \times \frac{T(\mathbf{x} \to \mathbf{x}')}{T(\mathbf{x}' \to \mathbf{x})} > 1$, $A(\mathbf{x}' \to \mathbf{x}) = 1$ and (8.29) is trivially verified. The same reasoning can be applied for the case $\frac{\Pi(\mathbf{x}')}{\Pi(\mathbf{x})} \times \frac{T(\mathbf{x}' \to \mathbf{x})}{T(\mathbf{x} \to \mathbf{x}')} > 1$.

The Metropolis-Hasting Algorithm can be then summarized in the following steps:

1. Generate a random state $\mathbf{x}'$ drawing from the (simple) transition probability $T(\mathbf{x}^{(i)} \to \mathbf{x}')$.

2. Compute the quantity

$$R \quad = \quad \frac{\Pi(\mathbf{x}')}{\Pi(\mathbf{x}^{(i)})} \times \frac{T(\mathbf{x}' \to \mathbf{x}^{(i)})}{T(\mathbf{x}^{(i)} \to \mathbf{x}')}. \tag{8.31}$$

3. Draw a uniformly distributed random number $\xi \in [0, 1)$.

4. If $R > \xi$, accept the new states, i.e. $\mathbf{x}^{(i+1)} = \mathbf{x}'$. Otherwise, the following state in the chain stays the current one: $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)}$.

Notice that steps 2-4 are necessary to decide whether to accept or reject the proposed state according to the Metropolis probability (8.30).

8.3.3 Estimating Errors and Auto-Correlation Times

Since Markov chains are generated transitioning from a state to the next one, it is natural to expect that adjacent points in the chain will be statistically correlated. To quantify this notion of correlation more precisely, let us first consider the Markov chain estimate for the expectation value of a given function:

$$\bar{g}_{N_s} = \frac{1}{N_s}\sum_{i}^{N_s} g_i, \tag{8.32}$$

where we have used the short-hand $g_i \equiv g(\mathbf{x}^{(i)})$. The law of large numbers states that

$$\lim_{N_s\to\infty} \bar{g}_{N_s} = \sum_{\mathbf{x}} \Pi(\mathbf{x}) g(\mathbf{x}), \tag{8.33}$$

and the central limit theorem says that $\bar{g}_{N_s}$ is a random variable normally distributed,

$$\mathrm{Prob}(\bar{g}_{N_s}) = \mathrm{Normal}(\bar{\mathrm{g}}_\infty, \sigma), \tag{8.34}$$

with expected value $\bar{g}_\infty$ and variance $\sigma^2 = \mathrm{Var_c}[\bar{g}_{N_s}]$, where the variance is computed over different realizations of the Markov chain. It explicitly reads

$$\begin{aligned}
\mathrm{Var_c}[\bar{g}_{N_s}] &= \mathrm{Var_c}\left[\frac{1}{N_s}\sum_{i}^{N_s} g_i\right] \\
&= \frac{1}{N_s^2}\sum_{ij}\mathbb{E}_\mathrm{c}[g_i g_j] - \frac{1}{N_s^2}\sum_{ij}\mathbb{E}_\mathrm{c}[g_i]\mathbb{E}_\mathrm{c}[g_j] \\
&= \frac{1}{N_s}\left(\frac{1}{N_s}\sum_{i}^{N_s}\left(\mathbb{E}_\mathrm{c}[g_i^2] - \mathbb{E}_\mathrm{c}[g_i]^2\right) + \frac{2}{N_s}\sum_{i}^{N_s}\sum_{j=i+1}^{N_s}\left(\mathbb{E}_\mathrm{c}[g_i g_j] - \mathbb{E}_\mathrm{c}[g_i]\mathbb{E}_\mathrm{c}[g_j]\right)\right) \\
&= \frac{1}{N_s}\mathrm{Var_c}(g_0) + 2\sum_{j=1}^{N_s}\left(\mathbb{E}_\mathrm{c}[g_0 g_j] - \mathbb{E}_\mathrm{c}[g_0]\mathbb{E}_\mathrm{c}[g_j]\right)\left(1 - \frac{j}{N_s}\right),
\end{aligned} \tag{8.35}$$

where we assumed that the Markov chain is stationary, thus the variance is independent on the index i, i.e. $\mathrm{Var_c}(g_i) = \mathrm{Var_c}(g_0)$, and the same for the covariance. Now, since all Markov chains we are averaging over generate samples from $\Pi(\mathbf{x})$, computing the variance over the chains is equivalent to computing the variance over $\Pi(\mathbf{x})$, thus $\mathrm{Var_c}(g_0) = \mathrm{Var}_\Pi[g(\mathbf{x})]$ and

$$\mathrm{Var_c}(\bar{g}_{n_s}) = \frac{1}{N_s}\mathrm{Var}_\Pi[g(\mathbf{x})]2\tau_\mathrm{int}, \tag{8.36}$$

having defined the integrated auto-correlation time as

$$\tau_\mathrm{int} = \frac{1}{2} + \frac{1}{\mathrm{Var}_\Pi[g(\mathbf{x})]}\sum_{j=1}^{N_s}\left(\mathbb{E}_\mathrm{c}[g_0 g_j] - \mathbb{E}_\mathrm{c}[g_0]\mathbb{E}_\mathrm{c}[g_j]\right)\left(1 - \frac{j}{N_s}\right). \tag{8.37}$$

We therefore see that unless the Markov chain samples are completely uncorrelated (i.e. $\mathbb{E}_{\mathrm{c}}[g_s g_j] - \mathbb{E}_{\mathrm{c}}[g_s]\mathbb{E}_{\mathrm{c}}[g_j] = 0$) the statistical error on the estimator $\bar{g}_{n_s}$ is increased by the positive factor τ_{int}.

A way to correctly estimate the integrated autocorrelation time is through the correlation function

$$\rho(j) \quad = \quad \frac{\langle g_0 g_j \rangle - \langle g \rangle^2}{\langle g^2 \rangle - \langle g \rangle^2}, \tag{8.38}$$

where $\langle \dots \rangle$ here denote averages over the Markov Chain. A numerically stable estimate of the correlation time is given by

$$\tau_{\mathrm{int}} \quad \simeq \quad \frac{1}{2} + \sum_{j=1}^{j_{\mathrm{cut}}} \rho(j), \tag{8.39}$$

where j_{cut} is chosen for numerical stability as the first j such that $\rho(j_{\max}) < 0$. In practice, given a sequence of estimates $g_1, \dots g_{n_s} = \mathbf{g}$, then the correlation function can be efficiently estimated with a sequence of Fast Fourier Transforms and its inverses:

$$\begin{aligned} A &= FFT(\mathbf{g} - \bar{g}), && (8.40) \\ B &= AA^{\star}, && (8.41) \\ \rho &= \frac{FFT^{-1}(B)}{\langle g^2 \rangle - \langle g \rangle^2}. && (8.42) \end{aligned}$$

8.4 Examples of spin wave functions

In the following we give two simple examples of variational wave functions for many spins, specifically considering the Tranverse-Field Ising model, as also introduced earlier in these lectures. Specifically, we can consider the 1-dimensional hamiltonian

$$\hat{H} = -J \sum_{i=1}^{N} \hat{\sigma}_i^z \hat{\sigma}_{i+1}^z - \Gamma \sum_{i=1}^{N} \hat{\sigma}_i^x. \tag{8.43}$$

8.4.1 Mean-Field Ansatz

We start considering a simple "mean-field" ansatz, that corresponds to taking a wave function that factorizes:

$$|\Psi\rangle \quad = \quad |\Phi_1\rangle \otimes |\Phi_2\rangle \dots |\Phi_N\rangle, \tag{8.44}$$

where

$$\langle s_1 s_2 \dots s_N | \Psi \rangle \quad = \quad \Pi_{i=1}^{N} \Phi_i(s_i). \tag{8.45}$$

In this case, there are $M = 2N$ variational parameters, corresponding to the amplitudes of the single-spin wave functions: $\theta_{2i} = \Phi_i(\uparrow)$ and $\theta_{2i+1} = \Phi_i(\downarrow)$. Without loss of generality, since in this case the Hamiltonian is real, we can take the variational parameters to be real valued.

Since this wave function is very simple, sampling from it can be done even without MCMC, by direct sampling. Most notably, we have that the probability density $\Pi(\mathbf{s}) = |\Psi(\mathbf{s})|^2$ factorizes:

$$\Pi(\mathbf{s}) = \Pi_{i=1}^{N} \Phi_i(s_i)^2, \tag{8.46}$$

thus a simple strategy to sample from it is to generate each spin $s_i = (\uparrow, \downarrow)$ according to the probability:

$$p_i(s) = \frac{\Phi_i(s)^2}{\Phi(\uparrow)^2 + \Phi_i(\downarrow)^2}. \tag{8.47}$$

This is nothing but a Bernoulli distribution for the variable s, and a simple algorithm to sample from it is the following:

1. Draw a uniformly distributed random number $\xi \in [0, 1)$
2. Compute the quantity

$$p_i(\uparrow) = \frac{|\Phi_i(\uparrow)|^2}{|\Phi_i(\uparrow)|^2 + |\Phi_i(\downarrow)|^2}. \tag{8.48}$$

3. If $p_i > \xi$ then $s_i =\uparrow$, otherwise $s_i =\downarrow$.

By repeating this algorithm for all spins s_i, and for all samples N_s, we will generate samples $\mathbf{s}^k = (s_1^k, s_2^k \dots s_N^k)$. Notice that these are all independent samples, at variance with MCMC approaches that instead carry a correlation.

The log derivatives are also easily computed, for example

$$\begin{aligned} D_{2i}(\mathbf{s}) &= \delta_{s_i,\uparrow} \frac{\partial_{\theta_{2i}} \Phi_i(\uparrow)}{\Phi_i(\uparrow)}, && (8.49) \\ &= \frac{\delta_{s_i,\uparrow}}{\Phi_i(\uparrow)}. && (8.50) \end{aligned}$$

8.4.2 Jastrow Ansatz

The mean field ansatz is nice and simple, however it does capture any of the correlations among spins, because of its factorized nature. A systematic way to improve on it is to consider an ansatz of the Jastrow-Feenberg form

$$\begin{aligned} \langle s_1 s_2 \dots s_N | \Psi \rangle = \exp \Bigg[& \sum_i J_i^{(1)}(s_i) + \sum_{i<j} J_{ij}^{(2)}(s_i, s_j) + \\ & + \dots \frac{1}{p!} \sum_{i_1 \neq i_2 \neq \dots i_p} J_{i_1 \dots i_p}^{(p)}(s_{i_1}, s_{i_2} \dots s_{i_p}) \Bigg], \end{aligned} \tag{8.51}$$

where the variational parameters are the functions $J_i^{(1)}(s_i)$,$J_{ij}^{(2)}(s_i, s_j)$,... $J_{i_1 \dots i_p}^{(p)}(s_{i_1}, s_{i_2} \dots s_{i_p})$. This expansion coincides with the mean field ansatz when $p = 1$ (prove it for yourself, assuming the J parameters are not necessarily real-valued), and it is clearly an exact description of the many-body state when $p = N$. The main limitation however is that the number of variational parameters also scales exponentially with p. Nonetheless, in practice one observes convergence to the exact ground-state much sooner, and it is often the case that two-body correlations only are enough to very accurately describe the ground state properties.

For this kind of wave functions, exact sampling is not possible in general, however a good strategy is to perform MCMC. For the TFIM it is commonly chosen a symmetric transition probability :

$$T(\mathbf{s} \to \mathbf{s}') \quad = \quad T(\mathbf{s}' \to \mathbf{s}), \tag{8.52}$$

such that the Metropolis Hastings ratio simply becomes

$$R \quad = \quad \frac{\Pi(\mathbf{s}')}{\Pi(\mathbf{s})}. \tag{8.53}$$

For example a common symmetric transition probability consists in picking a random spin i with uniform probability in $[1, N]$ and then flip it, such that

$$\mathbf{s}' \quad = \quad s_1 \cdots - s_i \ldots s_N. \tag{8.54}$$

Chapter 9

Machine Learning Methods for Many-Body Quantum Systems

Machine learning (ML) techniques are the modern answer to one of the most ancient desires of humanity: devising an Artificial Intelligence capable of independent thinking and problem solving. The applications of ML are ubiquitous, and you might have unconsciously made use of one of those. The tasks that can be solved are for example: language translation, face recognition, game playing. In this Chapter we will discuss the application of machine learning techniques (ML) as a tool to solve the many-body Schroedinger's equation.

9.1 Artificial Neural networks: the machine

There are several pure and applied problems in which the task to be solved whose solution can be expressed as a complex high-dimensional function. For example, let us take the case of identifying faces in a picture. To simplify, let us assume that the picture is monochromatic, and it can be seen, when digitalized, as a a set $\mathbf{x} = b_1, b_2 \ldots b_N$ of bits $b_1 = 0, 1$ (signaling if the given pixel should be black or white). Then, we can ask a machine to give the probability that in a given picture a certain person exists: this can be formulated as: find a function $F_{\text{image}}(\mathbf{x}) = P_{\text{inthere}}(\mathbf{x})$. Analogously, other tasks can be written in this form, like in the case of game playing, where f could be for example the game policy: given the current set of pieces on a board, call it $\mathbf{x}$, find the optimal move that will maximize the chance to win the game. The field of ML is mainly concerned then with the problem of finding compact approximations of such highly-dimensional functions. This goal is achieved through several *learning* paradigms, i.e. ways of using or generating relevant information to find the best functions.

In modern Deep Learning applications, artificial neural networks play a central role. An artificial neural network is nothing but a highly-dimensional function, composition of simple one-dimensional functions, and depending on internal parameters. Taking inspiration from the biology of the brain, the specific functions are taken to be for example of the logistic/sigmoidal form:

$$\phi(x) = \frac{1}{1 + \exp(-x)}.$$

In a feed-forward neural network we would start by taking a linear combination of the input, to define a new set of variables (which defines the first "layer" of the network):

$$a_j^{(1)} = \phi\left(\sum_i w_{ij}^{(1)} x_i + b_j^{(1)}\right),$$

where the matrix $w_{ij}^{(1)}$ and the *bias* vector $b_j^{(1)}$ are arbitrary parameters. Then, we can *feed* those variables into another layer:

$$a_j^{(2)} = \phi\left(\sum_i w_{ij}^{(2)} a_i^{(1)} + b_j^{(2)}\right),$$

and so on, until we reach the final layer:

$$\begin{aligned} a_j^{(D)} &= \phi\left(\sum_i w_{ij}^{(D)} a_i^{(D-1)} + b_j^{(D)}\right) \\ &\equiv [F(\mathbf{x})]_j. \end{aligned}$$

In this sense, the full function can be seen as a nested composition of non-linear vector functions:

$$F(x_1, x_2, \dots x_N) = \phi^{(D)} \circ W^{(D)} \dots \circ \phi^{(2)} \circ W^{(2)} \phi^{(1)} \circ W^{(1)} \mathbf{x}. \tag{9.1}$$

The size of the matrices at each layer are commonly referred to as the *widths* of the neural network, whereas D is referred to as the *depth*.

As a consequence of the Kolmogorov-Arnold representation theorem [1], it can be shown that it is possible to represent (almost) arbitrary high dimensional functions in the form above, provided that the width of the first layer is large enough or that, at fixed width, the depth L is large enough. In the worst case scenario, the width or depth of the network can be exponentially large in N, however in most applications it is found that polynomially large networks constitute excellent approximations. Barron's Theorem further relates the number of nodes in the neural network to the Fourier properties of the function to be approximated. The number of units is of order

$$N_{\text{units}} = O\left(\frac{C_f^2}{\epsilon}\right),$$

where ϵ is the desired error on the functional approximation, and C_f is a constant related to the smoothness of the function to be approximated. This also shows that if a function is smooth enough, a relatively small number of "neurons" (nodes in the network) is sufficient.

9.2 Supervised Learning

With the machine architecture specified by a set of parameters $\boldsymbol{\theta} = \theta_1, \dots \theta_M$ (for example the weights and bias in the previous equations), the remaining task is to find a convenient strategy to numerically determine the weights and the structure of the network. This task is the "learning" aspect of ML. The conceptually simplest approach to learning is the so-called "supervised learning".

In this context we assume that a collection of large, pre-existing data $\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_{N_s}$ exists, such that it is known in advance what the desired output of the machine should be on those, i.e. we also know the associated *labels*: $y_i \equiv F_{\text{ideal}}(\mathbf{x}_i)$. Goal of the supervised learning is to find the best neural-network weights such as $F(\mathbf{x};\boldsymbol{\theta}) \simeq F_{\text{ideal}}(\mathbf{x})$, where the ideal function is known only at the points in the dataset.

In order to quantify the quality of the machine at doing his task, we can define a "loss function":

$$\mathcal{L}(\boldsymbol{\theta}) \;=\; \frac{1}{N_{\text{s}}}\sum_{i}^{N_s} |F(\mathbf{x}_i;\boldsymbol{\theta}) - y_i|^2, \tag{9.2}$$

which attains a minimum value of zero for perfect reconstruction. The supervised learning problem is therefore a conceptually simple inference problem or, alternatively, a non-linear fitting problem.

Learning then amounts to minimizing the loss function with respect to the parameters $\boldsymbol{\theta}$, a task that is typically realized with iterative gradient descent techniques.

9.3 Neural-Network quantum states

Given the ability of artificial neural networks to represent complex high-dimensional functions, in recent years the idea of using them to represent variational wave functions has emerged. This representation is known as "Neural-Network quantum state" (NQS), as introduced in [2].

Formally, we set:

$$\langle \mathbf{x}|\Psi(\boldsymbol{\theta})\rangle = F(\mathbf{x};\theta_1,\dots\theta_M), \tag{9.3}$$

where F is the output of a suitably chosen artificial neural network, depending on a set of parameters $\boldsymbol{\theta}$. For example, in the case of spin 1/2 particles previously introduced, we can choose a neural-network representation $\Psi(s_1,s_2,\dots s_N) = F(s_1,s_2,\dots s_N)$. The only requirement is that the output of the neural network is a (complex-valued, in general) scalar. This can be achieved for example by taking complex-valued weight matrices $W^{(i)}$ and constraining the network widths in such a way that the last layer has a single output.

In practical applications, the most commonly parameterization is not for the amplitudes themselves but rather for the logarithm of the amplitudes, such that:

$$\langle \mathbf{x}|\Psi(\boldsymbol{\theta})\rangle \;=\; \exp\left[F_l(\mathbf{x};\theta_1,\dots\theta_M)\right], \tag{9.4}$$

where $F_l(\boldsymbol{\theta})$ is a feed-forward network.

9.3.1 The loss function

Given the problem of finding the best variational approximation for the ground state of a given hamiltonian $\hat{H}$, we have seen in the previous chapter that the most natural quantity to minimize is the expected valued of the energy, thus we can readily identify it as the loss function:

$$E(\boldsymbol{\theta}) = \frac{\langle\psi(\boldsymbol{\theta})|\hat{H}|\psi(\boldsymbol{\theta})\rangle}{\langle\psi(\boldsymbol{\theta})|\psi(\boldsymbol{\theta})\rangle}. \tag{9.5}$$

In the previous Chapter, we have discussed at length how this expectation value can be estimated using Monte Carlo methods, for example Markov-Chain Monte Carlo, generating N_s samples according to the probability density

$$\Pi(\mathbf{x};\boldsymbol{\theta}) = \frac{|\Psi(\mathbf{x};\boldsymbol{\theta})|^2}{\sum_{\mathbf{x}'}|\Psi(\mathbf{x}';\boldsymbol{\theta})|^2}, \tag{9.6}$$

since

$$\begin{aligned} E(\boldsymbol{\theta}) &= \mathbb{E}_\Pi[E_{\mathrm{loc}}(\mathbf{x})] && (9.7)\\ &\simeq \frac{1}{N_s}\sum_{i=1}^{N_s} E_{\mathrm{loc}}(\mathbf{x}^{(i)}). && (9.8)\end{aligned}$$

9.4 Taking Gradients

An appealing feature of neural networks is that they can be combined with automatic differentiation (AD) techniques. These are approaches to automatically compute (numerically exact) gradients of high-dimensional functions.

9.4.1 The BackPropagation algorithm

In the case of feed-forward neural networks, this is typically achieved through the so called "back-propagation" algorithm.

Let us recall that in a deep network

$$a_j^{(l)} = \phi\left(z_j^{(l)}\right), \tag{9.9}$$

$$z_j^{(l)} = \sum_i w_{ij}^{(l)} a_i^{(l-1)} + b_j^{(l)}, \tag{9.10}$$

with the “boundary” conditions

$$\begin{aligned} z_j^{(0)} &= x_j && (9.11)\\ a^{(D)} &= F(\mathbf{x}). && (9.12)\end{aligned}$$

For simplicity, and since this is the case relevant for NQS, we are focusing here on artificial neural networks with a scalar output, thus the last width is equal to 1 and the index j is omitted: $a_{j=1}^{(D)} \equiv a^{(D)}$.

We then define the *sensitivity* of the neural network output to a change in the weighted input as

$$\Delta_j^{(l)} \equiv \frac{\partial F}{\partial z_j^{(l)}}. \tag{9.13}$$

These sensitivity vectors can be efficiently computed in a recursive way, using the chain rule:

$$\begin{aligned}
\Delta_j^{(l)} &= \frac{\partial F}{\partial z_j^{(l)}} && (9.14)\\
&= \sum_k \frac{\partial F}{\partial z_k^{(l+1)}} \frac{\partial z_k^{(l+1)}}{\partial z_j^{(l)}} && (9.15)\\
&= \sum_k \Delta_k^{(l+1)} \frac{\partial z_k^{(l+1)}}{\partial z_j^{(l)}} && (9.16)\\
&= \sum_k \Delta_k^{(l+1)} \frac{\partial}{\partial z_j^{(l)}} \left(\sum_i w_{ik}^{(l+1)} a_i^{(l)} + b_k^{(l+1)} \right) && (9.17)\\
&= \sum_k \Delta_k^{(l+1)} \frac{\partial}{\partial z_j^{(l)}} \left(\sum_i w_{ik}^{(l+1)} \phi\left(z_i^{(l)} \right) + b_k^{(l+1)} \right) && (9.18)\\
&= \sum_k \Delta_k^{(l+1)} w_{jk}^{(l+1)} \phi'(z_j^{(l)}). && (9.19)
\end{aligned}$$

The key to the back-propagation algorithm is then the ability to compute these sensitivities efficiently, which can be realized also noticing that the sensitivity in the last layer is easy to compute:

$$\begin{aligned}
\Delta_j^{(D)} &= \frac{\partial F}{\partial z_j^{(D)}} && (9.20)\\
&= \phi'\left(z_j^{(D)} \right). && (9.21)
\end{aligned}$$

Thus the algorithm works in two passes:

1. Forward pass: compute and store both $a_j^{(0)}, a_j^{(1)} \dots a_j^{(D)}$ and $z_j^{(1)}, z_j^{(2)} \dots z_j^{(D)}$ using the feed-forward formula (9.1).

2. Backward pass: compute the sensitivities $\Delta_j^{(D)}, \Delta_j^{(D-1)} \dots \Delta_j^{(1)}$ using the recursive formula (9.19)

Finally, the gradients of the neural-network output with respect to the parameters (weights and biases) are simply related to the sensitivities, since

$$\begin{aligned}
\frac{\partial F}{\partial b_j^{(l)}} &= \frac{\partial F}{\partial b_j^{(l)}} \frac{\partial b_j^{(l)}}{\partial z_j^{(l)}} && (9.22)\\
&= \frac{\partial F}{\partial z_j^{(l)}} && (9.23)\\
&= \Delta_j^{(l)}, && (9.24)
\end{aligned}$$

and similarly for gradients with respect to the weights:

$$\begin{aligned}
\frac{\partial F}{\partial w_{ij}^{(l)}} &= \frac{\partial F}{\partial z_j^{(l)}} \frac{\partial z_j^{(l)}}{\partial w_{ij}^{(l)}} && (9.25)\\
&= \Delta_j^{(l)} a_i^{(l-1)}. && (9.26)
\end{aligned}$$

9.4.2 Computing gradients of the energy

The back-propagation algorithm can be readily combined with variational Monte Carlo. Restricting our attention to real-valued wave function, and using the results of the previous Chapter, we can write the gradient of the energy as

$$\begin{aligned}\frac{\partial}{\partial \theta_k} E(\boldsymbol{\theta}) &= \mathbb{E}_{\Pi}[G_k(\mathbf{x})] \\ &= 2\mathbb{E}_{\Pi}\left[\left(E_{\text{loc}}(\mathbf{x}) - \mathbb{E}_{\Pi}[E_{\text{loc}}(\mathbf{x})]\right) D_k(\mathbf{x})\right],\end{aligned}$$

with

$$D_k(\mathbf{x}) = \frac{\partial_{\theta_k}\Psi(\mathbf{x})}{\Psi(\mathbf{x})},$$

If we now take a parameterization $\Psi(\mathbf{x}) = \exp\left[F_l(\mathbf{x};\theta_1,\dots\theta_M)\right]$, then

$$D_k(\mathbf{x}) = \frac{\partial}{\partial \theta_k} F_l(\mathbf{x};\boldsymbol{\theta}),$$

thus

$$\frac{\partial}{\partial \theta_k} E(\boldsymbol{\theta}) \simeq \frac{2}{N_s}\sum_i \left(E_{\text{loc}}(\mathbf{x}^{(i)}) - \mathbb{E}_{\Pi}[E_{\text{loc}}(\mathbf{x})]\right) \frac{\partial}{\partial \theta_k} F_l(\mathbf{x}^{(i)};\boldsymbol{\theta}),$$

and the last term can be estimated efficiently using the back-propagation method.

Bibliography

[1] D. A. Sprecher, “On the structure of continuous functions of several variables”, Transactions Amer. Math. Soc, 115, 340–355,1965.

[2] Giuseppe Carleo, and Matthias Troyer. “Solving the quantum many-body problem with artificial neural networks”. *Science* 355, 602-606, 2017.

Chapter 10

Time-Dependent Variational Principle

10.1 Time-Dependent Variational States

The main goal of time-dependent variational methods is to find the optimal (in a sense to be clarified in the following) description of the time evolution of a state, in terms of a variational state. In practice, we consider a variational state whose parameters, $\boldsymbol{\theta}(t)$, explicitly depend on time t, and such that

$$|\psi\left[\boldsymbol{\theta}(t)\right]\rangle \simeq |\phi(t)\rangle, \tag{10.1}$$

where the state $|\phi(t)\rangle$ is the exact solution to the time-dependent Schrödinger's equation:

$$i\frac{\partial}{\partial t}|\phi(t)\rangle = \hat{H}|\phi(t)\rangle. \tag{10.2}$$

Notice that here we are taking the case of a time-independent Hamiltonian for simplicity, but the discussion of this Chapter can be readily generalized to time-dependent Hamiltonians as well.

In order to make progress in this problem, we can imagine that we start our exact dynamics at time t, from the current variational state, and consider a small time step:

$$\begin{aligned} |\phi(t+\delta_t)\rangle &= |\phi(t)\rangle - i\hat{H}\delta_t|\phi(t)\rangle + \mathcal{O}(\delta_t^2). && (10.3)\\ &= |\psi\left[\boldsymbol{\theta}(t)\right]\rangle - i\hat{H}\delta_t|\psi\left[\boldsymbol{\theta}(t)\right]\rangle + \mathcal{O}(\delta_t^2) && (10.4)\end{aligned}$$

On the other hand, since we have assumed that the variational parameters have a time dependence, we have that

$$|\psi\left(\left[\boldsymbol{\theta}(t+\delta_t)\right]\right)\rangle = |\psi\left[\boldsymbol{\theta}(t)\right]\rangle + \delta_t\sum_k \dot{\theta}_k(t)\partial_{\theta_k}|\psi\left[\boldsymbol{\theta}(t)\right]\rangle. \tag{10.5}$$

In order for the variational state to be as close as possible to the exact dynamics, we then need that

$$|\psi\left[\boldsymbol{\theta}(t+\delta_t)\right]\rangle \simeq |\phi(t+\delta_t)\rangle \tag{10.6}$$

There several ways to impose that Equation (10.6) is approximately verified, giving rise to different time-dependent variational principles. In this Lecture, we consider the variational principle due to McLachlan, that amounts to maximize the distance between the two states appearing on the left and on the right hand side in the equation above.

We can start by considering the distance between the vector changes to the original quantum state:

$$\Delta^2 = \left|\delta_t \sum_k \dot{\theta}_k(t)\partial_{\theta_k}|\psi\left[\boldsymbol{\theta}(t)\right]\rangle + i\delta_t\hat{H}|\psi(\boldsymbol{\theta})\rangle\right|^2. \tag{10.7}$$

This quantity is exactly zero if the time-dependent Schroedinger equation is satisfied exactly by the variational state. In general this is not exactly satisfied, but we can minimize Δ^2 to obtain the best possible variational state that most closely matches the exact dynamics. Since Δ^2 is an explicit function of $\dot{\theta}_k(t)$, we can proceed to minimizing it by computing its gradient:

$$\begin{aligned}\frac{\Delta^2}{\delta_t^2} &= \left|\sum_k \dot{\theta}_k|\partial_{\theta_k}\psi\rangle + i\hat{H}|\psi\rangle\right|^2 && (10.8)\\ &= \sum_{k,k'}\dot{\theta}_k\dot{\theta}_{k'}\langle\partial_{\theta_{k'}}\psi|\partial_{\theta_k}\psi\rangle - i\sum_k\dot{\theta}_k\langle\psi|H|\partial_{\theta_k}\psi\rangle + i\sum_k\dot{\theta}_k\langle\partial_{\theta_k}\psi|H|\psi\rangle + \\ &\quad +\langle\psi|\hat{H}^2|\psi\rangle, && (10.9)\end{aligned}$$

thus

$$\frac{\partial}{\partial\theta_k}\frac{\Delta^2}{\delta_t^2} = \sum_{k'}\dot{\theta}_{k'}\langle\partial_{\theta_{k'}}\psi|\partial_{\theta_k}\psi\rangle - i\langle\psi|H|\partial_{\theta_k}\psi\rangle + \text{c.c.} \tag{10.10}$$

with the stationary point given by

$$\begin{aligned}\sum_{k'}\dot{\theta}_{k'}\text{Re}\left[\langle\partial_{\theta_k}\psi|\partial_{\theta_{k'}}\psi\rangle\right] &= -\text{Re}\left[i\langle\partial_{\theta_k}\psi|H|\psi\rangle\right] && (10.11)\\ &= \text{Im}\langle\partial_{\theta_k}\psi|H|\psi\rangle. && (10.12)\end{aligned}$$

In addition to minimizing this distance, we should also enforce that the resulting state conserves the normalization, since the dynamics is unitary. In practice, this can be either done by adding a suitable Lagrange multiplier, or by considering a variational parameter that takes into account the normalization of the state:

$$|\bar{\psi}(t)\rangle = e^{\theta_0(t)}|\psi(t)\rangle. \tag{10.13}$$

I leave as an exercise to shown that this leads to

$$\sum_{k'} S^R_{kk'}\dot{\theta}_{k'}(t) = C^I_k, \tag{10.14}$$

with the superscript R and I denoting the real and imaginary part, respectively, of the following quantities:

$$\begin{aligned}C_k &= \frac{\langle\partial_{\theta_k}\psi|\hat{H}|\psi\rangle}{\langle\psi|\psi\rangle} - \frac{\langle\partial_{\theta_k}\psi|\psi\rangle}{\langle\psi|\psi\rangle}\frac{\langle\psi|\hat{H}|\psi\rangle}{\langle\psi|\psi\rangle} && (10.15)\\ S_{kk'} &= \frac{\langle\partial_{\theta_k}\psi|\partial_{\theta_{k'}}\psi\rangle}{\langle\psi|\psi\rangle} - \frac{\langle\partial_{\theta_k}\psi|\psi\rangle\langle\psi|\partial_{\theta_{k'}}\psi\rangle}{\langle\psi|\psi\rangle^2}. && (10.16)\end{aligned}$$

The latter quantity is also known in literature as the “quantum geometric tensor” and plays a fundamental role in setting the metric of associated with variational states.

10.2 Imaginary-time evolution

Up to now, we have considered solutions to the "standard" time-dependent Schrödinger equation, for the evolution of a quantum state in the physical time t. A rather important tool in computational quantum physics is found solving the Schrödinger equation in the so-called imaginary-time $\tau = it$. Let us consider again the case of static Hamiltonian, then in this case the imaginary-time evolved quantum state reads:

$$|\Psi(\tau)\rangle = e^{-\hat{H}\tau} |\Psi(0)\rangle . \tag{10.17}$$

We can immediately notice an important difference with respect to the real (or physical)-time evolution: in the imaginary-time case, the evolution is no longer unitary and it can, for example, systematically change the norm of our initial state. One of the reasons why imaginary-time evolution is important, is that it can be used as an alternative scheme to find the ground-state of a given hamiltonian H. To show this, we use the spectral decomposition of the initial state in terms of the eigenstates of the Hamiltonian, $|\phi_k\rangle$ of energy E_k :

$$|\Psi(\tau)\rangle = \sum_k e^{-E_k\tau} c_k |\phi_k\rangle . \tag{10.18}$$

We also imagine that we have sorted the eigenstates in ascending order with respect to the energy, such as that $E_0 < E_1 < E_2 \ldots$, and define the energy differences $\Delta E_k = E_k - E_0 \geq 0$. We then rewrite

$$|\Psi(\tau)\rangle = e^{-E_0\tau} \left[c_0 |\phi_0\rangle + \sum_{k>0} e^{-\Delta E_k\tau} c_k |\phi_k\rangle \right] . \tag{10.19}$$

When the initial state is non-orthogonal to the exact ground-state (i.e. when $|c_0| \neq 0$), the imaginary-time evolution converges to the exact ground state. This can be derived immediately from the last expression, since in the limit $\tau \gg \Delta E_0$ the right-hand-side vanished exponentially. Therefore, apart from an arbitrary normalization, we converge to the exact ground-state of the system, namely : $|\Psi(\tau)\rangle \simeq |\phi_0\rangle$. Imaginary-time evolution can be then used as an alternative approach to find the exact ground-state of the system, and can be implemented using the same methods presented before for the real-time evolution.

10.2.1 Variational Imaginary Time evolution

The discussion on the time dependent variational principle can be extended also to the case of imaginary time dynamics, and offers an alternative approach to find the variational ground state of a given Hamiltonian. The main difference is that the variational parameters are now taken to be dependent on the imaginary time $\tau$and we aim at finding optimal variational parameters derivatives, such that

$$|\psi[\boldsymbol{\theta}(\tau + \delta_\tau)]\rangle \ \ \simeq \ \ |\phi(t + \delta_\tau)\rangle \tag{10.20}$$

The optimal equation of the solved in this case is found by minimizing the distance between these two states:

$$|I\rangle = |\psi(\boldsymbol{\theta})\rangle - \delta_\tau \hat{H}|\psi(\boldsymbol{\theta})\rangle \tag{10.21}$$

$$|II\rangle = |\psi(\boldsymbol{\theta})\rangle + \delta_\tau \sum_k \dot{\theta}_k(\tau)\partial_{\theta_k}|\psi(\boldsymbol{\theta})\rangle, \tag{10.22}$$

which yields a very similar equation of motion:

$$\sum_{k'} S^R_{kk'}\dot{\theta}_{k'}(\tau) = -C^R_k, \tag{10.23}$$

the main difference being that the right hand side now contains the real part of C_k.

10.3 The Dirac-Frenkel variational principle

The time-dependent variational principle derived before (due to McLachlan) is valid for arbitrary states containing real-valued parameters, $\theta(t)$. In the special case in which the parameters are complex valued instead, we can still solve the equations of motions as written before, just considering twice as many variational parameters, corresponding to the real and imaginary part of each complex-valued parameter $\theta(t) = \theta^R + i\theta^I$. In this sense, the McLachlan variational principle is very general. However, there is an important family of variational states that depend on complex parameters and that are holomorphic (complex differentiable). In this context, this implies that the following Cauchy-Riemann conditions for the wave functions amplitudes derivatives are verified:

$$\frac{\partial\psi^R(\mathbf{x};\theta)}{\partial\theta^R_k} = \frac{\partial\psi^I(\mathbf{x};\theta)}{\partial\theta^I_k}, \tag{10.24}$$

$$\frac{\partial\psi^R(\mathbf{x};\theta)}{\partial\theta^I_k} = -\frac{\partial\psi^I(\mathbf{x};\theta)}{\partial\theta^R_k}. \tag{10.25}$$

Famous examples of holomorphic functions are polynomials, exponentials etc. In this case, instead of considering the McLachlan equations of motion with twice as many (real-valued) variational parameters, we can exploit holomorphicity to reduce the equations of motion to the following form (in the real time evolution case):

$$\sum_{k'} S_{kk'}\dot{\theta}_{k'}(t) = -iC_k. \tag{10.26}$$

Notice that in this case both the LHS matrix S and the vector C are complex valued, as well as the variational parameters derivatives. This version of the time dependent variational principle was actually discovered significantly before than the (more general) McLachlan case, and is traditionally attributed to Dirac and Frenkel.

10.4 Time-Dependent Variational Monte Carlo

In order to implement the time-dependent variational principle (both in real and imaginary time) with generic variational states, we can use an extension of Variational Monte Carlo. To simplify the equations, we start noticing that we can write the variational derivatives in terms of $D_k(\mathbf{x};\boldsymbol{\theta}) = \frac{\partial_{\theta_k}\Psi(\mathbf{x};\boldsymbol{\theta})}{\Psi(\mathbf{x};\boldsymbol{\theta})}$ and its associated diagonal operator $\hat{D}_k$, such that:

$$\begin{aligned}\partial_{\theta_k}\Psi(\mathbf{x};\boldsymbol{\theta}) &= \frac{\partial_{\theta_k}\Psi(\mathbf{x};\boldsymbol{\theta})}{\Psi(\mathbf{x};\boldsymbol{\theta})}\Psi(\mathbf{x};\boldsymbol{\theta}) && (10.27)\\ &= D_k(\mathbf{x};\boldsymbol{\theta})\Psi(\mathbf{x};\boldsymbol{\theta}) && (10.28)\end{aligned}$$

and

$$|\partial_{\theta_k}\psi(\boldsymbol{\theta})\rangle = \hat{D}_k|\psi(\boldsymbol{\theta})\rangle. \qquad (10.29)$$

We therefore have that the metric tensor can be estimated as an average of operators, and estimated as the covariance of the logarithmic derivatives over the Born distribution:

$$\begin{aligned}S_{kk'} &= \frac{\langle\psi|\hat{D}_k^\dagger\hat{D}_{k'}|\psi\rangle}{\langle\psi|\psi\rangle} - \frac{\langle\psi|\hat{D}_k^\dagger|\psi\rangle}{\langle\psi|\psi\rangle}\frac{\langle\psi|\hat{D}_{k'}|\psi\rangle}{\langle\psi|\psi\rangle} && (10.30)\\ &= \mathbb{E}_\Pi[D_k^\star(\mathbf{x})D_{k'}(\mathbf{x})] - \mathbb{E}_\Pi[D_k^\star(\mathbf{x})]\mathbb{E}_\Pi[D_{k'}(\mathbf{x})]. && (10.31)\end{aligned}$$

On the other hand the vector C can be estimated noticing that

$$\begin{aligned}\frac{\langle\partial_{\theta_k}\psi|\hat{H}|\psi\rangle}{\langle\psi|\psi\rangle} &= \frac{\sum_{\mathbf{x}}\langle\psi|\hat{D}_k^\dagger|\mathbf{x}\rangle\langle\mathbf{x}|\hat{H}|\psi\rangle}{\langle\psi|\psi\rangle} && (10.32)\\ &= \frac{\sum_{\mathbf{x}}|\psi(\mathbf{x})|^2 D_k^\star(\mathbf{x})\frac{\langle\mathbf{x}|\hat{H}|\psi\rangle}{\langle\mathbf{x}|\psi\rangle}}{\langle\psi|\psi\rangle} && (10.33)\\ &= \mathbb{E}_\Pi[D_k^\star(\mathbf{x})E_{\mathrm{loc}}(\mathbf{x})], && (10.34)\end{aligned}$$

thus it also takes the form of a statistical covariance:

$$C_k = \mathbb{E}_\Pi[D_k^\star(\mathbf{x})E_{\mathrm{loc}}(\mathbf{x})] - \mathbb{E}_\Pi[D_k^\star(\mathbf{x})]\mathbb{E}_\Pi[E_{\mathrm{loc}}(\mathbf{x})]. \qquad (10.35)$$

By comparing this with the results in the previous Lectures, we also notice that the real part of this vector is proportional to the energy gradient:

$$\frac{\partial}{\partial\theta_k}E(\boldsymbol{\theta}) = \frac{1}{2}C_k^R. \qquad (10.36)$$

10.4.1 Overall Algorithm

We summarize now the time-dependent Variational Monte Carlo in the general case of McLachlan variational principle. First off, we decided wether we want to simulate real or imaginary time dynamics. Then, the choice made, we go through the following steps:

0. Start the time evolution by initializing the variational parameters at some given values $\boldsymbol{\theta}(0)$. For example, these could random values, or could be parameters corresponding to a given state.

Then, at each time step:

1. Generate random samples $\mathbf{x}^{(1)} \dots \mathbf{x}^{(N_s)}$ drawing from the probability density $\Pi(\mathbf{x}) \propto |\psi(\mathbf{x};\boldsymbol{\theta}(t))|^2$.
2. Compute the quantities $S_{kk'}$ and C_k as averages over this samples using the expressions above
3. If doing real time evolution, solve the linear system:

$$\sum_{k'} S^R_{kk'} \dot{\theta}_{k'}(t) \;\; = \;\; C^I_k, \qquad (10.37)$$

for the vector of unknowns $\dot{\theta}_k(t)$. If doing imaginary time evolution, solve instead the linear system:

$$\sum_{k'} S^R_{kk'} \dot{\theta}_{k'}(\tau) \;\; = \;\; -C^R_k \qquad (10.38)$$

for the vector of unknowns $\dot{\theta}_k(\tau)$.

4. Use the time derivatives to update the parameters, for example with a simple Euler scheme:

$$\theta_k(t+\delta_t) \;\; = \;\; \theta_k(t) + \delta_t \dot{\theta}_k(t). \qquad (10.39)$$

In practice, in the last step one rarely uses the simple Euler scheme, because it would require very small time steps in order to get a stable trajectory. More often, higher-order integration schemes such as Runge-Kutta are employed. A noticeable exception is the case of imaginary time evolution, where it is often the case that the simple Euler scheme is preferred, giving rise to a method known as "Stochastic reconfiguration", that is commonly adopted as an alternative to stochastic gradient descent to find the variational ground state of a given Hamiltonian.

10.5 Example: dynamics of a mean-field variational state

As a relatively simple example of the formalism developed above, let us consider the case of 2 spins 1/2 on a lattice evolving according to the usual transverse-field Ising hamiltonian:

$$\hat{H} \;\; = \;\; -\Gamma\left(\hat{\sigma}^x_1 + \hat{\sigma}^x_2\right) - V\hat{\sigma}^z_1\hat{\sigma}^z_z, \qquad (10.40)$$

and we consider as an initial state

$$|\phi(0)\rangle \;\; = \;\; |\downarrow\rangle_1 \otimes |\downarrow\rangle_2. \qquad (10.41)$$

This problem is small enough that it can be solved exactly, however it is instructive to look at how variational dynamics works. We will consider a time-dependent variational ansatz of the form

$$|\psi(t)\rangle = e^{\theta(t)\left(\hat{\sigma}_1^x+\hat{\sigma}_2^x\right)}|\phi(0)\rangle, \tag{10.42}$$

where the variational parameter $\theta(t)$ is taken complex valued and it is to be determined by solving the corresponding equations of motion. The explicit form of the variational state is by construction factorized

$$|\psi(t)\rangle = |\Phi(t)\rangle_1 \otimes |\Phi(t)\rangle_2, \tag{10.43}$$

with

$$|\Phi(t)\rangle_k = \cosh(\theta(t))|\downarrow\rangle_k + \sinh(\theta(t))|\uparrow\rangle_k. \tag{10.44}$$

Because of the simple variational form, we can also determine the variational derivatives easily.

$$|\partial_\theta\psi(t)\rangle = \left(\hat{\sigma}_1^x + \hat{\sigma}_2^x\right)|\psi(t)\rangle. \tag{10.45}$$

Since the ansatz contains a complex parameter, $\theta(t)$, and its amplitudes are holomorphic (prove it!), we can solve the Dirac-Frenkel equations of motion, Eq. (10.26). The matrix S in this case is then just a scalar and reads:

$$\langle\partial_\theta\psi(t)|\partial_\theta\psi(t)\rangle - |\langle\psi(t)|\partial_\theta\psi(t)\rangle|^2 = 2 + 2\langle\hat{\sigma}_1^x\rangle\langle\hat{\sigma}_2^x\rangle - |\langle\hat{\sigma}_1^x\rangle + \langle\hat{\sigma}_2^x\rangle|^2 \tag{10.46}$$

$$= 2 - 2\langle\hat{\sigma}_1^x\rangle^2. \tag{10.47}$$

Analogous equations can be derived for the vector C (which is again just a scalar in this case). We leave as an exercise the task of solving the equations of motion and comparing the variational dynamics to the exact one.

In general, one can already expect that if the interactions are small, the variational dynamics with this simple mean-field ansatz should be accurate.

Bibliography

[1] Becca, F., & Sorella, S. (2017). Quantum Monte Carlo Approaches for Correlated Systems. Cambridge: Cambridge University Press. doi:10.1017/9781316417041

Chapter 11

Ground-State Quantum Monte Carlo

In the previous lectures we have started our journey through Quantum Monte Carlo methods, introducing the basis concept of stochastic sampling, as well as the Variational Monte Carlo approach. Even though Variational Monte Carlo is a very powerful method, in principle it can recover the *exact* ground-state solution only if a sufficiently general enough wave-function ansatz is used.

During this Lecture we will see how we can find, in some notable cases, the exact ground-state solution using a different set of Quantum Monte Carlo techniques.

11.1 Imaginary-Time Evolution

In order to find an exact (and computationally useful) representation of the exact ground-state wave-function, we start from the imaginary-time evolution of a some given initial state:

$$\langle \mathbf{x} | \Psi(\tau) \rangle \;\; = \;\; \langle \mathbf{x} | e^{-\tau \hat{H}} | \Psi(0) \rangle, \tag{11.1}$$

where $\hat{H}$ is the Hamiltonian, τ is the imaginary-time, and $|\Psi(0)\rangle$ is some chosen initial state (for example it might be the best possible variational estimate we have for the ground-state of $\hat{H}$). We have seen already previously that imaginary-time evolution converges to the exact ground-state $\langle \mathbf{x} | \Psi_0 \rangle$ in the limit $\tau \gg \Delta E_0 = E_1 - E_0$, provided that the initial state is non-orthogonal to the exact one i.e. if $|\langle \Psi_0 \, | \Psi(0) \rangle|^2 \neq 0$. A complete demonstration of this statement has been given in one of the previous chapters.

Goal of today's lecture is to devise a way to sample exactly, using Monte Carlo methods, from $|\langle \mathbf{x} | \Psi(\tau) \rangle|^2$. If we achieve this goal, then all the properties of interest can be measured using just statistical averages, as already discussed in the case of the Variational Monte Carlo approach.

11.1.1 Path-Integral Representation

In order to devise a practical sampling scheme, we need to write $\langle \mathbf{x} | \Psi(\tau) \rangle$ in a way that is more suitable for computations. The first step is to divide the total imaginary-time

τ into P small time-steps, such that $\tau = P \times \Delta_\tau$, and use the fact that

$$e^{-\tau\hat{H}} = \underbrace{e^{-\Delta_\tau\hat{H}}\dots e^{-\Delta_\tau\hat{H}}}_{P\text{ times}}. \tag{11.2}$$

We then derive the path-integral representation of the imaginary time-evolved state as:

$$\begin{aligned}
\langle\mathbf{x}|\Psi(\tau)\rangle &= \langle\mathbf{x}|e^{-\Delta_\tau\hat{H}}\dots e^{-\Delta_\tau\hat{H}}|\Psi(0)\rangle \\
&= \sum_{\mathbf{x}_0}\langle\mathbf{x}|\underbrace{e^{-\Delta_\tau\hat{H}}\dots e^{-\Delta_\tau\hat{H}}}_{P\text{ times}}|\mathbf{x}_0\rangle\,\langle\mathbf{x}_0\,|\Psi(0)\rangle \\
&= \sum_{\mathbf{x}_0}\sum_{\mathbf{x}_1}\langle\mathbf{x}|\underbrace{e^{-\Delta_\tau\hat{H}}\dots e^{-\Delta_\tau\hat{H}}}_{P-1\text{ times}}|\mathbf{x}_1\rangle\langle\mathbf{x}_1|e^{-\Delta_\tau\hat{H}}|\mathbf{x}_0\rangle\,\langle\mathbf{x}_0\,|\Psi(0)\rangle \\
&= \sum_{\mathbf{x}_0\mathbf{x}_1\dots\mathbf{x}_{P-1}}\langle\mathbf{x}|e^{-\Delta_\tau\hat{H}}|\mathbf{x}_{P-1}\rangle\dots\langle\mathbf{x}_1|e^{-\Delta_\tau\hat{H}}|\mathbf{x}_0\rangle\,\langle\mathbf{x}_0\,|\Psi(0)\rangle,
\end{aligned} \tag{11.3}$$

where we have inserted P completeness relations $\sum_{\mathbf{x}_j}|\mathbf{x}_j\rangle\langle\mathbf{x}_j| = \hat{I}$, and thus introduced a set of P imaginary-time dependent quantum numbers $\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_P$. We now also define the *propagator*:

$$G^{\Delta_\tau}(\mathbf{x},\mathbf{y}) \equiv \langle\mathbf{x}|e^{-\Delta_\tau\hat{H}}|\mathbf{y}\rangle, \tag{11.4}$$

and use the naming conventions $\mathbf{x} \equiv \mathbf{x}_P$, and $\langle\mathbf{x}_0\,|\Psi(0)\rangle \equiv \Psi(\mathbf{x}_0)$, thus we have

$$\langle\mathbf{x}|\Psi(\tau)\rangle = \sum_{\mathbf{x}_0\mathbf{x}_1\dots\mathbf{x}_{P-1}}\underbrace{G^{\Delta_\tau}(\mathbf{x}_P,\mathbf{x}_{P-1})\dots G^{\Delta_\tau}(\mathbf{x}_1,\mathbf{x}_0)}_{P\text{ times}}\Psi(\mathbf{x}_0). \tag{11.5}$$

The representation (11.5) is particularly useful because we typically know how to compute controlled numerical approximations of the short-time propagators (11.4).

For example, in most applications the Hamiltonian is the sum of two non-commuting terms $\hat{H} = \hat{H}_0 + \hat{H}_1$, where the first term is diagonal in the chosen basis $|\mathbf{x}\rangle$. We can then use the Trotter decomposition to find:

$$\begin{aligned}
G^{\Delta_\tau}(\mathbf{x},\mathbf{y}) &= \langle\mathbf{x}|e^{-\Delta_\tau\hat{H}_0}e^{-\Delta_\tau\hat{H}_1}|\mathbf{y}\rangle + \mathcal{O}(\Delta_\tau^2) \\
&= e^{-\Delta_\tau H_0(\mathbf{x})}\langle\mathbf{x}|e^{-\Delta_\tau\hat{H}_1}|\mathbf{y}\rangle + \mathcal{O}(\Delta_\tau^2) \\
&= e^{-\Delta_\tau H_0(\mathbf{x})}G_1^{\Delta_\tau}(\mathbf{x},\mathbf{y}) + \mathcal{O}(\Delta_\tau^2),
\end{aligned} \tag{11.6}$$

or the symmetric second-order approximation:

$$G^{\Delta_\tau}(\mathbf{x},\mathbf{y}) = e^{-\Delta_\tau\frac{H_0(\mathbf{x})}{2}}G_1^{\Delta_\tau}(\mathbf{x},\mathbf{y})e^{-\Delta_\tau\frac{H_0(\mathbf{y})}{2}} + \mathcal{O}(\Delta_\tau^3). \tag{11.7}$$

The important aspect of these decomposition is that, in interesting applications, it is typically possible to compute $G_1^{\Delta_\tau}(\mathbf{x},\mathbf{y})$ exactly, since the Hamiltonian H_1 can be efficiently diagonalized.

11.2 Particles in continuous space

Let us now give a specific example of the path-integral representation, and consider a system of interacting particles in continuous space, for which we consider again the general Hamiltonian:

$$\hat{H} = -\frac{\hbar^2}{2m}\sum_i^N \nabla^2_{\vec{r}_i} + \sum_i V_1(\vec{r}_i) + \sum_{i<j} V_2(\vec{r}_i,\vec{r}_j), \tag{11.8}$$

where V_1 and V_2 are generic one and two-body interaction potential. In this case, the computational basis denotes all the particle positions, i.e. $|\mathbf{x}\rangle = |\vec{r}_1, \vec{r}_2, \dots \vec{r}_N\rangle$ and it is therefore a vector of N 3-dimensional coordinates. On top of that, we must consider the additional imaginary-time index, i.e. a full state in the path-integral is represented by all the quantum numbers $|\mathbf{x}_j\rangle = |\vec{r}_{j1}, \vec{r}_{j2}, \dots \vec{r}_{jN}\rangle$, where for each particle we must specify its position in space at all the discrete times $j = 0, 1 \dots P$. We therefore see that this representation corresponds to effectively having N particles living in a 4-dimensional space (with the extra dimension coming from the imaginary time direction).

11.2.1 The Propagator

For the Hamiltonian (12.13), it is easy to derive a short-time propagator using the Trotter decomposition. In particular, call $H_0(\mathbf{X}) = \sum_i V_1(\vec{r}_i) + \sum_{i<j} V_2(\vec{r}_i,\vec{r}_j)$, the part of the Hamiltonian diagonal in the computational basis, and $\hat{H}_1 = -\frac{\hbar^2}{2m}\sum_i^N \nabla^2_{\vec{r}_i}$, the non-commuting kinetic energy. We then have that the free-particle propagator is:

$$\begin{aligned}
G_1^{\Delta_\tau}(\mathbf{x},\mathbf{y}) &= \langle \mathbf{x}| e^{-\Delta_\tau \sum_i^N \frac{\vec{P}_i^2}{2m}} |\mathbf{y}\rangle \\
&= \Pi_{i=1}^N \langle \vec{r}_i | e^{-\Delta_\tau \frac{\vec{P}_i^2}{2m}} | \vec{r}^{\,\prime}_i\rangle \\
&= \Pi_{i=1}^N \Pi_\alpha^{\{x,y,z\}} \langle r_{\alpha i} | e^{-\Delta_\tau \frac{\hat{P}^2_{\alpha i}}{2m}} | r'_{\alpha i}\rangle \\
&= \Pi_{i=1}^N \Pi_\alpha^{\{x,y,z\}} \sum_n \langle r_{\alpha i} | e^{-\Delta_\tau \frac{\hat{P}^2_{\alpha i}}{2m}} | P_{\alpha i}(n)\rangle \langle P_{\alpha i}(n) | r'_{\alpha i}\rangle \\
&= \frac{1}{L^{3N}} \Pi_{i=1}^N \Pi_\alpha^{\{x,y,z\}} \sum_n e^{-\Delta_\tau \frac{P^2_{\alpha i}(n)}{2m}} e^{i\frac{P_{\alpha i}(n)}{\hbar}(r_{\alpha i} - r'_{\alpha i})} \\
&= \left(\frac{m}{2\pi\hbar^2\Delta_\tau}\right)^{3N/2} \Pi_{i=1}^N \Pi_\alpha^{\{x,y,z\}} \exp\left[-\frac{(r_{\alpha i} - r'_{\alpha i})^2}{2\Delta_\tau\hbar^2/m}\right].
\end{aligned} \tag{11.9}$$

where $P_\alpha(n) = \hbar\frac{2\pi}{L}n_\alpha$ are the eigenvalues of the momentum operator along the cartesian coordinate α, $\langle r_{\alpha i}|P_{\alpha i}(n)\rangle = e^{i\frac{P_{\alpha i}(n)}{\hbar}r_{\alpha i}}/\sqrt{L}$ are the corresponding eigenfunctions. Notice that here we are assuming that we are dealing with a system having periodic boundary conditions in a 3 dimensional box of size $L \times L \times L$, thus yielding the quantization of the momentum $P_\alpha(n)$. The last line in the equation above is instead obtained approximating the discrete sum by a continuous integral, yielding the inverse Fourier transform of a gaussian function.

For a single, one-dimensional particle, the free propagator is the simple Gaussian:

$$G_1^{\Delta_\tau}(x, y) \quad = \quad \left(\frac{m}{2\pi\hbar^2\Delta_\tau}\right)^{1/2} \exp\left[-\frac{(x-y)^2}{2\Delta_\tau\hbar^2/m}\right]. \tag{11.10}$$

11.3 Path-Integral-Ground-State Monte Carlo

We have derived a representation of the wave-function in terms of sums over a path of configurations. Quantum Monte Carlo techniques based on this representation aim at sampling the exact ground-state wave-function, and in particular at sampling $|\langle \mathbf{x}|\Psi(\tau)\rangle|^2$, which can then be used to compute all the properties of the ground state, with an approach similar to the Variational method. We restrict now ourselves to real-valued wave-functions, for which we have:

$$\langle \mathbf{x}|\Psi(\tau)\rangle^2 = \sum_{\{\mathbf{x}_j,\mathbf{x'}_j\}\neq \mathbf{x}_P} \Psi(\mathbf{x}_0) G^{\Delta_\tau}(\mathbf{x}_0,\mathbf{x}_1)\dots G^{\Delta_\tau}(\mathbf{x}_{P-1},\mathbf{x})\times \\ \times G^{\Delta_\tau}(\mathbf{x},\mathbf{x'}_{P-1})\dots G^{\Delta_\tau}(\mathbf{x'}_1,\mathbf{x'}_0)\Psi(\mathbf{x}'_0), \quad (11.11)$$

or, doubling the number of imaginary time slices:

$$\langle \mathbf{x}|\Psi(\tau)\rangle^2 = \sum_{\{\mathbf{x}_j\}\neq \mathbf{x}_P} \Psi(\mathbf{x}_0) G^{\Delta_\tau}(\mathbf{x}_0,\mathbf{x}_1)\dots G^{\Delta_\tau}(\mathbf{x}_{P-1},\mathbf{x}_P)\times \\ \times G^{\Delta_\tau}(\mathbf{x}_P,\mathbf{x}_{P+1})\dots G^{\Delta_\tau}(\mathbf{x}_{2P-1},\mathbf{x}_{2P})\Psi(\mathbf{x}_{2P}). \quad (11.12)$$

11.3.1 Statistical estimators of quantum expectation values

The expression (11.12) can be immediately used to estimate expectation values of some given observable. Let us concentrate for simplicity on diagonal observables, $O_{\mathbf{x}\mathbf{x}'} = \delta_{\mathbf{x}\mathbf{x}'} O(\mathbf{x})$, for which we have

$$\begin{aligned}\frac{\langle\Psi(\tau)|\hat{O}|\Psi(\tau)\rangle}{\langle\Psi(\tau)|\Psi(\tau)\rangle} &= \frac{\sum_{\mathbf{x}}\langle\mathbf{x}|\Psi(\tau)\rangle^2 O(\mathbf{x})}{\sum_{\mathbf{x}}\langle\mathbf{x}|\Psi(\tau)\rangle^2} \\ &= \frac{\sum_{\{\mathbf{x_j}\}}\Pi(\mathbf{x}_0,\dots,\mathbf{x}_{2P})O(\mathbf{x}_P)}{\sum_{\{\mathbf{x_j}\}}\Pi(\mathbf{x}_0,\dots,\mathbf{x}_{2P})},\end{aligned} \quad (11.13)$$

where we have defined

$$\Pi(\mathbf{x}_0,\dots,\mathbf{x}_{2P}) = \Psi(\mathbf{x}_0) G^{\Delta_\tau}(\mathbf{x}_0,\mathbf{x}_1)\dots G^{\Delta_\tau}(\mathbf{x}_{P-1},\mathbf{x}_P)\times \\ \times G^{\Delta_\tau}(\mathbf{x}_P,\mathbf{x}_{P+1})\dots G^{\Delta_\tau}(\mathbf{x}_{2P-1},\mathbf{x}_{2P})\Psi(\mathbf{x}_{2P}). \quad (11.14)$$

We immediately see that if we want to interpret the quantum expectation value (11.13), then the quantity $\Pi(\mathbf{x}_0,\dots,\mathbf{x}_{2P})$ must have the meaning of a probability density, such that

$$\frac{\langle\Psi(\tau)|\hat{O}|\Psi(\tau)\rangle}{\langle\Psi(\tau)|\Psi(\tau)\rangle} = \mathbb{E}_{\Pi}[O(\mathbf{x}_P)], \quad (11.15)$$

where $\mathbb{E}_{\Pi}[\dots]$ denote *statistical* expectation values over the probability distribution $\Pi(\mathbf{x}_0,\dots,\mathbf{x}_{2P})$ of the $2P+1$ many-body particle coordinates.

An analogous expression can be derived for the estimator of the energy, using the fact that the Hamiltonian commutes with the imaginary-time evolution:

$$\begin{aligned}
\frac{\langle\Psi(\tau)|\hat{H}|\Psi(\tau)\rangle}{\langle\Psi(\tau)|\Psi(\tau)\rangle} &= \frac{\langle\Psi(\tau)|e^{-\tau\hat{H}}\hat{H}|\Psi(0)\rangle}{\langle\Psi(\tau)|\Psi(\tau)\rangle} \\
&= \frac{\sum_{\{\mathbf{x_j}\}}\Pi(\mathbf{x}_0,\ldots,\mathbf{x}_{2P})\frac{\langle\mathbf{x}_{2P}|\hat{H}|\Psi\rangle}{\langle\mathbf{x}_{2P}|\Psi\rangle}}{\sum_{\{\mathbf{x_j}\}}\Pi(\mathbf{x}_0,\ldots,\mathbf{x}_{2P})} \\
&= \frac{\sum_{\{\mathbf{x_j}\}}\Pi(\mathbf{x}_0,\ldots,\mathbf{x}_{2P})E_{\text{loc}}(\mathbf{x}_{2P})}{\sum_{\{\mathbf{x_j}\}}\Pi(\mathbf{x}_0,\ldots,\mathbf{x}_{2P})} \\
&= \mathbb{E}_{\Pi}[E_{\text{loc}}(\mathbf{x}_{2P})], \qquad (11.16)
\end{aligned}$$

where we have used the local energy, previously introduced when discussing the Variational Monte Carlo. Slightly reduced statistical fluctuations can be obtained taking the symmetrized estimator:

$$\begin{aligned}
\frac{\langle\Psi(\tau)|\hat{H}|\Psi(\tau)\rangle}{\langle\Psi(\tau)|\Psi(\tau)\rangle} &= \frac{1}{2}\left(\frac{\langle\Psi(\tau)|\hat{H}e^{-\tau\hat{H}}|\Psi(0)\rangle}{\langle\Psi(\tau)|\Psi(\tau)\rangle} + \frac{\langle\Psi(\tau)|e^{-\tau\hat{H}}\hat{H}|\Psi(0)\rangle}{\langle\Psi(\tau)|\Psi(\tau)\rangle}\right) \\
&= \frac{1}{2}\left(\mathbb{E}_{\Pi}[E_{\text{loc}}(\mathbf{x}_0) + E_{\text{loc}}(\mathbf{x}_{2P})]\right), \qquad (11.17)
\end{aligned}$$

i.e. taking the averages of the local energies at the ends of the path, yielding a reduction in the statistical error on the average of about a factor of $1/\sqrt{2}$.

11.3.1.1 Sign Problem

The exact ground-state properties (Eqs. 11.15,11.16) have the meaning of statistical averages over a probability distribution whenever $\Pi(\mathbf{x}_0\ldots\mathbf{x}_{2P}) \geq 0$ for all the possible values of the path configurations. It is easy to show that for particles in continuous space the sign of $\Pi(\mathbf{x}_0\ldots\mathbf{x}_{2P})$ is given only by the sign of the product of the wave-function at the ends of the path: $\text{sign}(\Pi(\mathbf{x}_0\ldots\mathbf{x}_{2P})) = \text{sign}\left[\Psi(\mathbf{x}_0)\times\Psi(\mathbf{x}_{2P})\right]$.

For bosons, it can be shown that the exact ground state wave-function satisfies $\Psi_0(\mathbf{x}) \geq 0$. If we want that $|\langle\Psi_0|\Psi(0)\rangle|^2 \neq 0$, this also implies that the wave function at $\tau = 0$ is positive, thus $\text{sign}(\Pi(\mathbf{x}_0\ldots\mathbf{x}_{2P})) \geq 0$. For bosons then we can find all the exact ground-properties just sampling from the path-integral distribution $\Pi(\mathbf{x}_0\ldots\mathbf{x}_{2P})$.

For fermions, instead, the wave-function $\Psi(0)$ must be antisymmetric with respect to particle permutations, if we want that $|\langle\Psi_0|\Psi(0)\rangle|^2 \neq 0$. Therefore, $\Psi(0)$ obeys Fermi statistics and can take both negative and positive values. This means that we cannot interpret any longer Π as a probability distribution, and we cannot find the exact ground-state properties (Eqs. 11.15,11.16) using statistical averages. This infamous situation is known as the *sign problem* in Quantum Monte Carlo methods.

11.4 Reptation Quantum Monte Carlo

If we are in the lucky situation that the path-integral density is positive definite, we can devise a Monte Carlo procedure that samples from $\Pi(\mathbf{x}_0, \ldots, \mathbf{x}_{2P})$. Here I describe a method known as "Reptation Quantum Monte Carlo" [1,2], which is one of the easiest strategies to sample from the path integral. The method is based on Markov Chain sampling, and therefore is fully specified by the transition probabilities $T(\mathbf{x} \to \mathbf{x}')$, of going from a given path configuration $\mathbf{x} = \mathbf{x}_0 \ldots \mathbf{x}_{2P}$ to a new one $\mathbf{x}' = \mathbf{x}'_0 \ldots \mathbf{x}'_{2P}$

11.4.1 Monte Carlo moves

In the Reptation Quantum Monte Carlo we have two possible moves: one is called the "Right" move and the other one is called the "Left" move. The move then consists in the following: first pick at random (with uniform probability) wether to move Left or Right.

11.4.1.1 Move Right

If we pick the Right direction, the proposed configuration is $\mathbf{x}' = \mathbf{x}_1\mathbf{x}_2 \ldots \mathbf{x}_{2P}\mathbf{x}_T$, where $\mathbf{x}_0$ is being discarded, and the new configuration $\mathbf{x}_T$ is added on the right. To generate $\mathbf{x}_T$ we use as a transition probability the free propagator:

$$T(\mathbf{x}_{2P} \to \mathbf{x}_T) \quad = \quad G_1^{\Delta_\tau}(\mathbf{x}_{2P}, \mathbf{x}_T), \tag{11.18}$$

which is a Gaussian, and therefore we can easily generate random values for $\mathbf{x}_T$.

The acceptance probability for this move can be now computed to be:

$$\begin{aligned} A_R &= \min\left[1, \frac{\Pi(\mathbf{x}_1\mathbf{x}_2 \ldots \mathbf{x}_{2P}\mathbf{x}_T)}{\Pi(\mathbf{x}_0\mathbf{x}_1\mathbf{x}_2 \ldots \mathbf{x}_{2P})} \times \frac{T(\mathbf{x}_1 \to \mathbf{x}_0)}{T(\mathbf{x}_{2P} \to \mathbf{x}_T)}\right] \\ &= \min\left[1, \frac{\Psi(\mathbf{x}_1)}{\Psi(\mathbf{x}_0)} \frac{\Psi(\mathbf{x}_T)}{\Psi(\mathbf{x}_{2P})} \exp\left(-\frac{\Delta_\tau}{2}\Delta S_R\right)\right], \end{aligned} \tag{11.19}$$

with

$$\Delta S_R \quad = \quad H_0(\mathbf{x}_T) + H_0(\mathbf{x}_{2P}) - H_0(\mathbf{x}_0) - H_0(\mathbf{x}_1).$$

This expression arises since most of the factors cancel out in the ratios, but for the ratio of factors at the ends of the path.

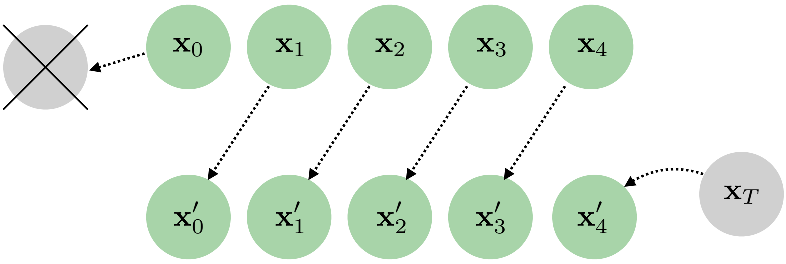


Figure 11.1: Example of *right* move in the Reptation Quantum Monte Carlo. All configurations are shifted by one position in the path integral, with the leftmost being discarded and the rightmost being generated with the importance-sampled propagator.

11.4.1.2 Move Left

If we pick the Left direction, the proposed configuration is $\mathbf{x}' = \mathbf{x}_T\mathbf{x}_0 \ldots \mathbf{x}_{2P-1}$, where $\mathbf{x}_{2P}$ is being discarded, and the new configuration $\mathbf{x}_T$ is added on the left of the path. To generate $\mathbf{x}_T$ we use the transition probability $T(\mathbf{x}_0 \to \mathbf{x}_T) = G_1^{\Delta_\tau}(\mathbf{x}_0 \to \mathbf{x}_T)$, and the acceptance probability now reads:

$$A_L = \min\left[1, \frac{\Psi(\mathbf{x}_T)}{\Psi(\mathbf{x}_0)}\frac{\Psi(\mathbf{x}_{2P-1})}{\Psi(\mathbf{x}_{2P})}\exp\left(-\frac{\Delta_\tau}{2}\Delta S_L\right)\right], \tag{11.20}$$

with

$$\Delta S_L = H_0(\mathbf{x}_T) + H_0(\mathbf{x}_0) - H_0(\mathbf{x}_{2P-1}) - H_0(\mathbf{x}_{2P}).$$

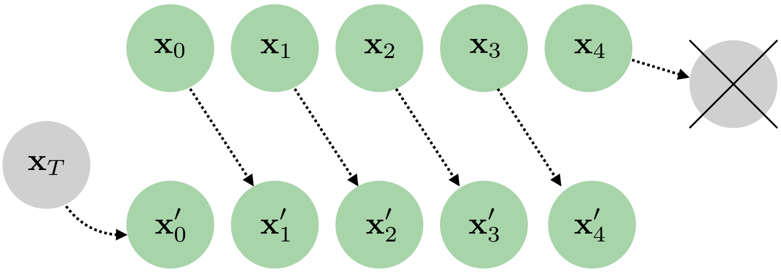


Figure 11.2: Example of *left* move in the Reptation Quantum Monte Carlo. All configurations are shifted by one position in the path integral, with the rightmost being discarded and the leftmost being generated with the importance-sampled propagator.

11.4.2 Practical aspects for the implementation

We now discuss some practical aspects useful for the implementation of the algorithm in the case of particles in continuous space.

In particular, a trial configuration is generated assigning positions to the N particles, i.e. $\mathbf{x}_T = \vec{r}_1', \ldots \vec{r}_N'$, with a probability $T(\mathbf{x} \to \mathbf{x}_T)$, as discussed previously.

To generate a trial configuration, starting from the configuration $\mathbf{x} = \vec{r}_1, \ldots \vec{r}_N$ (which in practice can be either the head or the tail of the path, depending on the chosen direction for the move), we do the following gaussian moves sampling from the free-particle propagator:

$$\vec{r}_i' = \vec{r}_i + \sqrt{\Delta_\tau}\vec{\chi}_i, \tag{11.21}$$

where $\vec{\chi}_i$ is a vector containing $3N$ independent, normally distributed random numbers.

11.4.2.1 Data Structure

As for almost any algorithm, first of all we have to specify a good data structure to hold the information we need. In particular, we want to store the path-integral configurations $\mathbf{x}_0\mathbf{x}_1\mathbf{x}_2 \ldots \mathbf{x}_{2P}$, where each $|\mathbf{x}_i\rangle = |\vec{r}_{i1}, \vec{r}_{i2}, \ldots \vec{r}_{iN}\rangle$. For example in python we might use a numpy ndarray with shape $(2P+1$,N,3) such that $r_{ij\alpha} = \mathrm{conf}[\mathrm{i}, \mathrm{j}, \alpha]$, where $i \in [0, 2P+1)$, $j \in [0, N)$ and $\alpha \in [0, 3)$ is the index corresponding to the spatial coordinate of the particle.

11.4.2.2 Putting everything together

At the beginning of our simulation, one typically does the following:

1. Pick the desired number of particles N

2. Pick a wave-function $\Psi(0)$ for which the local energy can be easily computed

3. Fix the total imaginary time τ and the number of time slices P.

4. Fix the total number of Monte Carlo moves to be done, $N_s \gg 1$.

A Monte Carlo move is then given by the following steps:

1. Generate the random direction d. To do so, take a uniform number $u \in [0,1)$, and if $u < 0.5$ $d =$ LEFT, otherwise $d =$ RIGHT.

2. If $d =$ LEFT generate a trial configuration $\mathbf{x}_T$ with probability $T(\mathbf{x}_{2P} \to \mathbf{x}_T)$, otherwise if $d =$ RIGHT with probability $T(\mathbf{x}_{2P} \to \mathbf{x}_T)$.

3. Compute the ratios entering the acceptance factors: if moving left compute$R_L = \frac{\Psi(\mathbf{x}_T)}{\Psi(\mathbf{x}_0)} \frac{\Psi(\mathbf{x}_{2P-1})}{\Psi(\mathbf{x}_{2P})} \exp\left(-\frac{\Delta_\tau}{2}\Delta S_L\right)$, otherwise $R_R = \frac{\Psi(\mathbf{x}_1)}{\Psi(\mathbf{x}_0)} \frac{\Psi(\mathbf{x}_T)}{\Psi(\mathbf{x}_{2P})} \exp\left(-\frac{\Delta_\tau}{2}\Delta S_R\right)$.

4. Generate a uniform random number $u \in [0,1)$, if $R_d > u$, then update the path configurations according to $\mathbf{x}^{(i+1)} = \mathbf{x}'$, otherwise leave the path unchanged, and increase the Markov chain counter i by one, i.e. $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)}$.

5. Measure the quantities of interest, for example compute the local energy on the end configuration of the path, or any other diagonal observable in the center of the path (Equations 11.15,11.16) and store these quantities in a vector $O^{(i)}$.

At the end of the simulation one then should:

1. Analyze the vector of observables, computing the average value $\mathbb{E}_\Pi[O(\mathbf{x})] \simeq \frac{1}{N_s}\sum_i O(\mathbf{x}^{(i)})$ and the variance (possibly corrected for correlations in the Markov chain, as explained in the previous lectures).

Further step to check the convergence to the ground-state wave-function and the effect of the Trotter splitting are:

1. Check that the time-step error is small enough, i.e. we can for example do another simulation with double the number of time slices (half the time step) and see if $\mathbb{E}_\Pi[O(\mathbf{x})]$ changes significantly beyond the statistical noise.

2. Check that the τ is large enough to have approached the ground-state energy. For example plot $\langle \hat{H} \rangle$ computed for values of increasing τ and see if you find convergence. This check should be done at parity of time-step, i.e. if for example we double the total imaginary time, we should equally double the value of P.

11.5 Importance Sampled Propagator in continuous space

In production codes, one rarely uses the simple form (11.18) for the transition probability. In particular, it is much more efficient to use the so-called *importance-sampled* propagator, which corresponds to taking a transition probability of the form:

$$\tilde{T}(\mathbf{x} \to \mathbf{x}_T) = \frac{1}{w(\mathbf{x})} G^{\Delta_\tau}(\mathbf{x}, \mathbf{x}_T) \frac{\Psi(\mathbf{x}_T)}{\Psi(\mathbf{x})}, \tag{11.22}$$

where $w(\mathbf{x}) = \int d\mathbf{y} G^{\Delta_\tau}(\mathbf{x}, \mathbf{y}) \frac{\Psi(\mathbf{y})}{\Psi(\mathbf{x})}$ is a normalization factor. Considering again the case of particles in continuous space, it can be shown that for sufficiently small time step one can generate samples from the transition probability above $\tilde{T}(\mathbf{r} \to \mathbf{r}')$ by following the prescription:

$$\vec{r}_i' = \vec{r}_i + \Delta_\tau \nabla_r \log \Psi(\vec{r}_1, \dots \vec{r}_N) + \sqrt{\Delta_\tau} \vec{\chi}_i, \tag{11.23}$$

where $\vec{\chi}_i$ is a vector containing $3N$ independent, normally distributed random numbers. When using the importance-sampled propagator, one has to change accordingly the acceptance probabilities, by taking into account the modified transition probability $\tilde{T}$. The resulting acceptance probabilities become:

$$\tilde{A}_R = \min\left[1, \exp\left(-\frac{\Delta_\tau}{2} \Delta \tilde{S}_R\right)\right] \tag{11.24}$$

$$\tilde{A}_L = \min\left[1, \exp\left(-\frac{\Delta_\tau}{2} \Delta \tilde{S}_L\right)\right], \tag{11.25}$$

with, respectively,

$$\Delta \tilde{S}_R = E_{\text{loc}}(\mathbf{x}_T) + E_{\text{loc}}(\mathbf{x}_{2P}) - E_{\text{loc}}(\mathbf{x}_0) - E_{\text{loc}}(\mathbf{x}_1) \tag{11.26}$$

$$\Delta \tilde{S}_L = E_{\text{loc}}(\mathbf{x}_T) + E_{\text{loc}}(\mathbf{x}_0) - E_{\text{loc}}(\mathbf{x}_{2P-1}) - E_{\text{loc}}(\mathbf{x}_{2P}). \tag{11.27}$$

Notice that the effect of taking this specific transition probability is that we have introduced the local energy in the acceptance factors, instead of the diagonal Hamiltonian H_0. This allows to have much less fluctuations in the weights entering the acceptance factors, since typically H_0 contains diverging terms coming from the Coulomb interactions, that can result in rejecting most of the moves. On the contrary, if the initial state Ψ is close to the exact ground state, then the fluctuations of the local energy are very small (see the discussion on the zero-variance principle in the previous lecture). Therefore, also the fluctuations of $\Delta\tilde{S}$ will be particularly small, and the acceptance probability very close to 1. This algorithm therefore would sample efficiently ground-state configurations, without rejection, and correcting for the variational bias of the initial state.

11.6 Diffusion Quantum Monte Carlo

An alternative approach to implement imaginary-time evolution is the Diffusion Quantum Monte Carlo method (see Ref. [3] for a review). This method is one of the oldest

Quantum Monte Carlo methods, and it is not based on the Metropolis algorithm. I will not discuss it during my lecture, mostly because of time constraints. It should also be noticed that Metropolis-based schemes, like the one discussed at length in this lecture, are typically superior to the Diffusion Monte Carlo in terms of efficiency on large systems.

Bibliography

[1] Stefano Baroni and Saverio Moroni, Phys. Rev. Lett. **82**, 4745 (1999).
[2] Giuseppe Carleo, Federico Becca, Saverio Moroni, and Stefano Baroni Phys. Rev. E **82**, 046719 (2010).
[3] W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal, Rev. Mod. Phys. **73**, 33 (2001).

Chapter 12

Finite-Temperature Quantum Monte Carlo

In the previous Lecture we have introduced our first path-integral methods. We have seen that one can obtain an exact representation of the imaginary time evolution, and then sample from the ground-state wave-function using Monte Carlo techniques. During this Lecture we will see how similar ideas can be applied to study *finite-temperature* properties.

12.1 Thermal Density Matrix

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

$$\langle \hat{O} \rangle = \frac{\mathrm{Tr}\left(\hat{O} e^{-\beta \hat{H}}\right)}{Z}, \tag{12.1}$$

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

$$Z = \mathrm{Tr}\left(\exp\left(-\beta \hat{H}\right)\right). \tag{12.2}$$

We denote the matrix elements of the unnormalized density matrix in some many-body basis $|\mathbf{x}\rangle$ as:

$$\rho^{\beta}\left(\mathbf{x}, \mathbf{y}\right) \equiv \left\langle \mathbf{x} \left| \exp\left(-\beta \hat{H}\right) \right| \mathbf{y} \right\rangle. \tag{12.3}$$

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

$$Z\left(N, T, V\right) = \sum_{\mathbf{x}} \rho^{\beta}\left(\mathbf{x}, \mathbf{x}\right), \tag{12.4}$$

and we have made explicit that in general this quantity depends on the number of particles, the temperature, and the volume V in which we assume the particles are confined.

Comparing Equation (12.3) with the expression for the imaginary-time propagator, introduced in the previous lecture, we immediately see that the density matrix elements coincides with the propagator of total imaginary-time β:

$$\rho^{\beta}(\mathbf{x},\mathbf{y}) \;=\; G^{\beta}(\mathbf{x},\mathbf{y}). \tag{12.5}$$

The inverse temperature then plays here the role of an effective imaginary time. This important connection allows us to immediately use all the path-integral machinery we have introduced in the previous Lecture.

In particular, we introduce again a small time step Δ_τ, such that $\beta = P \times \Delta_\tau$, and write the density matrix in the path-integral form:

$$\rho^{\beta}(\mathbf{x},\mathbf{y}) \;=\; \sum_{\mathbf{x}_1\ldots\mathbf{x}_{P-1}} \underbrace{G^{\Delta_\tau}(\mathbf{x}_0,\mathbf{x}_1)\ldots G^{\Delta_\tau}(\mathbf{x}_{P-1},\mathbf{x}_P)}_{P\text{ times}}, \tag{12.6}$$

where we have identified $\mathbf{x}_0 \equiv \mathbf{x}$ and $\mathbf{x}_P \equiv \mathbf{y}$. As explained in the previous lecture, the representation (12.6) is useful because we typically know how to compute controlled numerical approximations of the short-time propagators $G^{\Delta_\tau}(\mathbf{x}_0,\mathbf{x}_1)$. In particular, we have seen that in several interesting applications the Hamiltonian is the sum of two non-commuting terms, $\hat{H} = \hat{H}_0 + \hat{H}_1$, where the first term is diagonal in the chosen basis $|\mathbf{x}\rangle$. In that case we then use the second-order Trotter approximation:

$$G^{\Delta_\tau}(\mathbf{x},\mathbf{y}) \;=\; e^{-\Delta_\tau \frac{H_0(\mathbf{x})}{2}} G_1^{\Delta_\tau}(\mathbf{x},\mathbf{y}) e^{-\Delta_\tau \frac{H_0(\mathbf{y})}{2}} + \mathcal{O}(\Delta_t^2). \tag{12.7}$$

Using this decomposition, we see that the trace of the density matrix (12.4) takes the form:

$$\begin{aligned}
\sum_{\mathbf{x}_0} \rho^{\beta}\left(\mathbf{x}_0,\mathbf{x}_0\right) &= \sum_{\mathbf{x}_0,\mathbf{x}_1\ldots\mathbf{x}_{P-1}} G^{\Delta_\tau}(\mathbf{x}_0,\mathbf{x}_1)\ldots G^{\Delta_\tau}(\mathbf{x}_{P-1},\mathbf{x}_0) \\
&= \sum_{\mathbf{x}_0,\mathbf{x}_1\ldots\mathbf{x}_{P-1}} G_1^{\Delta_\tau}(\mathbf{x}_0,\mathbf{x}_1) e^{-\Delta_\tau H_0(\mathbf{x}_0)} \ldots G_1^{\Delta_\tau}(\mathbf{x}_{P-1},\mathbf{x}_0) e^{-\Delta_\tau H_0(\mathbf{x}_{P-1})} \\
&= \sum_{\mathbf{x}_0,\mathbf{x}_1\ldots\mathbf{x}_{P-1}} \Pi_{i=0}^{P-1} G_1^{\Delta_\tau}(\mathbf{x}_i,\mathbf{x}_{i+1}) e^{-\Delta_\tau H_0(\mathbf{x}_i)}. \qquad (12.8)
\end{aligned}$$

Since we have periodic boundaries in imaginary time, i.e. $\mathbf{x_p} = \mathbf{x_0}$, then all points in the path are equivalent. In this case we can imagine that the paths are closed rings, whereas in the zero-temperature, ground-state Path-Integral, the paths have open ends (reptiles with heads and tails).

12.1.1 Computing Properties

Having found an explicit path-integral expression for the thermal density matrix, we can also express the expectation values of observables in terms of this representation. For example, the expectation value of a diagonal operator:

$$\begin{aligned}
\langle \hat{O} \rangle &= \frac{\mathrm{Tr}\left(\hat{O} e^{-\beta \hat{H}}\right)}{Z} \\
&= \frac{\sum_{\mathbf{x}_0} \rho^{\beta}\left(\mathbf{x}_0,\mathbf{x}_0\right) O(\mathbf{x_0})}{\sum_{\mathbf{x}_0} \rho^{\beta}\left(\mathbf{x}_0,\mathbf{x}_0\right)}. \qquad (12.9)
\end{aligned}$$

If we introduce

$$\Pi(\mathbf{x}_0, \mathbf{x}_1, \ldots \mathbf{x}_P) \quad \equiv \quad \Pi_{i=0}^{P-1} G_1^{\Delta_\tau}(\mathbf{x}_i, \mathbf{x}_{i+1}) e^{-\Delta_\tau H_0(\mathbf{x}_i)}, \tag{12.10}$$

we can immediately notice that the finite-temperature quantum expectation values, Eq. (12.9), can be interpreted as *statistical* expectation values over the distribution Π, if this is positive definite. In the absence of a sign problem, we can therefore write

$$\langle O \rangle \quad = \quad \mathbb{E}_\Pi[O(\mathbf{x}_0)]. \tag{12.11}$$

This estimator can be further improved using the periodic boundaries in the path, i.e. we can average the estimator over all the points:

$$\langle O \rangle \quad = \quad \frac{1}{P} \sum_i \mathbb{E}_\Pi[O(\mathbf{x}_i)], \tag{12.12}$$

which ideally leads to a factor $1/\sqrt{P}$ improvement in the statistical uncertainty.

12.2 Continuous-Space particles

We specialize now our discussion to the case of particles in continuous space. We consider again the Hamiltonian for a system of N particles, of coordinates $\vec{r}_1, \vec{r}_2 \ldots \vec{r}_N$, and subjected to both one and two-body interactions:

$$\hat{H} \quad = \quad -\frac{\hbar^2}{2m} \sum_i^N \nabla^2_{\vec{r}_i} + \sum_i V_1(\vec{r}_i) + \sum_{i<j} V_2(\vec{r}_i, \vec{r}_j). \tag{12.13}$$

We first assume that particles, although being identical, are distinguishable. Therefore, we do not need to impose the Bose or Fermi restriction to the Hilbert space. In section 12.2.1 we will describe the treatment of identical particles obeying Bose statistics.

In this case, the path-integral probability density reads:

$$\Pi(\mathbf{x}_0, \mathbf{x}_1 \ldots \mathbf{x}_P) \quad \propto \quad \Pi_{i=0}^{P-1} \Pi_{j=1}^{N} \exp\left[-\frac{\left(\vec{r}_j(i) - \vec{r}_j(i+1)\right)^2}{2\Delta_\tau \hbar^2/m} \right] e^{-\Delta_\tau H_0(\mathbf{x}_i)}, \tag{12.14}$$

where we denote $\vec{r}_j(i)$ the position of the $j-$th particle at discrete imaginary-time i, i.e. $|\mathbf{x}_i\rangle = |\vec{r}_1(i), \vec{r}_2(i), \ldots \vec{r}_N(i)\rangle$ and, again, $\mathbf{x}_P = \mathbf{x}_0$.

Note that eq. (12.14) represents the Boltzmann weight of a classical system of polymers. Every polymer is a necklace of *beads* (particles at a given imaginary-time) interacting as if they were connected by ideal springs. This harmonic interaction is due to the kinetic density matrix. In the Trotter approximation, *beads* with the same imaginary time index i, i.e., belonging to the same time-slice, interact with the inter-particle potential $H_0(\mathbf{x}_i)$. With higher order approximations one generally introduces effective inter-particle 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 12.2.1 we will see how the indistinguishability of identical particles modifies the "polymer" description of the quantum many body system.

12.2.1 Bose symmetry

The expression (12.14) 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 (12.6) is to sum over all possible permutations of the particle labels in one of the two arguments:

$$\rho^{\beta}_{\mathrm{Bose}}\left(\mathbf{x}_1, \mathbf{x}_2\right) = \frac{1}{N!}\sum_{\mathcal{P}} \rho^{\beta}\left(\mathbf{x}_1, \mathcal{P}(\mathbf{x}_2)\right), \tag{12.15}$$

where $\mathcal{P}$ denotes each of the $N!$ permutations of the particle labels; this means that $\mathcal{P}(\mathbf{x}) = (\vec{r}_{\mathcal{P}_1}, \vec{r}_{\mathcal{P}_2}, \ldots, \vec{r}_{\mathcal{P}_N})$, where $\mathcal{P}_i$, 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. (12.15) we obtain the partition function for identical Bose particles:

$$Z_{\mathrm{Bose}}(N, V, T) \quad = \quad \frac{1}{N!}\sum_{\mathcal{P}} \int d\mathbf{x}_0 \ldots d\mathbf{x}_P \Pi(\mathbf{x}_0, \mathbf{x}_1 \ldots \mathbf{x}_P)\delta(\mathbf{x}_P - \mathcal{P}(\mathbf{x}_0)), \tag{12.16}$$

with the new boundary condition $\mathbf{x}_P = \mathcal{P}(\mathbf{x}_0)$. As a consequence of symmetrization the necklaces constituting the polymers are not closed on themselves. The last bead of the i-th polymer is connected to the first bead of the $\mathcal{P}_i$-th world-line (see Fig. 12.1 for an example of path configuration).

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.(12.16) 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. (12.16). 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.

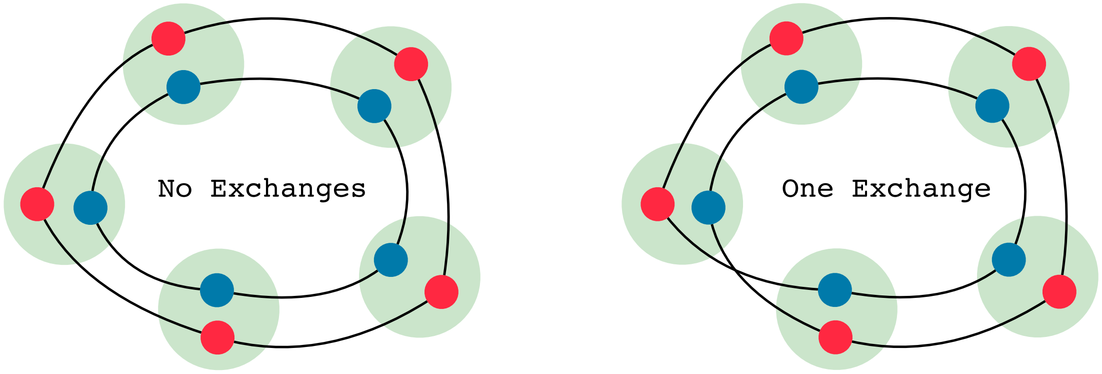


Figure 12.1: Example of path-integral configurations for a two-particle systems. In the left path, we have two separate closed polymers (rings) one for particle 1 (red beads) and one for particle 2 (blue beads). In the right configuration we are sampling instead a particle exchange between particle 1 and 2. In this case the two polymers become a single, larger one.

12.3 Path-Integral Monte Carlo

In this section we describe the Monte Carlo procedure to sample path-integrals.

The Path-Integral Monte Carlo method is again based on Markov Chain sampling, and therefore is fully specified by the transition probabilities $T(\mathbf{x} \to \mathbf{x}')$, of going from a given path configuration $\mathbf{x} = \mathbf{x}_0 \dots \mathbf{x}_P$ to a new one $\mathbf{x}' = \mathbf{x}'_0 \dots \mathbf{x}'_P$.

In general, we will need both moves that change the particle positions at given imaginary times (beads), and Monte Carlo moves that sample the permutations. Here we discuss only the essential steps to build an elementary Path-Integral algorithm. More advanced moves can be found in Refs. [1, 2, 3] .

12.3.1 Moving a single bead

The simplest move we can imagine starts with randomly picking a discrete imaginary time i, which we sample uniformly in $i \in [0, P-1)$, and a random particle index $j \in [1, N]$. Then we propose to randomly move the particle (at position $\vec{r}_j(i)$), to another point $\vec{r}_j(i)'$. For simplicity we pick a Gaussian transition probability $T(\vec{r}_j(i), \vec{r}_j(i)')$, such that the new particle position is generated according to $\vec{r}_j(i)' = \vec{r}_j(i) + \overrightarrow{\mathrm{Normal}}(\sigma)$. Here $\overrightarrow{\mathrm{Normal}}(\sigma)$ denotes the sum of 3 independent normal distributions, one per spatial direction. Since in this case the transition probability is symmetric, the acceptance ratio is just given by the ratio of the two probabilities distribution : $\Pi(\mathbf{x}')/\Pi(\mathbf{x})$, with $\mathbf{x}' = \mathbf{x}_0 \dots \mathbf{x}'_i \dots \mathbf{x}_{P-1}$, and $\mathbf{x}'_i = \vec{r}_1(i), \dots \vec{r}_j(i)', \dots \vec{r}_N(i)$. The probability to accept the move is then:

$$A_s = \min \left[1, \frac{\exp\left[-\frac{(\vec{r}_j(i-1) - \vec{r}_j(i)')^2 + (\vec{r}_j(i)' - \vec{r}_j(i+1))^2}{2\Delta_\tau} \right]}{\exp\left[-\frac{(\vec{r}_j(i-1) - \vec{r}_j(i))^2 + (\vec{r}_j(i) - \vec{r}_j(i+1))^2}{2\Delta_\tau} \right]} \exp\left[-\Delta_\tau \left(H_0\left(\mathbf{x}'_j\right) - H_0\left(\mathbf{x}_j\right) \right) \right] \right]. \tag{12.17}$$

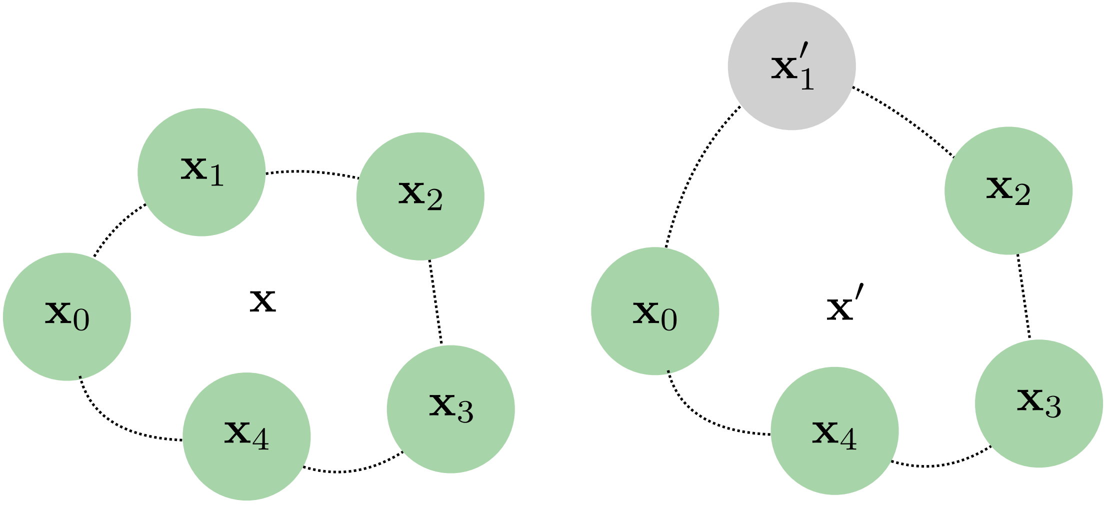


Figure 12.2: Example of a single-bead move in the Path-Integral Quantum Monte Carlo. One random bead is picked (in this case bead 1) and it is displaced to a random configuration with a normal distribution.

12.3.2 Moving multiple beads

The previously describe 'single bead' move becomes extremely inefficient when the number of time-slices P increases (*critical slowing down*), so one faces ergodicity problems. A better approach is to reconstruct a larger segment of the path, involving a certain number of m adjacent (in time) beads. An example of such multi-bead move is depicted in Figure 12.3.

The most commonly adopted strategy to sample a piece of path is to fix the two ends (in the example, $\mathbf{x}_0$ and $\mathbf{x}_4$) and generate the missing beads in the middle with a probability proportional to the free propagator. In the example shown in Figure 12.3, the transition probability would then correspond to:

$$T_{\mathrm{mb}}(\mathbf{x} \to \mathbf{x}') = G_1^{\Delta_\tau}(\mathbf{x}_0, \mathbf{x}'_1) G_1^{\Delta_\tau}(\mathbf{x}'_1, \mathbf{x}'_2) G_1^{\Delta_\tau}(\mathbf{x}'_2, \mathbf{x}'_3) G_1^{\Delta_\tau}(\mathbf{x}'_3, \mathbf{x}_4). \tag{12.18}$$

This process effectively corresponds to bridging the two ends with a sequence of Gaussians. Since the transition probability samples exactly the free-particle part of the path-integral weight, it is easy to see that the acceptance probability for this move depends only on the potential/interaction energy:

$$A_{\mathrm{mb}} \quad = \quad \min\left[1, \exp\left[-\Delta_\tau \left(\sum_j^m H_0\left(\mathbf{x}'_{j_0+j}\right) - H_0\left(\mathbf{x}_{j_0+j}\right)\right)\right]\right], \tag{12.19}$$

where j_0 is the initial bead for the bridge, and m is the number of beads that are being changed by this move (in the example, we would have $j_0 = 0$ and $m = 3$. The remaining point to be addressed is then just how to to sample points distributed according to the general transition probability:

$$T_{\mathrm{mb}}(\mathbf{x} \to \mathbf{x}') \quad = \quad \Pi_{j=0}^m G_1^{\Delta_\tau}(\mathbf{x}'_{j_0+j}, \mathbf{x}'_{j_0+j+1}), \tag{12.20}$$

where we have set $\mathbf{x}'_{j_0} \equiv \mathbf{x}_{j_0}$ and $\mathbf{x}'_{j_0+m+1} \equiv \mathbf{x}_{j_0+m+1}$.

In order to solve this problem, let us for simplicity concentrate on the one-dimensional case, and address the issue of sampling from a single point x', from the probability $G_1^{\Delta_\tau^1}(x_A, x')G_1^{\Delta_\tau^2}(x', x_B)$, where we take fixed end points x_A and x_B, bridged by a free propagator of time step Δ_τ^1 on the left and a free propagator of time step Δ_τ^2 on the right. Considering the explicit expressions for the free propagators, we have:

$$\begin{aligned} G_1^{\Delta_\tau^1}(x_A, x')G_1^{\Delta_\tau^2}(x', x_B) &\propto \exp\left[-\frac{x_A^2 - 2x'x_A + x'^2}{2\Delta_\tau^1} - \frac{x'^2 - 2x'x_B + x_B^2}{2\Delta_\tau^2}\right] \\ &\propto \exp\left[-\frac{(x' - x_{AB})^2}{2\sigma^2}\right], \end{aligned} \tag{12.21}$$

where

$$\sigma^2 = \left(1/\Delta_\tau^1 + 1/\Delta_\tau^2\right)^{-1} \tag{12.22}$$

and

$$x_{AB} = \frac{\Delta_\tau^2 x_A + \Delta_\tau^1 x_B}{\Delta_\tau^1 + \Delta_\tau^2}. \tag{12.23}$$

Equation (12.21) therefore tells us that to sample a point bridging two fixed ends x_A and x_B, we just need to consider a modified Gaussian, whose variance is given Eq. (12.22) and whose mean value is given by Eq. (12.23). This procedure can be easily used to reconstruct the full bridge. One starts from the left end (say $x_A \equiv x_{j_0}$ and $x_B \equiv x_{j_0+m+1}$), samples a point $x'_{j_{0+1}}$ at imaginary time j_0+1 using the previously introduced modified Gaussian, with $\Delta_\tau^1 = \Delta_\tau$ and $\Delta_\tau^2 = m\Delta_\tau$. Then, one advances the left end of the bridge, fixing now $x_A \equiv x'_{j_0+1}$ and sampling x'_{j_0+2}, with $\Delta_\tau^2 = (m-1)\Delta_\tau$. We end this procedure when all the points have been generated. For a complete discussion of the algorithm we suggest to read Ref. [2] (page 152).

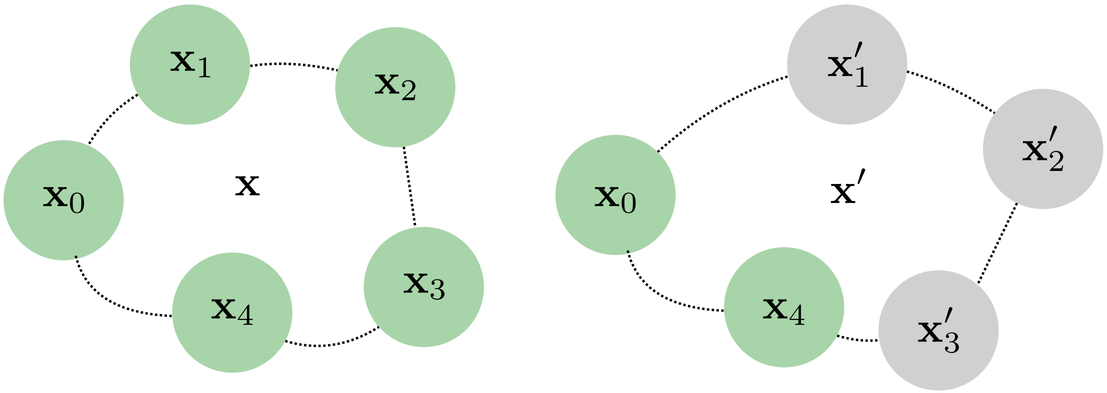


Figure 12.3: Example of a multi-bead move in the Path-Integral Quantum Monte Carlo. A path segment is randomly picked (in this case from bead 1 to bead 3) and it is reconstructed using a Gaussian bridge.

12.3.3 Displacing entire polymers

Even if we allow for the construction of large segments of paths, one can still face critical slowing down in the sampling. The reason is that if the number of beads is large, then it takes many moves to significantly displace *entire* polymers. Let us consider for simplicity the case in which we ignore particle exchanges. In this case, what we can do is to pick a random closed polymer $j \in [1, N]$, consider all the particles in that ring: $\vec{r}_j(i)$ with $i = 1, \ldots P$ and displace them according to $\vec{r}'_j(i) = \vec{r}_j(i) + \Delta R_j$, where $\Delta R_j = \vec{\text{Normal}}(\sigma)$ is a random variable which is identical for all the beads at different imaginary-time (i.e. we do a rigid translation of all the polymer corresponding to particle j). We immediately see that this move does not affect the kinetic part of the propagator (since it depends only on position differences), but only the part containing the potential energy. Therefore the acceptance is

$$A_d \quad = \quad \min\left[1, \exp\left[-\Delta_\tau \sum_i \left(H_0\left(\vec{r}_j(i) + \Delta R_j\right) - H_0\left(\vec{r}_j(i)\right)\right)\right]\right]. \qquad (12.24)$$

12.3.4 Sampling permutations

Apart from the moves previously described, one has also to sample all the possible particle permutations $\mathbf{x}_P = \mathcal{P}\mathbf{x}_0$. Any path-integral Monte Carlo methods for bosons should therefore provide a move which exchanges particles. For example this can be done cutting two independent polymers, and then connecting them. Sampling permutations with ergodic moves can be rather challenging, and only relatively recently a powerful algorithm has introduced which efficiently samples permutations for thousands of particles. This method is the Worm algorithm of Reference [3]. During this Lecture we will not have the time to discuss in detail how to implement exchange moves, but the interested student can read the paper [3], where all the several steps to implement the Worm algorithm are described in detail. Notice that implementing the algorithm from scratch is a nontrivial task for a beginner in the field, and might require several weeks of coding and debugging.

12.3.5 Energy expectation value

In 12.1.1 we have seen how to recast thermal expectation values of diagonal observables in terms of statistical expectation values over the path distribution $\Pi(\mathbf{x}_0, \ldots \mathbf{x}_P)$. The energy per particle, E/N, can be found through its thermodynamic definition. In particular, E/N is just the 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 partition function defined through the path-integral representation of Eq. (13.7), we obtain the following estimator for the energy per particle (called *thermodynamic estimator*):

$$\frac{E_{\text{th}}}{N} = \mathbb{E}_\Pi\left[\frac{d}{2\Delta_\tau} - \frac{1}{2(\Delta_\tau)^2 PN}\sum_{i=0}^{P-1}\sum_j^N \left(\vec{r}_j(i) - \vec{r}_j(i+1)\right)^2 + \frac{1}{PN}\sum_i^{P-1} H_0\left(\mathbf{x}_i\right)\right], \quad (12.25)$$

where d is the system dimensionality. More details, and other energy estimators based on the Virial theorem can be found in Ref. [1].

Bibliography

[1] D. M. Ceperley, Review of Modern Physics **67**, 279 (1995).

[2] W. Krauth, Statistical Mechanics: Algorithms and Computations, Oxford Master Series in Physics (2006).

[3] M. Boninsegni, N. Prokof'ev, and B. Svistunov, Physical Review E 74, 036701 (2006).

[4] 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.

Chapter 13

Path-Integral Monte Carlo for Lattice Models

In the previous Lecture we have introduced the finite-temperature path-integral Monte Carlo, and applied it to particles in continuous space. We now want to see how the path-integral formulation is applied to lattice (spin) models, and how we can devise a Monte Carlo algorithm to obtain finite-temperature properties.

13.1 Transverse-Field Ising model

The finite-temperature path-integral representation we have derived in the previous Lecture is completely general, and can be readily generalized to the case of spin systems. Let us consider once more as an example the transverse-field Ising model in one dimension:

$$\hat{H} = -\Gamma \sum_{i=0}^{N-1} \hat{\sigma}_i^x - J \sum_{i=0}^{N-1} \hat{\sigma}_i^z \hat{\sigma}_{i+1}^z, \tag{13.1}$$

for a system of N spins and with periodic boundary conditions $\hat{\sigma}_N^z \equiv \hat{\sigma}_0^z$.

As a state vector, we will use the value of the spin projections along the z direction, i.e. $|\mathbf{x}\rangle = |\sigma_1^z \sigma_2^z \dots \sigma_N^z\rangle$, exactly as we have previously done with other techniques applied to spin models.

13.1.1 Short-Time propagator

The first thing we need to derive a useful path-integral formulation is then the propagator

$$G^{\Delta_\tau}(\mathbf{x}, \mathbf{y}) \equiv \langle \mathbf{x}| e^{-\Delta_\tau \hat{H}} |\mathbf{y}\rangle, \tag{13.2}$$

for which we invoke again the second-order Trotter decomposition:

$$G^{\Delta_\tau}(\mathbf{x}, \mathbf{y}) = e^{-\Delta_\tau \frac{H_0(\mathbf{x})}{2}} G_1^{\Delta_\tau}(\mathbf{x}, \mathbf{y}) e^{-\Delta_\tau \frac{H_0(\mathbf{y})}{2}} + \mathcal{O}(\Delta_t^2). \tag{13.3}$$

In this expression we identify the diagonal part: $H_0(\mathbf{x}) = -\sum_i \hat{\sigma}_i^z \hat{\sigma}_{i+1}^z$ the classical energy, and

$$\begin{aligned} G_1^{\Delta_\tau}(\mathbf{x},\mathbf{y}) &= \langle \mathbf{x}| e^{\Delta_\tau \Gamma \sum_i \hat{\sigma}_i^x} |\mathbf{y}\rangle \\ &= \Pi_i \langle \mathbf{x}| e^{\Delta_\tau \Gamma \hat{\sigma}_i^x} |\mathbf{y}\rangle \\ &= \Pi_i g_1^{\Delta_\tau}(\sigma_i^z(\mathbf{x}), \sigma_i^z(\mathbf{y})), \end{aligned} \quad (13.4)$$

where we have first used the fact that all the different $\hat{\sigma}_i^x$ commute (thus the N spin propagator can be factorized, in the second line) and then introduced the single-spin propagators

$$\begin{aligned} g_1^{\Delta_\tau}(\sigma^z, \sigma^{z'}) &= \langle \sigma^z| \exp\left[\Delta_\tau \Gamma \hat{\sigma}^x\right] |\sigma^{z'}\rangle \\ &= \langle \sigma^z| \begin{pmatrix} \cosh \Gamma \Delta_\tau & \sinh \Gamma \Delta_\tau \\ \sinh \Gamma \Delta_\tau & \cosh \Gamma \Delta_\tau \end{pmatrix} |\sigma^{z'}\rangle \\ &= \delta_{\sigma^z, \sigma^{z'}} \cosh \Gamma \Delta_\tau + (1 - \delta_{\sigma^z, \sigma^{z'}}) \sinh \Gamma \Delta_\tau. \end{aligned} \quad (13.5)$$

13.1.2 Path-Integral expression

We then consider the thermal density matrix of the system $\hat{\rho}^\beta = \exp\left(-\beta \hat{H}\right)/Z$, where $\beta = 1/k_B T$, with k_B the Boltzmann's constant, and T the temperature of the system and the partition function:

$$Z = \sum_{\mathbf{x}_0} \rho^\beta \left(\mathbf{x}_0, \mathbf{x}_0\right). \quad (13.6)$$

The path-integral expression for the partition function has been derived in the previous Lecture and reads:

$$Z = \sum_{\mathbf{x}_0 \dots \mathbf{x}_{P-1}} \Pi_{j=0}^{P-1} G_1^{\Delta_\tau}(\mathbf{x}_j, \mathbf{x}_{j+1}) e^{-\Delta_\tau H_0(\mathbf{x}_j)}, \quad (13.7)$$

with periodic boundaries in imaginary time, i.e. $\mathbf{x_p} = \mathbf{x_0}$. In this case the path-integral configurations are completely determined by the value of the i-th spin at each imaginary time $j = 0, \dots P$. We call these values σ_{ij}^z, such that the full many-body configuration at given imaginary-time is given by $\mathbf{x}_j = \sigma_{1j}^z, \sigma_{2j}^z \dots, \sigma_{Nj}^z$. The path-integral formulation then maps the 1D quantum model for N spins onto the 2D classical model for $N \times P$ effective spins.

13.1.3 Classical 2D Ising Model

We can actually do more than that and show that the 2D classical system is a simple classical Ising model, with some specific couplings. In order to do so, we start noticing that the single-spin propagator can be written as the exponential of an effective interaction between the spins σ^z and $\sigma^{z\prime}$:

$$\begin{aligned}
g_1^{\Delta_\tau}(\sigma^z, \sigma^{z'}) &= \exp\left[\frac{1}{2}\left(1+\sigma^z\sigma^{z'}\right)\log\left(\cosh\Gamma\Delta_\tau\right)+\right. \\
&\left.+\frac{1}{2}\left(1-\sigma^z\sigma^{z'}\right)\log\left(\sinh\Gamma\Delta_\tau\right)\right] \qquad (13.8) \\
&= \exp\left[\sigma^z\sigma^{z'}\frac{1}{2}\left[\log\left(\cosh\Gamma\Delta_\tau\right)-\log\left(\sinh\Gamma\Delta_\tau\right)\right]\right]\times \\
&\times\exp\left[\frac{1}{2}\left[\log\left(\cosh\Gamma\Delta_\tau\right)+\log\left(\sinh\Gamma\Delta_\tau\right)\right]\right] \qquad (13.9) \\
&= e^{-\sigma^z\sigma^{z'}J^{\text{eff}}}\times\text{const}, \qquad (13.10)
\end{aligned}$$

with $J^{\text{eff}} = \frac{1}{2}\log\left(\tanh\Gamma\Delta_\tau\right)$. From this expression, it is then easy to realize that that the quantum partition function is completely equivalent to the classical partition function:

$$\begin{aligned}
Z &= \sum_{\{\sigma^z_{i,j}\}} e^{-\beta^{\text{eff}}E^{\text{eff}}(\sigma^z_{i,j})} \qquad (13.11) \\
&= \sum_{\{\sigma^z_{i,j}\}} \exp\left[-\sum_i\sum_j\left(-\Delta_\tau J\sigma^z_{i,j}\sigma^z_{i+1,j}+J^{\text{eff}}\sigma^z_{i,j}\sigma^z_{i,j+1}\right)\right]. \qquad (13.12)
\end{aligned}$$

The classical energy is therefore simply that of a classical Ising model in 2 dimensions:

$$E^{\text{eff}}(\sigma^z_{i,j}) = \sum_{i=0}^{N-1}\sum_{j=0}^{P-1}\left(-\Delta_\tau J\sigma^z_{i,j}\sigma^z_{i+1,j}+J^{\text{eff}}\sigma^z_{i,j}\sigma^z_{i,j+1}\right), \qquad (13.13)$$

where we have periodic boundary conditions both in the spatial axis ($\sigma^z_N \equiv \sigma^z_0$) and in the imaginary-time axis ($\sigma^z_P \equiv \sigma^z_0$), and the effective classical inverse temperature is just $\beta^{\text{eff}} = 1$. An example of configuration is shown in Figure 13.1.

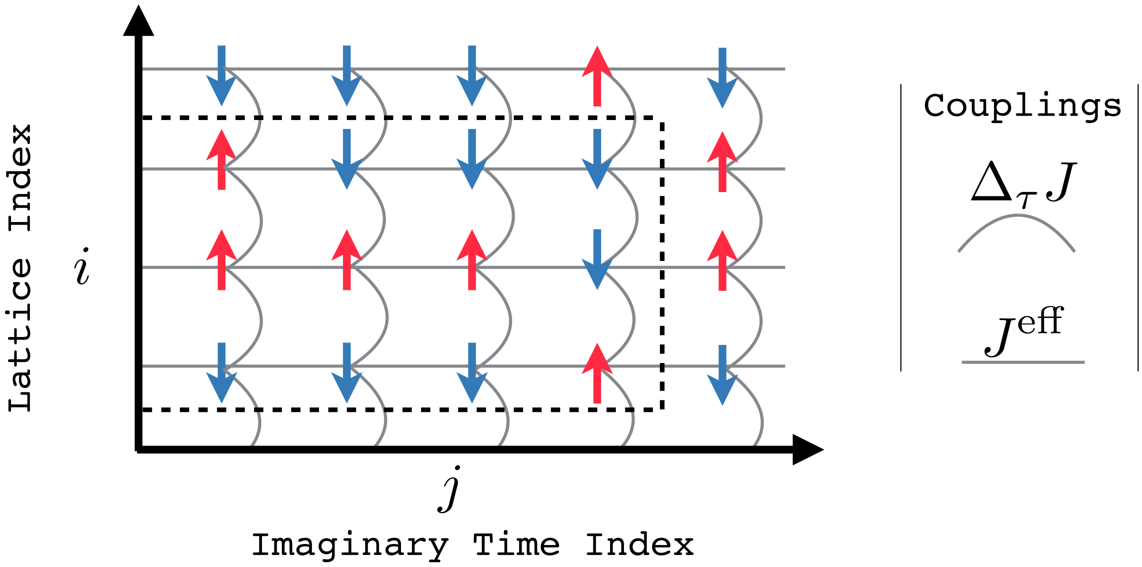


Figure 13.1: Example of path-integral configurations for the transverse-field Ising model in one dimension. The quantum partition function is equivalent to the classical partition function of a two-dimensional Ising model with couplings $\Delta_\tau J$ along the spatial direction (vertical axis in the Figure) and $J^{\text{eff}} = -\frac{1}{2}\log(\tanh \Gamma\Delta_\tau)$ along the imaginary-time direction (horizontal axis in the Figure). In the Figure, there are $N = 3$ spins and $P = 4$ imaginary-time slices, with periodic boundary conditions along both directions.

13.1.4 Monte Carlo Sampling

Once established the equivalent classical partition function, Eq. (13.11), we can readily devise a Markov-Chain Monte Carlo sampling strategy to generate path-integral configurations. In particular, any classical Monte Carlo algorithm used to sample the partition function of the Ising model can be used. One of the simplest approach (albeit ineffective in the ferromagnetic phase) is to use single spin flip moves.

We can just pick uniformly both a spatial and a time index, respectively $i \in [0, N)$ and $j \in [0, P)$ and propose a spin flip for the σ^z_{ij} spin. In the new configuration we would then have $\sigma^{z\prime}_{ij} = -\sigma^z_{ij}$, and the acceptance probability is readily computed to be:

$$\begin{aligned} A_S(\sigma^z_{ij} \to \sigma^{z\prime}_{ij}) = \\ &= \min\left[1, \frac{\exp\left[-\Delta_\tau J\left(\sigma^z_{i-1,j}\sigma^{z\prime}_{i,j} + \sigma^{z\prime}_{i,j}\sigma^z_{i+1,j}\right) - J^{\text{eff}}\left(\sigma^z_{i,j-1}\sigma^{z\prime}_{i,j} + \sigma^{z\prime}_{i,j}\sigma^z_{i,j+1}\right)\right]}{\exp\left[-\Delta_\tau J\left(\sigma^z_{i-1,j}\sigma^z_{i,j} + \sigma^z_{i,j}\sigma^z_{i+1,j}\right) - J^{\text{eff}}\left(\sigma^z_{i,j-1}\sigma^z_{i,j} + \sigma^z_{i,j}\sigma^z_{i,j+1}\right)\right]}\right] \\ &= \min\left[1, \exp\left[2\Delta_\tau J\left(\sigma^z_{i-1,j}\sigma^z_{i,j} + \sigma^z_{i,j}\sigma^z_{i+1,j}\right) + 2J^{\text{eff}}\left(\sigma^z_{i,j-1}\sigma^z_{i,j} + \sigma^z_{i,j}\sigma^z_{i,j+1}\right)\right]\right]. \quad (13.14) \end{aligned}$$

13.1.5 Energy Estimator

An estimator for the expectation value of the energy can be immediately found using again the thermodynamics relation $\langle H\rangle_T = -\partial_\beta \log Z$. I leave as an exercise to derive it in this case.

13.2 Continuous-Time path integrals

The most remarkable difference between lattice and continuous-space systems is that for lattice models we can efficiently (and exactly) take the continuous-time limit $\Delta_\tau \to 0$, whereas for systems in continuous space this is not the case. To see how we can eventually get rid of the Trotter error, we need to introduce a different representation of the partition function, based on a perturbative expansion.

The starting point is the Dyson series for the exponential of a matrix:

$$e^{-\beta(\hat{H}_0+\hat{H}_1)} = e^{-\beta\hat{H}_0} \sum_{n=0}^{\infty}(-1)^n \underbrace{\int d\tau_1 \dots d\tau_n}_{0<\tau_1\leq\dots\leq\tau_n<\beta} \hat{H}_1(\tau_n)\dots\hat{H}_1(\tau_1),$$

where $\hat{H}_1(\tau) = e^{\tau\hat{H}_0}\hat{H}_1 e^{-\tau\hat{H}_0}$ is the time-evolved $\hat{H}_1$ in the so-called "interaction representation". The Dyson series is basically the exact summation of all the contributions in perturbation theory, of order $\hat{H}_1^n$ with $n = 0, 1\dots\infty$. Notice that the imaginary times arising in the integrals are t-ordered, i.e. $0 < \tau_1 \leq \dots \leq \tau_n < \beta$. [1]

As much as we have done for the path-integral representation of the exponential, we can derive now an expression we can sample from using Monte Carlo. In particular, we will need now not only to sample over the spin configurations at different imaginary times, but also over all the possible orders in perturbation theory n, and over all the possible values of the times. Let us start by considering the first few terms, and inserting completeness in the middle:

$$\begin{aligned} Z &= \sum_{\mathbf{x}_0}\langle\mathbf{x}_0|e^{-\beta\hat{H}_0}|\mathbf{x}_0\rangle - \sum_{\mathbf{x}_1} e^{-\beta H_0(\mathbf{x}_1)} \int_0^\beta d\tau_1 \langle\mathbf{x}_1|\hat{H}_1|\mathbf{x}_1\rangle + \\ &+ \sum_{\mathbf{x}_1,\mathbf{x}_2} e^{-\beta H_0(\mathbf{x}_2)} \int_0^\beta d\tau_2 \int_0^{\tau_2} d\tau_1 e^{(\tau_2-\tau_1)H_0(\mathbf{x}_2)}\langle\mathbf{x}_2|\hat{H}_1|\mathbf{x}_1\rangle e^{(\tau_1-\tau_2)H_0(\mathbf{x}_1)}\langle\mathbf{x}_1|\hat{H}_1|\mathbf{x}_2\rangle + \\ &+ \dots \end{aligned} \quad (13.15)$$

The goal of our Monte Carlo sampling will be now to sample from the probability distribution

$$\begin{aligned} &\Pi_{\text{ct}}(\mathbf{x}_1\dots\mathbf{x}_n, \tau_1\dots\tau_n, n) = \\ &\quad = \frac{1}{Z}e^{-\beta H_0(\mathbf{x}_n)}(-1)^n\hat{H}_1(\mathbf{x}_n,\mathbf{x}_{n-1},\tau_n)\dots\hat{H}_1(\mathbf{x}_2,\mathbf{x}_1,\tau_2)\hat{H}_1(\mathbf{x}_1,\mathbf{x}_n,\tau_1), \end{aligned} \quad (13.16)$$

where we have introduced the matrix elements $H_1(\mathbf{x},\mathbf{x}',\tau) = e^{\tau H_0(\mathbf{x})}\langle\mathbf{x}|\hat{H}_1|\mathbf{x}'\rangle e^{-\tau H_0(\mathbf{x}')}$.

In general, in order to sample from this probability distribution we have not include a set of Markov-Chain moves that:

1. Change the spin configurations, $\mathbf{x}_j$
2. Change the perturbation order, n

[1] A good and self-contained derivation of the Dyson series can be found at https://en.wikipedia.org/wiki/Dyson_series.

3. Move the imaginary times, according to the time-ordering constraint

As a result of this extended sampling space (with respect to the standard path-integral configurations, comprising only the set 1 in the previous list), we will have that our simulation will not carry any time-step error!

I will now discuss an example where these moves can be seen explicitly at work.

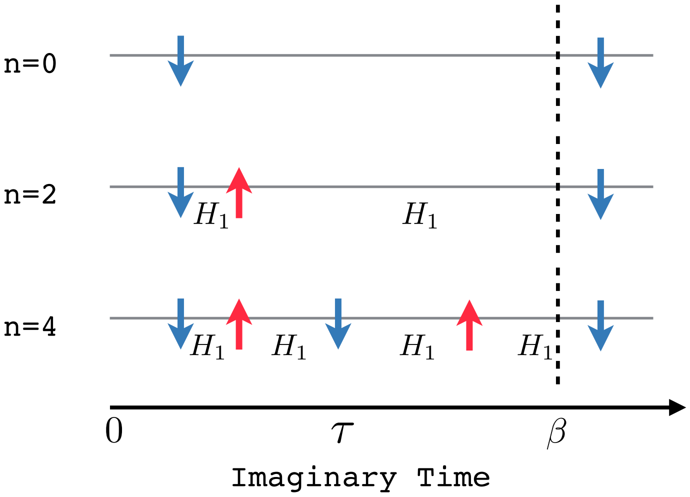


Figure 13.2: Example of allowed configurations for the continuous-time path-integral for a single spin. Different perturbative order are shown (notice that they are all even), and vertical dashed lines denote periodic boundaries in imaginary time. In each configuration (but for the special case $n = 0$) the total number of up spins must match the total number of down spins, and each spin is followed in imaginary time by a spin of opposite sign.

13.2.1 Example: single spin Hamiltonian

To give a concrete example, let us now examine the expressions above in the case of a simple single-spin Hamiltonian: $\hat{H} = -\Gamma\hat{\sigma}_x + \Gamma_z\hat{\sigma}_z$, and take $\hat{H}_0 = \Gamma_z\hat{\sigma}_z$, $\hat{H}_1 = -\Gamma\hat{\sigma}_x$. We therefore have that $H_1(\sigma^z, \sigma^{z'}, \tau) = \langle\sigma^z|H_1(\tau)|\sigma^{z\prime}\rangle = -\Gamma e^{\Gamma_z\tau(\sigma_z-\sigma'_z)}(1-\delta_{\sigma_z,\sigma'_z}) = -(1-\delta_{\sigma_z,\sigma'_z})\Gamma e^{2\Gamma_z\tau\sigma^z}$, where in the last equality we have used the fact that we need $\sigma'_z = -\sigma_z$ for the matrix element to be non-vanishing. In this case the probability to be sampled is therefore:

$$\begin{aligned}\Pi_{\mathrm{ct}}(\sigma^z_1 \ldots \sigma^z_n, \tau_1 \ldots \tau_n, n) = \\ = \frac{1}{Z}\Gamma^n e^{-\beta\sigma^z_n} e^{2\Gamma_z \sum_{j=1}^n \tau_j \sigma^z_j} C(\sigma, n, \tau),\end{aligned} \tag{13.17}$$

where we have introduced a constraint on the allowed configurations:

$$C(\sigma, n, \tau) \quad = \quad \delta(\sigma^z_j \neq \sigma^z_{(j+1) \mod n}) \times \delta_{n,2k} \times \delta(0 < \tau_1 \leq \tau_2 \leq \ldots \tau_n < \beta). \tag{13.18}$$

The first constraint comes from the fact that the matrix elements of $\hat{H}_1$ are purely off-diagonal, and we must flip a spin every time we introduce a term $\hat{H}_1$ in the path.

The second constraint implies that we must have only an even number of terms $\hat{H}_1$, since because of the periodic boundary conditions in time this is the only way to have a non-vanishing total weight. The third constraint comes from the t-ordered product in the Dyson series.

Examples of allowed configurations corresponding to the continuous-time path integral are those represented in Fig. 13.2.

Let us now discuss in detail how to devise a set of moves for all the previously listed quantities.

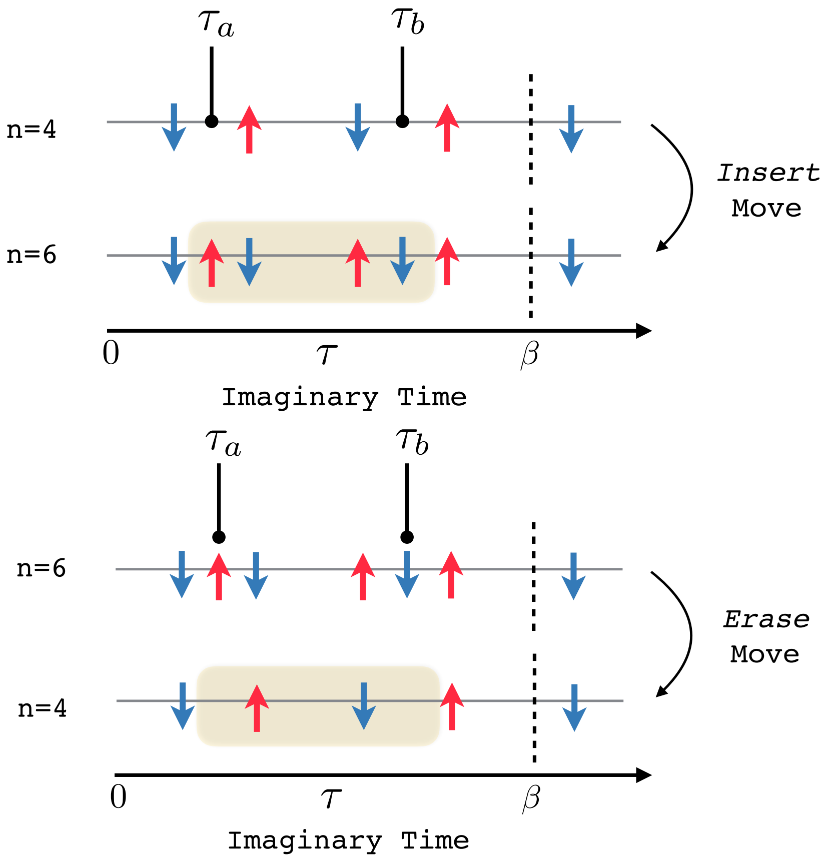


Figure 13.3: Example of moves in the continuos-time path-integral Monte Carlo. The left Figure shows the insertion of two spins at two random times $\tau_a < \tau_b$, followed (if necessary) by flipping the intermediate spins to satisfy the constraint. The right Figure shows the inverse move, in which two random spins at times τ_a and τ_b are randomly picked and deleted. This move is again followed by flipping any intermediate spin to satisfy the constraint in the final configuration.

13.2.1.1 Inserting and erasing pair of vertexes

Since the perturbation order can change only in multiples of 2, in our Monte Carlo sampling we should have a move that inserts/deletes pairs of terms $H_1(\tau_a)H_1(\tau_b)$. To this end, we generate two random times $\tau_a \in [0, \beta)$ and $\tau_b \in [0, \beta)$. Let us assume that $\tau_a < \tau_b$ (if it is not the case, just exchange the labels $a \leftrightarrow b$. Now, call $\tau_{\text{left}} < \tau_a$ the first time already existing in the path on the left of τ_a, and call $\tau_{\text{right}} > \tau_b$ the first time already existing in the path on the right of τ_b. In all the discussion we always assume periodic boundary conditions along the time direction, which also of course reflect on the correct ordering of the times.

When inserting a spin flip at τ_a we must set $\sigma^z_a = -\sigma^z_{\text{left}}$ and equally $\sigma^z_b = -\sigma^z_{\text{right}}$. If in the existing path we had intermediate times between τ_a and τ_b, say $\tau_a < \tau_j < \tau_{j+1} \dots < \tau_b$ then we must flip all the intermediate spins in order to satisfy the constraint that consecutive spins in imaginary times have opposite signs. An example of this operation is shown in Fig. 13.3.

13.3 Stochastic Series Expansion

The continuous-time formulation is not the only way of getting rid of the time-step error. An alternative approach is baed on the Taylor series for the matrix exponential:

$$e^{-\beta\hat{H}} = \sum_{n=0}^{\infty} \frac{(-\beta\hat{H})^n}{n!}, \tag{13.19}$$

and is dubbed “Stochastic Series Expansion”[2]. Using this expansion, we then just insert completeness at each order n, in a way that the partition function becomes:

$$\begin{aligned} Z &= \sum_{\mathbf{x}_1} \langle \mathbf{x}_1 | \hat{I} | \mathbf{x}_1 \rangle + \sum_{\mathbf{x}_1} \langle \mathbf{x}_1 | -\beta\hat{H} | \mathbf{x}_1 \rangle + \\ &\quad + \sum_{\mathbf{x}_1,\mathbf{x}_2} \frac{\langle \mathbf{x}_1 | -\beta\hat{H} | \mathbf{x}_2 \rangle \langle \mathbf{x}_2 | -\beta\hat{H} | \mathbf{x}_1 \rangle}{2} + \\ &\quad + \dots \end{aligned} \tag{13.20}$$

The probability distribution to be sampled then depends on the perturbation order and on the path configurations:

$$\begin{aligned} &\Pi_{\text{sse}}(\mathbf{x}_1 \dots \mathbf{x}_n, n) \propto \\ &\qquad = \frac{\beta^n}{n!} \langle \mathbf{x}_1 | -\hat{H} | \mathbf{x}_2 \rangle \dots \langle \mathbf{x}_{-1} | -\hat{H} | \mathbf{x}_n \rangle \langle \mathbf{x}_n | -\hat{H} | \mathbf{x}_1 \rangle. \end{aligned} \tag{13.21}$$

A Monte Carlo procedure in this case must then be able to propose changes in the perturbation order and in the configurations at different points in the SSE path.

For the previously considered example of the single-spin Hamiltonian, the SSE paths would not need to obey the constraint of alternating opposite spins $\sigma^z_j \neq \sigma^z_{(j+1) \bmod n}$, since the matrix elements entering the SSE weight are those of the full Hamiltonian and not only of the off-diagonal part $\hat{H}_1$. I leave as an exercise to think about possible ways of updating the SSE paths.

A point to be remarked here is that, albeit the sum over n in principle goes to infinity, in practice the number of orders that need to be sampled is finite, and can be done efficiently using Monte Carlo moves. A way to think about this is when sampling from the probability distribution of a Normal variable Normal(x), for which in principle we

[2] Anders W. Sandvik, Phys. Rev. E 68, 056701 (2003)

should generate values in the interval $x \in (-\infty, +\infty)$, however the probability of having very large (in modulus) values of x is exponentially suppressed, and typical samples will lie in a finite interval. The same thing happens for the discrete variable $n \in [0, \infty)$, which in fact has a finite average value $|\langle n \rangle| = \beta \langle H \rangle \propto N\beta$ and a finite variance $\langle n^2 \rangle - \langle n \rangle^2 \propto N\beta$. These two last relations can be easily derived directly considering the definition of the expectation values, for example we have the very elegant relation:

$$
\begin{aligned}
\langle \hat{H} \rangle &= \frac{1}{Z} \sum_{n=0}^{\infty} \sum_{\mathbf{x_1}} \frac{\beta^n}{n!} \langle \mathbf{x}_1 | (-\hat{H})^n \hat{H} | \mathbf{x}_1 \rangle && (13.22) \\
&= -\frac{1}{Z} \sum_{n=0}^{\infty} \sum_{\mathbf{x_1}} \frac{\beta^n}{n!} \langle \mathbf{x}_1 | (-\hat{H})^{n+1} | \mathbf{x}_1 \rangle && (13.23) \\
& \quad -\frac{1}{Z} \sum_{n=1}^{\infty} \sum_{\mathbf{x_1}} \frac{\beta^{n-1}}{n-1!} \langle \mathbf{x}_1 | (-\hat{H})^n | \mathbf{x}_1 \rangle && (13.24) \\
&= -\frac{1}{Z\beta} \sum_{n=1}^{\infty} \sum_{\mathbf{x_1}} \frac{\beta^n n}{n!} \langle \mathbf{x}_1 | (-\hat{H})^n | \mathbf{x}_1 \rangle && (13.25) \\
&= -\frac{\langle n \rangle}{\beta}. && (13.26)
\end{aligned}
$$

Chapter 14

Quantum computing

14.1 Quantum bits and quantum gates

In 1982 Feynman suggested that the problem of exponential complexity of simulating a quantum system can be solved by using quantum mechanics itself for computing, thus laying the foundation for the field of quantum computing [1]. Just as there are many ways to build a classical computer and lots of different conventions, also quantum computers could be designed in many different ways. Since we don't have any large scale quantum computer yet we are free to choose a design that is simple from a theoretical point of view.

14.1.1 Quantum bits

The basic memory element is typically chosen as the quantum bit, or qubit for short – a two-level system like a spin-1/2, where we associate the up spin state with the 0 bit and the down spin state with the 1 bit:

$$|0\rangle = |\uparrow\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{14.1}$$

$$|1\rangle = |\downarrow\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{14.2}$$

Just like for quantum spin-1/2s the quantum bit can exist in an arbitrary superposition of these two states:

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{14.3}$$

where the normalization condition requires that $|\alpha|^2 + |\beta|^2 = 1$. While such a state needs an infinite number of classical bits to be described accurately (think of the binary representation of α and β), a measurement will only give a single bit of information, either 0 or 1. A register of N qubits can store the wave function of N spin-1/2s. This gives an exponential advantage in memory use. However, since we can only do one measurement on each qubit only N bits of information can ever be read out. This is the first restriction we have to face when devising quantum algorithms, for which a clever use of these quantum bits needs to be devised.

Gate Name	Symbol	Matrix Form
Pauli-X (NOT)	$-\boxed{X}-$	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
Pauli-Y	$-\boxed{Y}-$	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$
Pauli-Z	$-\boxed{Z}-$	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
Hadamard gate	$-\boxed{H}-$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
Phase gate	$-\boxed{S}-$	$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$
T gate or $\pi/8$ gate	$-\boxed{T}-$	$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi 4} \end{pmatrix}$
Rz(θ) gate	$-\boxed{\mathrm{Rz}(\theta)}-$	$\begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}$

Table 14.1: List of commonly used single-qubit gates

14.1.2 Quantum gates

Since quantum mechanical time evolution is unitary (apart from measurements that collapse the wave function), we can only perform unitary operations on quantum bits and measurements. This is the second big restriction.

Just as for classical computers it is convenient to build a quantum circuit from a set of quantum gates that act on a limited set of qubits. Classical circuits are typically built from a set of gates that include OR, AND, NOT, XOR and more. However, in principle only the NAND (not-AND) gate is necessary since all other gates can be built from it. The NAND gate is thus called universal: any classical computation can be done purely with NAND gates, and any boolean function can be written in terms of NAND gates. It still makes sense however to consider more types of gates when building circuits, to make the design of circuits easier.

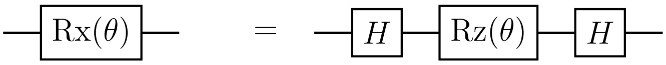
For quantum circuits one similarly often uses a larger set of gates than is strictly necessary. In the following we will discuss a set of typically used one and two qubit gates and will then discuss which ones are strictly necessary.

14.1.2.1 Single qubit gates

A few remarks may be useful. The X gate is the quantum analog of a classical NOT gate. The Hadamard gate (H) squares to the identity and is essentially a ninety degree rotation around the y axis, rotating a state aligned with z to x. The T gate is also called $\pi/8$ gate since it can – up to an irrelevant global phase – be written as

$$T = e^{i\pi/8} \begin{pmatrix} e^{-i\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{pmatrix}. \tag{14.4}$$

The Rz gate performs a rotation around the z axis in spin space. Similar rotations around the x and y axis are performed by the Rx and Ry gates. For example, a rotation around the x axis can be performed by swapping z and x by a Hadamard gate, performing a rotation around the z axis, and then rotating back:

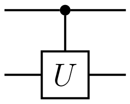


(14.5)

14.1.2.2 Two-qubit gates

A set of common two-qubit gates are controlled gates, consisting of a control qubit and a target qubit. The controlled version CU of a single qubit gate U (any of the list above) performs the single qubit operation U on the target qubit only if the control qubit is set to 1.

The quantum circuit for such a gate is:

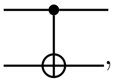


(14.6)

Denoting the matrix representation of the gate U as $\mathcal{U}$, the matrix representation of CU in a basis $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ is

$$\left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & \multicolumn{2}{c}{\multirow{2}{*}{$\mathcal{U}$}} \\ 0 & 0 & & \end{array}\right). \tag{14.7}$$

Maybe the most important two-qubit gate is the controlled-NOT-gate (CNOT), which is the same as a controlled-X gate. It is typically drawn as:

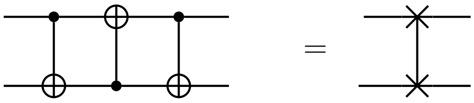


, (14.8)

its matrix representation is

$$\mathrm{CNOT} \doteq \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{14.9}$$

Other two-qubit gates can be built from single qubit gates and the CNOT gate. For example, the SWAP gate, which swaps the states of two qubits, can be built from three CNOT gates as:



(14.10)

In matrix representation the previous circuit corresponds to:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{14.11}$$

The last gate corresponds to the SWAP operation, indeed one can immediately verify that its action corresponds to

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_{00} \\ c_{01} \\ c_{10} \\ c_{11} \end{pmatrix} = \begin{pmatrix} c_{00} \\ c_{10} \\ c_{01} \\ c_{11} \end{pmatrix}. \tag{14.12}$$

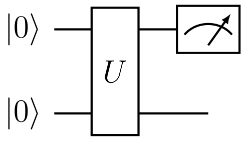
14.1.2.3 Universal gate sets

Of the above gates just the Hadamard, $\pi/8$ and CNOT gates are sufficient to implement any quantum circuit. All the other gates can be built from these gates, similar to the NAND gate being universal for classical computing.

The tricky part is how to represent arbitrary rotations using a discrete gate set. The Solovay-Kitaev algorithm allows to approximate arbitrary rotations to within any desired accuracy ϵ, with just poly(log($1/\epsilon$) gates.[1] Better algorithms for approximation of rotations have recently been invented and this is still an interesting field of research.

14.1.3 Measurements

Measurements in quantum circuits can be done on an arbitrary number of qubits, and the conventional setting of quantum computing is to consider measurements of qubits in the computational basis, thus corresponding to measuring the spin in the Z direction. Diagrammatically, measurements are indicated as in the figure below, that shows a measurement of the uppermost qubit:

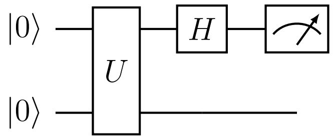


(14.13)

with a probability of observing one of the two possible outcomes (+1 or −1) given in this case by

$$P(s_1 = \pm 1) \ = \ \sum_{s_2=\pm 1} |\langle s_1, s_2 | \hat{U} | 0, 0 \rangle|^2 \tag{14.14}$$

Measurements in other directions are possible after applying suitable rotations of the spin, for example measuring in the X direction can be done by performing a rotation through the Hadamard gate. The circuit below shows a measurement of this type:

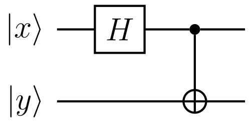


(14.15)

[1] The notation poly(x) indicates an effort that is polynomial in x.

14.1.4 Example: Creating entangled quantum states

As a first simple quantum algorithm that can be constructed using the build blocks discussed above, we will consider the task of creating entangled quantum states. Those can be very useful, for instance, in quantum communication tasks. One of the most prototypical entangled states are Bell states, also called EPR pairs after the Einstein-Podolsky-Rosen paper. These are produced by the following circuit:



$$= \qquad |\beta_{x,y}\rangle \tag{14.16}$$

Starting from an initial state $|\psi_0\rangle = |x\rangle\,|y\rangle$, where x and y are either 0 or 1 and not in superposition, the circuit produces a final state

$$|\beta_{xy}\rangle \equiv \frac{|0,y\rangle + (-1)^x|1,\overline{y}\rangle}{\sqrt{2}}. \tag{14.17}$$

For pure initial states $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$ we obtain the four Bell states called $|\beta_{00}\rangle$, $|\beta_{01}\rangle$, $|\beta_{10}\rangle$ and $|\beta_{11}\rangle$. The four possible outcomes are summarized in the Table below:

In	Out
$\lvert 00\rangle$	$\frac{\lvert 00\rangle+\lvert 11\rangle}{\sqrt{2}}$
$\lvert 01\rangle$	$\frac{\lvert 01\rangle+\lvert 10\rangle}{\sqrt{2}}$
$\lvert 10\rangle$	$\frac{\lvert 00\rangle-\lvert 11\rangle}{\sqrt{2}}$
$\lvert 11\rangle$	$\frac{\lvert 01\rangle-\lvert 10\rangle}{\sqrt{2}}$

Table 14.2: Input-Output table for the circuit creating Bell states.

14.2 Simulating the dynamics of quantum systems

14.2.1 Time evolution of a quantum spin model

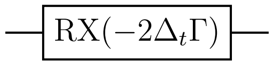
Exponential speedup over classical computers can be obtained for the simulation of the dynamics of quantum systems. As an example, we will consider once more the transverse field Ising model. Notation-wise, we will adopt a calligraphic notation for the Hamiltonian to avoid confusion with the Hadamard gate and we will also use the notation for Pauli matrices normally adopted in quantum computing: $\hat{Z}_i = \hat{\sigma}_i^z$, $\hat{X}_i = \hat{\sigma}_i^x$, $\hat{Y}_i = \hat{\sigma}_i^y$, thus our Hamiltonian is

$$\hat{\mathcal{H}} = -\Gamma \sum_i \hat{X}_i + \sum_{\langle i,j \rangle} J_{ij} \hat{Z}_i \hat{Z}_j. \tag{14.18}$$

In order to simulate the time evolution on a quantum computer we have to use a Trotter decomposition just like in the classical case, and again have the choice between simpler low-order approximations or more accurate high-order ones that are more complex but also more accurate.

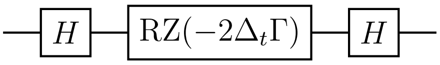
In this sense, the situation is entirely analogous to what we have seen when doing exact time evolution of quantum systems in the first lectures. The big advantage of quantum computers shows in the implementation of the individual terms of the Trotterized time evolution, which is now much easier. First, we need just N qubits instead of 2^N complex numbers in a classical code and requires only $\mathcal{O}(N)$ instead of $\mathcal{O}(2^N)$ operations.

The time evolution under the transverse field term $e^{i\Delta_t \Gamma \sigma_i^x}$ is trivial to implement, since it is just a rotation around the x axis, implemented by an RX(θ) gate with an angle $\theta = -2\Delta_t\Gamma$:



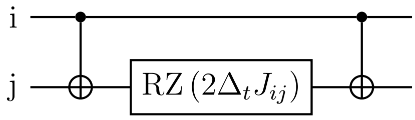
(14.19)

If we the quantum hardware does not offer an RX gate, but only (for example) arbitrary rotations around the z directions through the RZ(θ), then a basis transformation will be needed. It is easy to convince oneself that the Hadamard gate is the unitary matrix that transforms from the Z basis to the X basis, thus the Trotter step associated to the dynamics a single spin under the transverse field can be written as :



(14.20)

The Ising term is a 2-spin coupling and requires a 2-qubit gate. To implement $e^{-i\Delta_t J_{ij} \hat{\sigma}_i^z \hat{\sigma}_j^z}$ one needs to rotate by an angle $-\Delta_t J_{ij}$ if the two spin values are the same and $+\Delta_t J_{ij}$ if they differ. The following simple circuit can realize this operation:



(14.21)

Similar circuits can be designed for other quantum models, as we will do in the exercises for a quantum Heisenberg model.

14.3 Quantum Phase Estimation

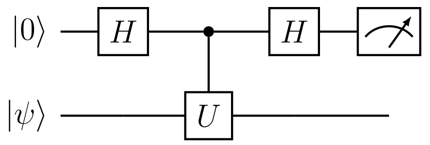
14.3.1 Measuring energies from phases

We now discuss an algorithm to estimate the energy of a given quantum state. The most straightforward way of measuring the energy of a quantum state $|\psi\rangle$ is to measure all the terms that make up the Hamiltonian $\hat{\mathcal{H}}$ and thus evaluate $\langle\psi|\hat{\mathcal{H}}|\psi\rangle$. However, this approach has several disadvantages. The wave function $|\psi\rangle$ gets destroyed with every measurement and we get only N bits of information. As this approach is similar to Monte Carlo sampling, we need $\mathcal{O}(\epsilon^{-2})$ measurements and thus $\mathcal{O}(\epsilon^{-2})$ preparations of the wave function $|\psi\rangle$ to measure the energy to an accuracy ϵ.

An alternative is to measure the phase which a state $|\psi\rangle$ picks up under time evolution with $\hat{\mathcal{H}}$. Let us first assume that $|\psi\rangle = |E_n\rangle$ be an eigenstate $|E_n\rangle$ of $\hat{\mathcal{H}}$ with eigenvalue E_n. Under time evolution $e^{-i\hat{\mathcal{H}}t}|\psi\rangle = e^{-iE_nt}|E_n\rangle$ the state picks up a phase E_nt. Measuring this phase would thus allow to measure the energy.

14.3.2 Quantum phase estimation algorithm

But, how do we measure the phase under time evolution? At first sight the phase is not an observable quantity. However, we can set up an "interference experiment" to determine the phase. We add an auxiliary qubit and perform the evolution under $\hat{U} = e^{-i\hat{\mathcal{H}}t}$ only if the auxiliary qubit is on:



(14.22)

Let us analyze step by the step what this circuit does. First, we start from $|0\rangle|\psi\rangle$ and apply a Hadamard gate to the auxiliary qubit, giving

$$\frac{1}{\sqrt{2}}\left(|0\rangle|\psi\rangle + |1\rangle|\psi\rangle\right) \tag{14.23}$$

We then apply the evolution controlled by the auxiliary qubit:

$$\frac{1}{\sqrt{2}}\left(|0\rangle|\psi\rangle + |1\rangle\hat{U}|\psi\rangle\right). \tag{14.24}$$

If $|\psi\rangle$ is an eigenstate $|E_n\rangle$ we pick up the corresponding phase $\phi = E_nt$

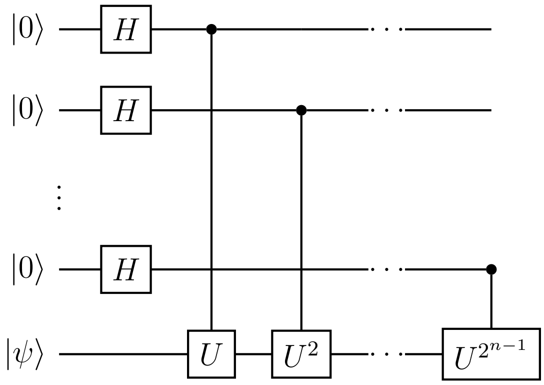
$$\frac{1}{\sqrt{2}}\left(|0\rangle|\psi\rangle + e^{-i\phi}|1\rangle|\psi\rangle\right). \tag{14.25}$$

Measuring now would not give us any information since the phase cannot be determined from a direct measurement. However, we can interfere the two cases by another Hadamard transform on the auxiliary qubit, obtaining

$$\frac{1}{2}\left[\left(1 + e^{-i\phi}\right)|0\rangle|\psi\rangle + \left(1 - e^{-i\phi}\right)|1\rangle|\psi\rangle\right]. \tag{14.26}$$

Now we measure the auxiliary qubit and we notice that the probability of measuring 0 is $(1+\cos\phi)/2 = \cos(\phi/2)^2$. By repeating then this experiment many times we can experimentally reconstruct (just counting how many times we measure 0) the probability of measuring 0, and consequently reconstruct ϕ. Using the histogram method, this probability can be determined to an accuracy of $1/\sqrt{M}$, if M measurements are realized. Thus, one needs to go through at least $M \sim 2^{2m}$ independent rounds of measurements to obtain m accurate binary digits of ϕ.

A more efficient version of this algorithm is due to Kitaev. We only sketch here the idea of this improved approach, that performs time evolutions with times $2^k t$ ($k = 1, \dots, n$), that are powers of 2, by adding n auxiliary qubits rather than one as in the simpler approach discussed previously. One starts by preparing the state as given by the circuit below:



(14.27)

This is a generalization of the case with a single auxiliary qubit, and we can readily see that the output state of the circuit above is

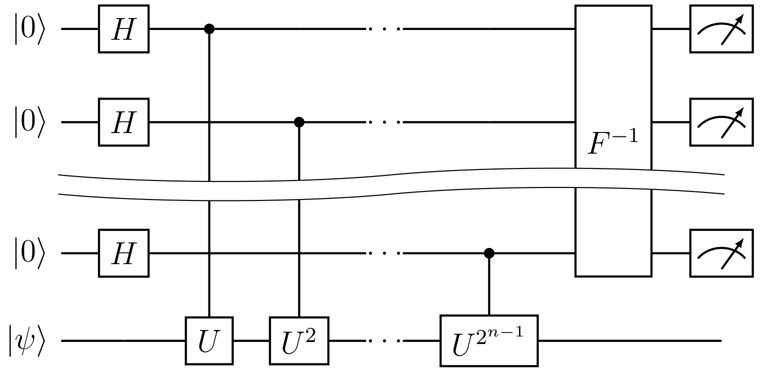
$$\frac{1}{\sqrt{2^n}}\left(|0\rangle + e^{-i\phi}|1\rangle\right) \otimes \left(|0\rangle + e^{-i2\phi}|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{-i2^{n-1}\phi}|1\rangle\right) \otimes |\psi\rangle. \tag{14.28}$$

We therefore end up with a register containing the Fourier transform of the phase:

$$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{-i\phi k} \, |k\rangle \, |\psi\rangle \tag{14.29}$$

Employing an inverse quantum Fourier transform F^{-1} (which is a unitary operation and can be implemented efficiently on a quantum computer) we measure a binary representation $\pi x/2^n$ of the phase ϕ.

The resulting algorithm is written in diagrammatic form below:



(14.30)

The main advantage of Kitaev's phase estimation approach is that it allows to reconstruct the phase much more accurately than using a single auxiliary qubit, since we can determine its value "digit-by-digit" in binary form through the inverse Fourier transform.

14.3.3 Quantum phase estimation to find the exact eigenvalues

It is especially interesting to use the quantum phase estimation algorithm in the case when $|\psi\rangle$ is not an exact eigenstate of the Hamiltonian, but rather a generic state. We can then expand it in the the eigenbasis of $\hat{\mathcal{H}}$:

$$|\psi\rangle \;=\; \sum_{l=0}^{2^N-1} c_l|E_l\rangle. \qquad (14.31)$$

Since all the operations we have carried on in the quantum phase estimation are linear, it is not hard to convince one-self that if we apply the controlled time evolutions to an approximate state we get

$$\frac{1}{\sqrt{2^n}}\sum_{l=0}^{2^N-1} c_l\left(|0\rangle+e^{-iE_lt}|1\rangle\right)\otimes\left(|0\rangle+e^{-i2E_lt}|1\rangle\right)\otimes\cdots\otimes\left(|0\rangle+e^{-i2^{n-1}E_lt}|1\rangle\right)\otimes|E_l\rangle =$$

$$= \frac{1}{\sqrt{2^n}}\sum_{l=0}^{2^N-1}\sum_{k=0}^{2^n-1} c_l e^{-i(E_lt)k}|k\rangle|E_l\rangle. \qquad (14.32)$$

Further applying the inverse Fourier transform to this state then will return frequencies that correspond to the exact eigen-energies of the Hamiltonian! The intensity of these peaks will be proportional to the coefficients $|c_l|^2$. We therefore see that if we have access to a reasonable approximation of the ground state $|\psi\rangle$ such that its overlap with the exact ground state is not exponentially small, then by performing quantum phase estimation over such state will give us the exact energies of the system. In this sense, QPE is a very powerful technique. The important caveat, however, is that in general it is not easy to prepare a simple state $|\psi\rangle$ that has a sizable overlap with the exact ground state. In general, finding such an initial state is an exponentially hard task, however we will analyze strategies to do so in the next lecture.

Bibliography

[1] Feynman, R.P., 1982. Simulating physics with computers. Int J Theor Phys 21, 467–488.

[2] Kaye, P., Laflamme, R., Mosca, M., 2007. An Introduction to Quantum Computing, 1st edition. ed. Oxford University Press, Oxford.

[3] Rieffel, E.G., Polak, W.H., 2011. Quantum Computing: A Gentle Introduction, Scientific and Engineering Computation. MIT Press, Cambridge, MA, USA.

Chapter 15

Variational Quantum Algorithms

15.1 Variational State Preparation

Here we describe approaches that aim at preparing a quantum state with a variational approach on a quantum computer. This general family of algorithms is based on variational principles and intrinsically demands a hybrid classical-quantum approach, as we will see in the following.

Generally speaking, the idea of these methods is to reformulate the task of preparing a certain quantum state as a carefully chosen optimization problem.

In the general setting we have three main ingredients:

1. A parameterized quantum circuit (sometimes also improperly called *quantum neural networks*), consisting of a sequence of unitaries parameterize by some $\theta = (\theta_1 \dots \theta_l)$. In the following we assume the form

   $$\hat{U}(\theta) \quad = \quad \hat{U}_l(\theta_l) \dots \hat{U}_2(\theta_2) \hat{U}_1(\theta_1), \tag{15.1}$$

   thus preparing a family of *variational* states

   $$|\Psi(\theta)\rangle \quad = \quad \hat{U}(\theta)|0\rangle. \tag{15.2}$$

2. A "loss function" that is representative of the task one wants to solve. The choice of the loss function is problem dependent, however in several applications it takes the form of an expectation value of an Hermitean operator B

   $$\mathcal{L}(\theta) \quad = \quad \langle 0|\hat{U}^\dagger(\theta)\hat{B}\hat{U}(\theta)|0\rangle, \tag{15.3}$$

   and the best variational approximations coincides with the set of parameters that minimize the loss function.

3. A classical optimizer, that uses suitable measurements from the quantum computer in order to find a good approximation for the minimum of the loss function, i.e.

   $$\bar{\theta} \quad = \quad \mathrm{argmin}_\theta \mathcal{L}(\theta). \tag{15.4}$$

15.2 Loss Functions

We now present several interesting problems in physics, chemistry, and mathematics that can be formulated through a variational approach. The main conceptual requirement in this phase is rewriting the specific problem at hand as a suitable optimization problem, for which it is therefore important to:

1. Identity a suitable loss function $\mathcal{L}(\theta)$ such that, in the ideal case, the state $|\Psi(\bar{\theta})\rangle$ prepares the given target quantum state;
2. Make sure that the loss function is efficiently measurable on quantum hardware;
3. Make sure that sufficient progress is made in optimizing the loss.

15.2.1 Ground-State Preparation

In the context of quantum simulation, a very important task is preparing the ground state of a given hamiltonian, $\hat{\mathcal{H}}$. In this case, the most natural loss function for state preparation is the total energy of the quantum system:

$$E(\theta) \; = \; \langle 0|\hat{U}^{\dagger}(\theta)\hat{\mathcal{H}}\hat{U}(\theta)|0\rangle. \tag{15.5}$$

From the variational principle of quantum mechanics we know that

$$E(\theta) \; \geq \; E_0, \tag{15.6}$$

where E_0 is the exact ground state energy. Thus by minimizing $E(\theta)$ we seek to approximate the ground state of a given Hamiltonian using a fixed-depth quantum circuit. This algorithm is known as the Variational Quantum Eigensolver (VQE).

In practical applications, one is interested in Hamiltonians that are sum of local operators, most prominently taking the form of sum of “Pauli strings” :

$$\hat{\mathcal{H}} \; = \; \sum_k c_k \hat{A}_{k,1}\hat{A}_{k,2}\dots\hat{A}_{k,N}, \tag{15.7}$$

with the coefficients $c_k \in \mathbb{R}$ and $A_{k,i} \in [I, X, Y, Z]$. To measure the total energy $E(\theta)$ then a simple approach is to measure the Pauli strings

$$\hat{P}_k \; = \; \hat{A}_{k,1}\hat{A}_{k,2}\dots\hat{A}_{k,N}, \tag{15.8}$$

in the σ^z computational basis, after a suitable basis rotation.

To give a concrete example, imagine that we would like to find the ground state of our best friend, the Transverse Field Ising Model (TFIM) :

$$\hat{\mathcal{H}} = -\Gamma\sum_i \hat{X}_i + \sum_{\langle i,j\rangle} J_{ij}\hat{Z}_i\hat{Z}_j. \tag{15.9}$$

In this case, to estimate the expectation value of the energy we need to measure for example $P_k = \hat{X}_k$. This can be done noticing that

$$\langle\hat{X}_k\rangle \; = \; \langle 0|\hat{U}^{\dagger}(\theta)\hat{H}_k\hat{Z}_k\hat{H}_k\hat{U}(\theta)|0\rangle, \tag{15.10}$$

where we do a simple basis rotation through the Hadamard gate:

$$\hat{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

In conclusion, all the terms of the energy that include the transverse field parts can be simply measured by collecting the output of the circuit depicted in Fig. 15.1.

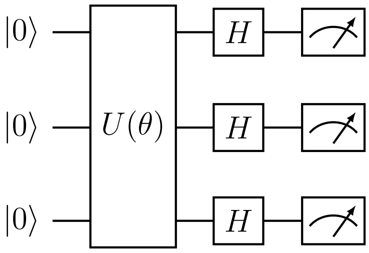


Figure 15.1: Example of circuit to measure $\sum_k \langle \hat{X}_k \rangle$ on the variational state.

The accuracy of this estimate will depend on the number of measurements (or shots) taken, N_s, with a statistical error decreasing like $1/\sqrt{N_s}$.

15.2.2 Excited States

There have been a few proposals of cost functions to find excited states of a given Hamiltonian. It is fair to say that this is still a matter of research activity, thus we sketch here only the conceptually simplest approach (albeit not necessarily a very efficient one). The idea is to first prepare an approximation of the ground state, such that $|\Psi(\bar{\theta}_0)\rangle \simeq |\Psi_0\rangle$, then one considers the cost function

$$E_1(\theta, \lambda) = \langle 0|\hat{U}^\dagger(\theta)\hat{\mathcal{H}}\hat{U}(\theta)|0\rangle + \lambda |\langle \Psi(\bar{\theta}_0)|\hat{U}(\theta)|0\rangle|^2, \tag{15.11}$$

where we have introduced a Lagrange multiplier λ, such that the resulting state $|\Psi(\theta_1^\star)\rangle = \hat{U}(\theta_1^\star)|0\rangle$ will be state of lowest energy that is also orthogonal to the given approximation of the ground state, i.e. $\langle \Psi(\bar{\theta}_0)|\Psi(\bar{\theta}_1)\rangle = 0$. The first term in this cost function can be estimated as much as in standard VQE, whereas the second term can be estimated noticing that

$$|\langle \Psi(\bar{\theta}_0)|\hat{U}(\theta)|0\rangle|^2 = |\langle 0|\hat{U}^\dagger(\bar{\theta}_0)\hat{U}(\theta)|0\rangle|^2, \tag{15.12}$$

is the probability of measuring $|0\rangle$ after having prepared the state

$$|\Phi\rangle = \hat{U}^\dagger(\bar{\theta}_0)\hat{U}(\theta)|0\rangle, \tag{15.13}$$

thus requiring a doubling of the circuit depth, as shown in Figure 15.2.

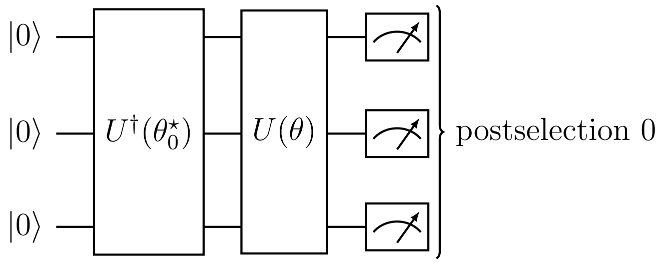


Figure 15.2: Example circuit to measure the overlap between two parameterized circuits.

15.3 Parameterized Quantum Circuits

Here we give a few example of the specific form that parameterized quantum circuits can take, especially in the context of ground state problems. In order to get some insight, it is useful first to recall how, at least in principle, one can prepare the ground state of a given Hamiltonian using purely unitary time evolution.

15.3.1 Ground states through adiabatic state preparation

Adiabatic state preparation is based on the quantum adiabatic theorem. We assume that at $t = 0$ we start in the ground state $|\psi_0(t)\rangle$ of some time-dependent Hamiltonian $\hat{\mathcal{H}}(t)$. If we change the Hamiltonian $\hat{\mathcal{H}}(t)$ adiabatically slowly we will remain in the ground state $|\psi_0(t)\rangle$ of $\hat{\mathcal{H}}(t)$ for all times $t > 0$. This opens a way to prepare the ground state of a quantum system. Let us start at time $t = 0$ in a Hamiltonian $\hat{\mathcal{H}}_0$ of which we can easily compute the ground state. In order to find the ground state of an unknown Hamiltonian $\hat{\mathcal{H}}_f$ we adiabatically interpolate between $\hat{\mathcal{H}}$ and $\hat{\mathcal{H}}_f$ to arrive at the desired Hamiltonian at time t_f:

$$\hat{\mathcal{H}}(t) = \left(1 - \frac{t}{t_f}\right)\hat{\mathcal{H}}_0 + \frac{t}{t_f}\hat{\mathcal{H}}_f. \tag{15.14}$$

Instead of this linear interpolation any other function can be chosen as long as $\hat{\mathcal{H}}(0) = \hat{\mathcal{H}}_0$ and $\hat{\mathcal{H}}(t_f) = \hat{\mathcal{H}}_f$. If we choose t_f long enough we are guaranteed to end up in the ground state with only exponentially small errors.

What is meant by “long enough”? The quantum adiabatic theorem states that this time should be much longer than a scale set by the minimum gap:

$$t_f \gg \mathcal{O}\left(\min_t \Delta(t)^{-2}\right), \tag{15.15}$$

where the gap

$$\Delta(t) = E_1(t) - E_0(t) \tag{15.16}$$

is the difference between the ground state energy $E_0(t)$ and the energy of the first excited state $E_1(t)$. Since in practice we know neither the minimum gap of the unknown quantum system, nor the constants that go into this inequality, we will have to perform numerical experiments on our (quantum) computer to determine when the results start to converge as a function of t_f. It is important to remark that for generic physical hamiltonians we expect the time $t_f \sim \exp(N)$, as a result of exponentially closing gap during the adiabatic state preparation. There is no known classical or quantum algorithm that can efficiently prepare the ground state of a generic physical hamiltonian in polynomial time, and the adiabatic algorithm is no exception.

15.3.2 Quantum alternating operator

The adiabatic state preparation gives a hint at what kind of ansatz we can take to study ground state problems. Let us consider the case in which the time-dependent Hamiltonian takes the generic form

$$\hat{\mathcal{H}}(t) = \sum_s^M \hat{\mathcal{H}}_s(t), \tag{15.17}$$

where the terms $[\hat{\mathcal{H}}_s, \hat{\mathcal{H}}_{s'}] \neq 0$ in general but are chosen in a way that each $\mathcal{H}_s$ contains all terms mutually commuting. In general each term can contain both parts of $\hat{\mathcal{H}}_0$ and $\hat{\mathcal{H}}_f$.

In order to implement the adiabatic preparation algorithm on a quantum computer, we need to write the time evolution induced by the Hamiltonian in terms of gates. A simple approach would be to consider a small time step δ_t such that

$$|\Psi(t)\rangle \simeq \Pi_p^{[t/\delta_t]} e^{-i\delta_t \hat{\mathcal{H}}(p\delta_t)} |\Psi(0)\rangle. \tag{15.18}$$

Then the small time-step propagators are written in terms of a Trotter decomposition, such that

$$|\Psi(t)\rangle \simeq \Pi_p^{[t/\delta_t]} e^{-i\delta_t \hat{\mathcal{H}}_1(p\delta_t)} e^{-i\delta_t \hat{\mathcal{H}}_2(p\delta_t)} \dots e^{-i\delta_t \hat{\mathcal{H}}_M(p\delta_t)} |\Psi(0)\rangle. \tag{15.19}$$

To give a concrete example, imagine that we would like to find the ground state of the Transverse Field Ising Model :

$$\hat{\mathcal{H}}_f = \sum_{\langle i,j \rangle} J_{ij} \hat{Z}_i \hat{Z}_j + \Gamma \sum_i \hat{X}_i. \tag{15.20}$$

In this case, we can define $\hat{\mathcal{H}}_0 = \Gamma \sum_i \hat{X}_i$, such that the ground state of this part, $|\psi_0(0)\rangle$, is easily prepared as a product state.

We split the Hamiltonian into $M = 2$ non-commuting terms, the classical Ising interaction

$$\hat{\mathcal{H}}_{zz} = \sum_{\langle i,j \rangle} J_{ij} \hat{Z}_i \hat{Z}_j \tag{15.21}$$

and the transverse field term

$$\hat{\mathcal{H}}_x = \Gamma \sum_i \hat{X}_i, \tag{15.22}$$

such that

$$\begin{aligned} \hat{\mathcal{H}}(t) &= \left(1 - \frac{t}{t_f}\right) \hat{\mathcal{H}}_x + \frac{t}{t_f} \left(\hat{\mathcal{H}}_x + \hat{\mathcal{H}}_{zz}\right) && (15.23) \\ &= \hat{\mathcal{H}}_x + \frac{t}{t_f} \hat{\mathcal{H}}_{zz} && (15.24) \\ &= \hat{\mathcal{H}}_1(t) + \hat{\mathcal{H}}_2(t). && (15.25) \end{aligned}$$

Each of these terms, when exponentiated, correspond to local unitaries. Since the Ising term gives a product of two-local unitaries that are diagonal in the computational basis

$$e^{-i\hat{\mathcal{H}}_2(t)\delta_t} = e^{-i\delta_t \frac{t}{t_f}\sum_{\langle i,j\rangle} J_{ij}\hat{Z}_i\hat{Z}_j} = \prod_{\langle i,j\rangle} e^{-i\left(\delta_t\frac{t}{t_f}\right)J_{ij}\hat{Z}_i\hat{Z}_j}. \tag{15.26}$$

The transverse field term splits into N commuting terms for each of the spins:

$$e^{-i\hat{\mathcal{H}}_1(t)\delta_t} = e^{-i\delta_t\Gamma\sum_i \hat{X}_i} = \prod_i e^{-i\delta_t\Gamma\hat{X}_i}. \tag{15.27}$$

The time evolution under the transverse field term $e^{-i\delta_t\Gamma\hat{X}_i}$ is trivial to implement, since it is just a rotation around the x axis, implemented by an RX gate, depicted in Fig. 15.3.

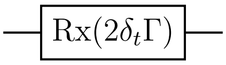


Figure 15.3: Simple rotation gate realizing $e^{-i\delta_t\Gamma\hat{X}_i}$.

The Ising term is a 2-spin coupling and thus requires a 2-qubit gate. To implement $e^{-i\left(\delta_t\frac{t}{t_f}\right)J_{ij}\hat{Z}_i\hat{Z}_j}$ one needs to rotate by an angle $\Phi_{ij}(t) = -\delta_t J_{ij} t/t_f$ if the two spin values are the same and $-\Phi_{ij}(t)$ if they differ. The circuit in Fig. 15.4 can realize this operation.

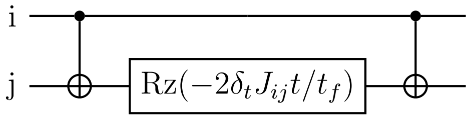


Figure 15.4: Circuit realizing RZZ $= e^{-i\left(\delta_t\frac{t}{t_f}\right)J_{ij}\hat{Z}_i\hat{Z}_j}$.

The resulting unitary preparing $|\psi(t)\rangle$ is then schematically shown in Figure 15.5.

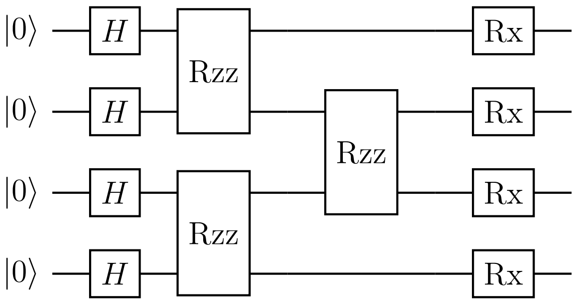


Figure 15.5: Circuit realizing the first Trotter step of adiabatic time evolution for the Transverse-Field Ising model.

It can be noticed that this circuit is not parametric, since all the gates are fixed. The main idea of QAOA/Hamiltonian Variational ansatz is to make these gates parametric:

$$|\Psi(\theta)\rangle \;=\; \Pi_{s,p} e^{-i\hat{\mathcal{H}}_p \theta_{sp}} |\Psi_0\rangle. \tag{15.28}$$

A nice way of interpreting this is in terms of a time evolution that can possibly break the adiabatic condition but that can in principle prepare the state faster (i.e. with a short circuit, as compared with the adiabatic state preparation that would take very deep circuits).

15.3.3 Hardware Efficient Ansatz

In addition to ansatz wave functions inspired by the adiabatic state preparation, another possible (and popular) choice is to to take parameterized circuits that are not necessarily physically motivated but that at least can be implemented efficiently on existing quantum hardware. Typically, this family of states takes the form

$$|\Psi(\theta)\rangle \;=\; \Pi_k \hat{U}_k(\theta_k)\hat{W}_k|0\rangle, \tag{15.29}$$

where the gates $\hat{W}_k$ are fixed and do not carry a variational parameter to be optimized. In several applications, for example, only single-qubit gates are parameterized and a fixed set of two-qubit gates, $\hat{W}_k$, are positioned on qubit edges that are consistent with the physical connectivity of the devices.

15.4 Optimization Algorithms

Here we discuss optimization strategies based on gradients. To make our discussion more concrete, we focus again on parameterized states

$$|\Psi(\theta)\rangle \;=\; \hat{U}_l(\theta_l)\ldots\hat{U}_2(\theta_2)\hat{U}_1(\theta_1)|0\rangle, \tag{15.30}$$

and loss functions of the form

$$\mathcal{L}(\theta) \;=\; \langle\Psi(\theta)|\hat{B}|\Psi(\theta)\rangle, \tag{15.31}$$

where B is some hermitean operator. We have seen in the previous discussion that several of the loss functions of interest can be written in this form.

In their simplest setting, gradient-based hybrid optimization algorithms do the following iterative procedure do minimize the loss function:

1. At iteration i, use the quantum computer to estimate $\mathcal{L}(\theta^{(i)})$ with the current set of variational parameters $\theta^{(i)}$ and the components of the gradient, $G_k(\theta^{(i)}) = \partial_{\theta_k}\mathcal{L}(\theta^{(i)})$;

2. Update the variational parameters according, for example, to a gradient descent step $\theta_k^{(i+1)} = \theta_k^{(i)} - \eta G_k(\theta^{(i)})$, where η is a small constant, known as the "learning rate", following standard machine learning terminology

3. Quit when converged to a minimum of the loss function.

15.4.1 Parameter Shift Rule

For specific applications, we have already seen how to compute the loss function efficiently, using measurements in the computational basis. We now describe also an efficient strategy, known in literature as the parameter shift rule, to estimate the gradient of the loss function. More details can be bound in Ref. [2].

For simplicity, we restrict ourselves to unitaries of the form

$$\hat{U}_k(\theta_k) \;=\; e^{-i\hat{S}_k\frac{\theta_k}{2}}, \tag{15.32}$$

where the operator S_k is taken to be involutory, i.e. satisfying $S_k^2 = \mathbb{I}$. Notice that this family of gates is quite general and encompasses for example all single qubit rotations generated by pauli operators, i.e. $S_k \in (X, Y, Z)$ as well as several two-qubit gates acting on qubits i and j such that $S_k \in (X_iX_j, Y_iY_j, \dots)$. Because of the involutory property, it is easy to verify that

$$\begin{aligned}
\hat{U}_k(\theta_k) &= \hat{I} - i\hat{S}_k\frac{\theta_k}{2} - \frac{1}{2!}\hat{S}_k^2\left(\frac{\theta_k}{2}\right)^2 + \frac{i}{3!}\hat{S}_k^3\left(\frac{\theta_k}{2}\right)^3 + \dots && (15.33)\\
&= \left[1 - \frac{1}{2!}\left(\frac{\theta_k}{2}\right)^2 + \frac{1}{4!}\left(\frac{\theta_k}{2}\right)^4 + \dots\right]\hat{I} + && (15.34)\\
&\quad -i\hat{S}_k\left[\frac{\theta_k}{2} - \frac{1}{3!}\left(\frac{\theta_k}{2}\right)^3 + \frac{1}{5!}\left(\frac{\theta_k}{2}\right)^5 + \dots\right] && (15.35)\\
&= \cos\left(\frac{\theta_k}{2}\right)\hat{I} - i\sin\left(\frac{\theta_k}{2}\right)\hat{S}_k, && (15.36)
\end{aligned}$$

we also notice that the unitary conjugation of an arbitrary operator can be written as

$$\begin{aligned}
\hat{K}(\theta_k) &= \hat{U}_k^\dagger(\theta_k)\hat{K}\hat{U}_k(\theta_k) && (15.37)\\
&= \hat{A} + \hat{B}\cos(\theta_k) + \hat{C}\sin(\theta_k), && (15.38)
\end{aligned}$$

where the operators $\hat{A}, \hat{B}, \hat{C}$ do not depend on the parameter. This last property is especially interesting for us, since the derivative is now easily found:

$$\begin{aligned}
\partial_{\theta_k}\hat{K}(\theta_k) &= -\hat{B}\sin(\theta_k) + \hat{C}\cos(\theta_k) && (15.39)\\
&= \frac{\hat{K}(\theta_k + \pi/2) - \hat{K}(\theta_k - \pi/2)}{2}. && (15.40)
\end{aligned}$$

Now, let us define $\hat{U}(\theta) = \hat{V}_k\hat{U}_k(\theta)\hat{W}_k$, where the operators $\hat{V}_k$ and $\hat{W}_k$ do not depend explicitly on the parameter θ_k but can depend on all the other parameters. With this definition, we can write the derivative of the loss explicitly

$$\begin{aligned}
\partial_{\theta_k}\mathcal{L}(\theta_1,\dots,\theta_k,\dots\theta_l) &= \partial_{\theta_k}\langle 0|\hat{W}_k^\dagger\hat{U}_k^\dagger(\theta_k)\hat{V}_k^\dagger\hat{B}\hat{V}_k\hat{U}_k(\theta_k)\hat{W}_k|0\rangle && (15.41)\\
&= \partial_{\theta_k}\langle\Psi_k|\hat{U}_k^\dagger(\theta_k)\hat{Q}_k\hat{U}_k(\theta_k)|\Psi_k\rangle && (15.42)\\
&= \partial_{\theta_k}\langle\Psi_k|\hat{Q}_k(\theta_k)|\Psi_k\rangle && (15.43)\\
&= \frac{1}{2}\left\{\langle\Psi_k|\hat{U}_k\left(\theta_k+\frac{\pi}{2}\right)^\dagger\hat{Q}_k\hat{U}_k\left(\theta_k+\frac{\pi}{2}\right)|\Psi_k\rangle+\right. && (15.44)\\
&\left.-\langle\Psi_k|\hat{U}_k\left(\theta_k-\frac{\pi}{2}\right)^\dagger\hat{Q}_k\hat{U}_k\left(\theta_k-\frac{\pi}{2}\right)|\Psi_k\rangle\right\} && (15.45)\\
&= \frac{\mathcal{L}(\theta_1,\dots,\theta_k+\frac{\pi}{2},\dots\theta_l)-\mathcal{L}(\theta_1,\dots,\theta_k-\frac{\pi}{2},\dots\theta_l)}{2} && (15.46)
\end{aligned}$$

This formula is very interesting because it tells us that the gradient can be computed just as the difference of two loss functions, thus it can be estimated just as done for the loss function itself.

15.5 Qubits Encodings

15.5.1 Fermions

Quantum computers intrinsically operate with spin degrees of freedom, since in the standard model of quantum computation basis states are eigenstates $|\uparrow\rangle, |\downarrow\rangle$ of the σ^z operator,

$$\begin{aligned}
|\uparrow\rangle &\equiv |0\rangle && (15.47)\\
|\downarrow\rangle &\equiv |1\rangle. && (15.48)
\end{aligned}$$

In Nature however particles can obey to different statistics, most notably interacting fermionic matter plays a crucial role in the determination of chemical and physical properties of materials, molecules and much more.

When studying electronic problems with variational quantum algorithms, it is then necessary to map these fermionic degrees of freedom into qubits. When doing exact diagonalization of fermions, we have faced exactly the same problem, when me mapped fermions onto spins. In general, we consider electronic Hamiltonians defined in terms of the usual fermionic operators $\hat{c}_i^\dagger$ and $\hat{c}_i$ satisfying the anti-commutation relations

$$\{\hat{c}_l^\dagger,\hat{c}_m\} = \delta_{lm}\hat{I}. \qquad (15.49)$$

The Jordan-Wigner mapping is one of the main (and the oldest) strategies to "convert" these fermionic operators into spins. This is achieved through the following transformation

$$\begin{aligned}
\hat{c}_l &= \left(\Pi_{j=1}^{l-1}\hat{Z}_j\right)\hat{\sigma}_l^- && (15.50)\\
\hat{c}_l^\dagger &= \left(\Pi_{j=1}^{l-1}\hat{Z}_j\right)\hat{\sigma}_l^+, && (15.51)
\end{aligned}$$

with the usual raising and lowering spin operators

$$\hat{\sigma}_l^+ \;=\; \frac{\hat{X}_l + i\hat{Y}_l}{2} \tag{15.52}$$

$$\hat{\sigma}_l^- \;=\; \frac{\hat{X}_l - i\hat{Y}_l}{2}, \tag{15.53}$$

that satisfy

$$\{\hat{\sigma}_l^+, \hat{\sigma}_l\} \;=\; \hat{I}. \tag{15.54}$$

It can be verified that the fermionic operators defined through the mapping satisfy the required commutation relations, for example

$$\begin{aligned} \{c_l^\dagger, c_l\} &= \{\left(\Pi_{j=1}^{l-1}\hat{Z}_j\right)\hat{\sigma}_l^+, \left(\Pi_{j=1}^{l-1}\hat{Z}_j\right)\hat{\sigma}_l^-\} && (15.55)\\ &= \left(\Pi_{j=1}^{l-1}\hat{Z}_j\right)^2 \{\hat{\sigma}_l^+, \hat{\sigma}_l\} && (15.56)\\ &= \left(\Pi_{j=1}^{l-1}\hat{Z}_j\right)^2, && (15.57) \end{aligned}$$

and the last term is nothing but the identity, since Pauli matrices square to one, thus $\left(\Pi_{j=1}^{l-1}\hat{Z}_j\right)^2 = \Pi_{j=1}^{l-1}\left(\hat{Z}_j\right)^2 = \hat{I}$.

With this mapping at hand, it is clear that we can write arbitrary fermionic hamiltonians in the form of sum of Pauli strings, and we can use all the techniques discussed above for spin/qubits hamiltonians.

Bibliography

[1] Bharti, K., Cervera-Lierta, A., Kyaw, T.H., Haug, T., Alperin-Lea, S., Anand, A., Degroote, M., Heimonen, H., Kottmann, J.S., Menke, T., Mok, W.-K., Sim, S., Kwek, L.-C., Aspuru-Guzik, A., 2022. Noisy intermediate-scale quantum algorithms. Rev. Mod. Phys. 94, 015004.

[2] Mari, A., Bromley, T.R., Killoran, N., 2021. Estimating the gradient and higher-order derivatives on quantum hardware. Phys. Rev. A 103, 012405.