Computational Quantum Physics
FS2023

Mark H Fischer
Department of Physics, University of Zürich

June 5, 2023

Based on lectures by Matthias Troyer, Alexey Soluyanov, Giuseppe Carleo, Sebastian Huber, Titus Neupert, Kiryl Pakrouski

If you notice mistakes or typos, please report them to bernhard.luescher@physik.uzh.ch.

Contents

1 Introduction

Computational physics generally concerns the solution of physics problems with the help of a computer. Such a problem can be the evaluation of a concrete property of the model, such as a scattering cross section or a susceptibility, to which no analytical solution exists. Often, however, the models considered are too complicated to even formulate such quantities. Then, the whole system (dynamics) is simulated, similar to an actual experiment. Due to this dichotomy, computational physics is often considered a third pillar of physics research, see Fig. 1.

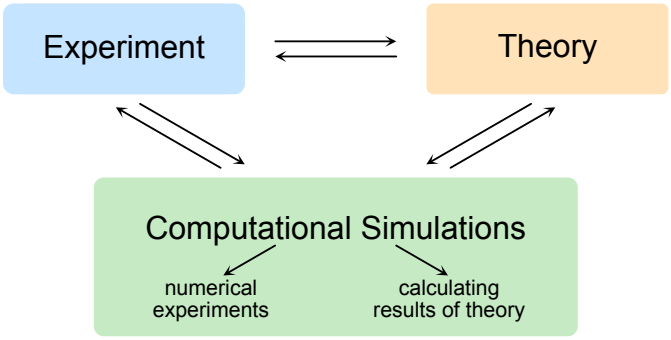


Figure 1: Role of computational simulations in modern physics.

When dealing with quantum mechanical problems—the subject of this course—problems emerge not encountered in classical mechanics. While in the latter, a collection of particles lives in the phase space, a quantum mechanical state is an element of an (exponentially) larger Hilbert space. In order to make progress, we thus need to decide what "kind of solution" we are interested in. For a single particle, we might be able to calculate the wave function or a scattering cross section exactly. For an interacting many-particle system, only approximate solutions are available, which vary depending on the properties of the system we are interested in. Figure 2 provides an overview over the various approaches and their organization within this course.

While the development of new methods and algorithms in computational quantum physics is an active field, the aim of this course is for students to understand the basic concepts and master the necessary tools for their own applications. Thus, after this course

- students are able to describe the difficulties of quantum-mechanical simulations,
- students are able to explain the strengths and weaknesses of the methods covered,
- students are able to select an appropriate method for a given problem,
- students are able to implement basic versions of all algorithms discussed.

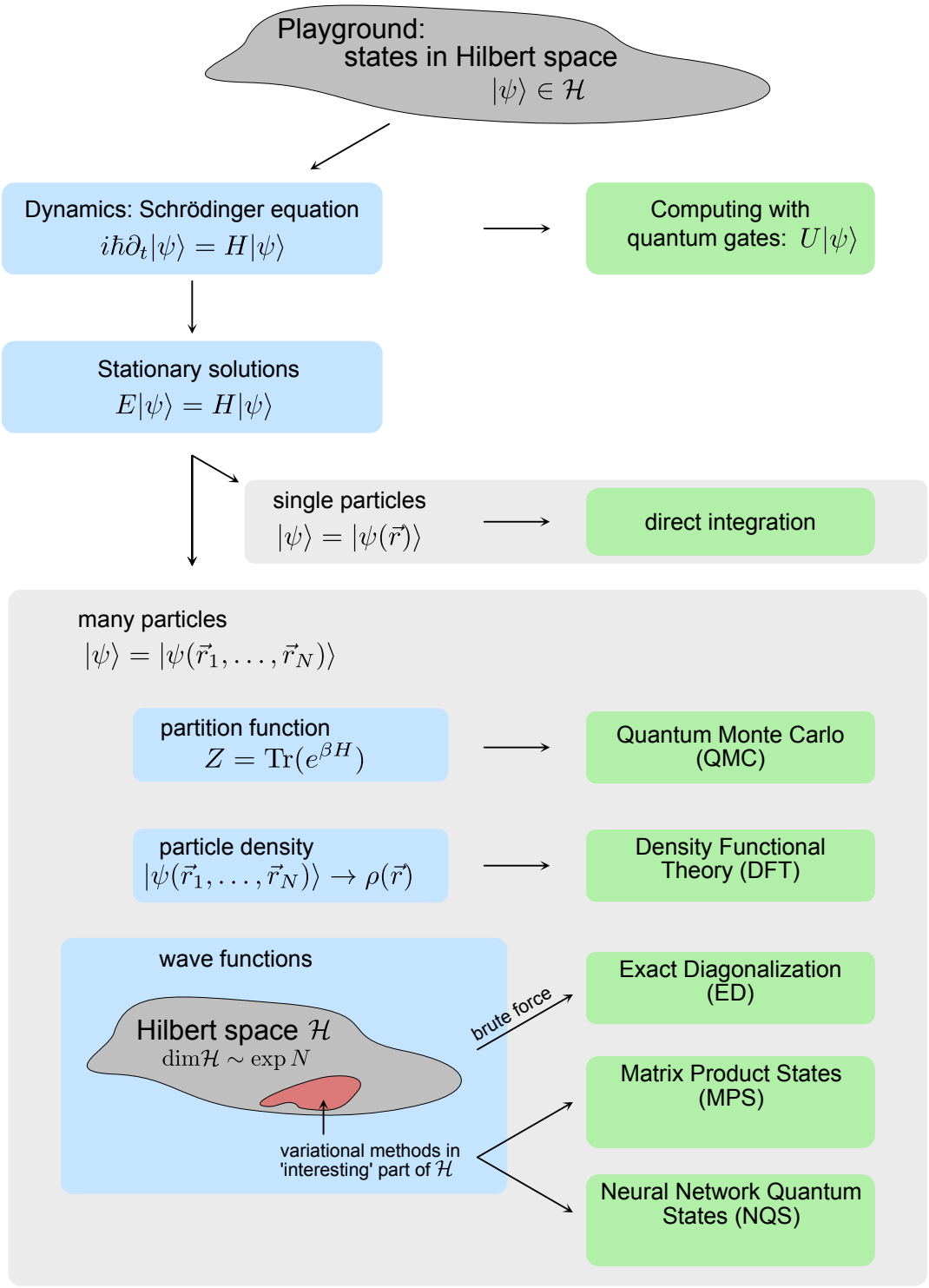


Figure 2: Overview of challenges and methods of computational quantum physics.

1.1 Prerequisites and tools

For this course, we assume the following prerequisites:

- While we will introduce the main concepts of quantum mechanics at the beginning of the semester to refresh your knowledge, a basic knowledge is assumed.
- Basic knowledge of numerical tools, such as numerical differentiation and integration, linear solvers, eigensolvers, root solvers, optimization.
- Basic knowledge of a programming language. For the teaching assignments, you are free to choose your preferred one. The solutions will typically be distributed in Python in the form of Jupyter notebooks.

Please, don't hesitate to ask questions if any notions are unclear.

1.2 References

For further reading, we recommend the following books and reviews:

- **General:** *Computational Physics*, Jos Thiisen (Cambridge University Press [1])
- **ED:** *Computational Studies of Quantum Spin Systems*, A. W. Sandwik, AIP Conference Proceedings **1297**, 135 (2010)
- **DMRG:** *The density-matrix renormalization group in the age of matrix product states*, U. Schollwöck, Annals of Physics **326**, 96 (2011)
- **Monte Carlo:** *Quantum Monte Carlo Approaches for Correlated Systems*, F. Becca and S. Sorella (Cambridge University Press)
- **DFT:** *Electronic Structure, Basic Theory and Practical Methods*, R. M. Martin (Cambridge University Press)

At the beginning of each chapter, we provide more specific or additional references if applicable.

[1] Note that publications by Cambridge University Press are available as online resources within the ETHZ network (including eduroam).

2 Quantum mechanics crash course

The following pages cannot replace a proper introduction to quantum mechanics. However, they serve as a refresher and should fix our notation.

2.1 The basics

2.1.1 Wave functions and the Hilbert space

Quantum mechanics is in essence linear algebra. The ingredients are simple: A pure state of a quantum system is described by a wave function $|\psi\rangle$, which is an element of a Hilbert space

$$|\psi\rangle \in \mathcal{H}. \tag{2.1}$$

As the wave function has the interpretation of a probability distribution, we typically work with normalized wave functions

$$\||\psi\rangle\| \equiv \sqrt{\langle\psi|\psi\rangle} = 1. \tag{2.2}$$

Here and in the following, we use the bra-ket notation and $\langle\psi|\varphi\rangle$ is the scalar product of two elements $|\psi\rangle$ and $|\varphi\rangle$ in $\mathcal{H}$.

The canonical example of a simple Hilbert space is a spin-1/2 system, for example the two spin states of an electron. Classically, the spin $\vec{S}$ of an electron can point in any direction. In quantum mechanics, this is encoded in the complex two-dimensional Hilbert space $\mathcal{H} = \mathbb{C}^2$. Usually, we use the basis states

$$\text{"up"}\colon\ |\uparrow\rangle = \begin{pmatrix}1\\0\end{pmatrix} \qquad \text{and} \qquad \text{"down"}\colon\ |\downarrow\rangle = \begin{pmatrix}0\\1\end{pmatrix}.$$

This is reminiscent of the Ising model, though here, the spin state can be in any superposition

$$|\psi\rangle = \alpha\,|\uparrow\rangle + \beta\,|\downarrow\rangle \tag{2.3}$$

with $|\alpha|^2 + |\beta|^2 = 1$. In particular,

$$|\rightarrow\rangle = \frac{1}{\sqrt{2}}\big(\,|\uparrow\rangle + |\downarrow\rangle\,\big) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} \tag{2.4}$$

is oriented in the positive x direction.

2.1.2 Observables and Operators

Any physical observable is represented by a self-adjoint linear operator acting on $\mathcal{H}$. For a finite-dimensional $\mathcal{H}$, this means the observable can be represented by a hermitian matrix.

For the spin-1/2 example, the components of the angular momentum are represented by

$$\begin{aligned}\hat{S}^x &= \frac{\hbar}{2}\sigma_x = \frac{\hbar}{2}\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix},\\ \hat{S}^y &= \frac{\hbar}{2}\sigma_y = \frac{\hbar}{2}\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix},\\ \hat{S}^z &= \frac{\hbar}{2}\sigma_z = \frac{\hbar}{2}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix},\end{aligned} \tag{2.5}$$

where σ_x, σ_y, σ_z are the Pauli matrices. For a spin pointing along an arbitrary direction $\vec{e} = (e_x, e_y, e_z)$, we write

$$\vec{e}\cdot\hat{\vec{S}} = \frac{\hbar}{2}\begin{pmatrix}e_z & e_x - ie_y\\ e_x + ie_y & -e_z\end{pmatrix}. \tag{2.6}$$

Note that

$$\begin{aligned}[\hat{S}^x, \hat{S}^y] &\equiv \hat{S}^x\hat{S}^y - \hat{S}^y\hat{S}^x = i\hbar\hat{S}^z,\\ [\hat{S}^y, \hat{S}^z] &= i\hbar\hat{S}^x,\\ [\hat{S}^z, \hat{S}^x] &= i\hbar\hat{S}^y.\end{aligned} \tag{2.7}$$

The spin operators don't commute.

2.1.3 Schrödinger equation

The equation governing the dynamics of a quantum state is the Schrödinger equation

$$i\hbar\partial_t \left|\psi\left(t\right)\right\rangle = \hat{H}\left|\psi\left(t\right)\right\rangle, \tag{2.8}$$

where $\hat{H}$ is the Hamilton operator, or Hamiltonian, encoding the energy of the system. Equation (2.8) is a *first-order* linear differential equation.

For a stationary solution of the Schrödinger equation we can use the ansatz

$$\left|\psi\left(t\right)\right\rangle = e^{-iEt/\hbar}\left|\psi\left(0\right)\right\rangle, \tag{2.9}$$

where E is the energy of the system. The Schrödinger equation then simplifies to a linear eigenvalue problem

$$\hat{H}\left|\psi\right\rangle = E\left|\psi\right\rangle. \tag{2.10}$$

2.1.4 Mixed states and density matrix

Unless perfectly isolated, a system is not in a pure state $|\psi\rangle$, but rather in an incoherent mixture of several states. Such a general state is described by a density matrix $\hat{\rho}$, a positive semidefinite *operator* with

$$\mathrm{Tr}(\hat{\rho}) = 1. \tag{2.11}$$

For a pure state $|\psi\rangle$, the density matrix is given by

$$\hat{\rho}_{\mathrm{pure}} = \left|\psi\right\rangle\left\langle\psi\right|. \tag{2.12}$$

For the example of a spin pointing in the x direction, we find

$$\hat{\rho}_{\rightarrow} = |\rightarrow\rangle\langle\rightarrow| = \frac{1}{2}\begin{pmatrix}1\\1\end{pmatrix}\begin{pmatrix}1 & 1\end{pmatrix} = \begin{pmatrix}1/2 & 1/2\\1/2 & 1/2\end{pmatrix}. \tag{2.13}$$

More generally, we can write a density matrix as

$$\hat{\rho} = \sum_{i,j} p_{ij} |\psi_i\rangle\langle\psi_j| . \tag{2.14}$$

For example, for a system in a probabilistic mixture of the pure states $|\uparrow\rangle$ and $|\downarrow\rangle$, the density matrix reads

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

Finally, we define the purity of the system as $\mathrm{Tr}\left(\hat{\rho}^2\right)$ [Compare $\mathrm{Tr}\left(\hat{\rho}^2_{\text{pure}}\right)$ and $\mathrm{Tr}\left(\hat{\rho}^2_{\text{mixed}}\right)$].

The unitary time evolution of the density matrix follows from Eqs. (2.14) and (2.8) as

$$i\hbar\partial_t\hat{\rho}(t) = \left[\hat{H}, \hat{\rho}(t)\right]. \tag{2.16}$$

2.1.5 Thermal density matrix

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

$$p_i = \frac{1}{Z} e^{-\beta E_i} \tag{2.17}$$

with $Z = \sum_i e^{-\beta E_i}$ the partition function. We have already seen how to describe statistical mixtures through the density matrix: In a quantum system, knowing the basis of eigenstates $|i\rangle$ with energy E_i,

$$\hat{H}|i\rangle = E_i|i\rangle , \tag{2.18}$$

the thermal density matrix can be written analogously as

$$\hat{\rho}_\beta = \frac{1}{Z}\sum_i e^{-\beta E_i} |i\rangle\langle i| . \tag{2.19}$$

Unfortunately, we usually don't know $|i\rangle$, as this requires full diagonalization of the Hamiltonian. In general, the density matrix can be written as

$$\hat{\rho}_\beta = \frac{1}{Z} e^{-\beta\hat{H}} \tag{2.20}$$

with

$$Z = \mathrm{Tr}\left(e^{-\beta\hat{H}}\right). \tag{2.21}$$

2.1.6 Measurement process

The outcome of a measurement in a quantum system is (usually) intrusive and not deterministic. After measuring an observable A, the wave function of the system collapses to an eigenvector of $\hat{A}$ and the outcome of the measurement is given by the respective eigenvalue. The state of the system is thus changed by the measurement process.

Typically, we are not interested in a single measurement, but in the average value of an operator $\hat{A}$. This expectation value can be calculated in the general case as

$$\langle \hat{A} \rangle = \mathrm{Tr}\Big(\hat{\rho}\hat{A}\Big). \tag{2.22}$$

For a pure state $\hat{\rho} = |\psi\rangle \langle\psi|$, Eq. (2.22) reduces to

$$\langle \hat{A} \rangle = \langle\psi|\hat{A}|\psi\rangle . \tag{2.23}$$

For a system at temperature T, we find from Eqs. (2.20) and (2.21) the thermal expectation value of an observable A

$$\langle \hat{A} \rangle_\beta = \mathrm{Tr}\Big(\hat{A}\hat{\rho}_\beta\Big) = \frac{\mathrm{Tr}\Big(\hat{A}e^{-\beta\hat{H}}\Big)}{\mathrm{Tr}\Big(e^{-\beta\hat{H}}\Big)}. \tag{2.24}$$

2.1.7 Simultaneous measurement of non-commuting operators

We have mentioned that a measurement projects a state onto an eigenvector of the operator $\hat{A}$. This projection renders commuting operators, $[\hat{A}, \hat{B}] = 0$, special, as they have the same eigenspaces and thus, can be measured successively without a further collapse of the wave function.

If, on the other hand, the operators do not commute, it is not possible to measure both observables exactly. In particular, if

$$[\hat{A}, \hat{B}] = i\hbar, \tag{2.25}$$

then the product of the root-mean-square deviations ΔA and ΔB of observables A and B fulfill

$$\Delta A \cdot \Delta B \geq \frac{\hbar}{2}. \tag{2.26}$$

The relation (2.26) is called the Heisenberg uncertainty relation. Operators that obey Eq. (2.25) are called canonically conjugate. For general operators, the right-hand side of Eq. (2.25) is still an operator, not a complex number, see for example Eqs. (2.7).

2.2 Examples

2.2.1 Single quantum particle in an external potential

The Hilbert space of wave functions in n spatial dimensions is $L_2(\mathbb{R}^n)$, the space of all twice differentiable, square-integrable, complex functions over $\mathbb{R}^n$, and is thus

infinite dimensional. Working in the coordinate representation, the position operator is given by

$$\hat{\vec{r}} = \vec{r}, \tag{2.27}$$

the scalar product is given by

$$\langle f|g\rangle = \int \mathrm{d}^n r \, f^*\left(\vec{r}\right) g\left(\vec{r}\right), \tag{2.28}$$

and

$$\langle f|\hat{\vec{r}}|g\rangle = \int \mathrm{d}^n r \, f^*\left(\vec{r}\right) \vec{r} g\left(\vec{r}\right). \tag{2.29}$$

The momentum operator in the real-space representation becomes

$$\hat{\vec{p}} = -i\hbar\boldsymbol{\nabla}. \tag{2.30}$$

Notice that

$$[\hat{r}_i, \hat{p}_i] = i\hbar \neq 0. \tag{2.31}$$

The Hamiltonian of the system is

$$H\left(\hat{\vec{r}}, \hat{\vec{p}}\right) = \frac{(\hat{\vec{p}})^2}{2m} + V(\hat{\vec{r}}) = -\frac{\hbar^2}{2m}\boldsymbol{\nabla}^2 + V\left(\vec{r}\right), \tag{2.32}$$

where the first and the second terms are the kinetic and potential energy, respectively. We then find the Schrödinger equation

$$i\hbar\frac{\partial\psi\left(\vec{r}\right)}{\partial t} = -\frac{\hbar^2}{2m}\boldsymbol{\nabla}^2\psi\left(\vec{r}\right) + V\left(\vec{r}\right)\psi\left(\vec{r}\right). \tag{2.33}$$

2.2.2 Harmonic oscillator

A second canonical example is the (one-dimensional) harmonic oscillator with a potential $V(\hat{q}) = \frac{k}{2}\hat{q}^2$, where, as is conventional, we use the real-space coordinate q. Using units with $m = \hbar = k = 1$, the time-independent Schrödinger equation reads

$$H\left|n\right\rangle = \frac{1}{2}\left(\hat{p}^2 + \hat{q}^2\right)\left|n\right\rangle = E_n\left|n\right\rangle. \tag{2.34}$$

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

$$-\frac{1}{2}\phi_n''\left(q\right) + \frac{q^2}{2}\phi_n\left(q\right) = E_n\phi_n\left(q\right) \tag{2.35}$$

with eigenfunctions

$$\phi_n\left(q\right) = \frac{1}{\sqrt{2^n n!\sqrt{\pi}}}e^{-q^2/2}H_n\left(q\right), \tag{2.36}$$

where $H_n\left(q\right)$ are the Hermite polynomials, and $n \in \mathbb{N}_0$. Finally, the eigenvalues are

$$E_n = n + 1/2. \tag{2.37}$$

There is a more elegant approach for solving this problem, motivated by the observation that $\langle m|\hat{q}|n\rangle$ and $\langle m|\hat{p}|n\rangle$ only have non-zero values for $m = n \pm 1$. In particular, we can introduce the so-called ladder operators

$$\begin{aligned} \hat{a} &= \frac{1}{\sqrt{2}}\left(\hat{q} + i\hat{p}\right), \\ \hat{a}^\dagger &= \frac{1}{\sqrt{2}}\left(\hat{q} - i\hat{p}\right) \end{aligned} \tag{2.38}$$

with commutation relations

$$\begin{aligned} [\hat{a}, \hat{a}] &= [\hat{a}^\dagger, \hat{a}^\dagger] = 0, \\ [\hat{a}, \hat{a}^\dagger] &= 1. \end{aligned} \tag{2.39}$$

The raising operator $\hat{a}^\dagger$ and the lowering operator $\hat{a}$ act as

$$\begin{aligned} \hat{a}^\dagger |n\rangle &= \sqrt{n+1}\,|n+1\rangle, \\ \hat{a}\,|n\rangle &= \sqrt{n}\,|n-1\rangle. \end{aligned} \tag{2.40}$$

Rewriting the momentum and position operators in terms of the ladder operators,

$$\begin{aligned} \hat{q} &= \frac{1}{\sqrt{2}}\left(\hat{a} + \hat{a}^\dagger\right), \\ \hat{p} &= \frac{1}{i\sqrt{2}}\left(\hat{a} - \hat{a}^\dagger\right), \end{aligned} \tag{2.41}$$

we arrive at the Hamiltonian

$$\hat{H} = \hat{a}^\dagger\hat{a} + 1/2 \tag{2.42}$$

with spectrum

$$E_n = n + 1/2, \tag{2.43}$$

as we already obtained above.

2.2.3 Spin-S problem

In section 2.1.1, we have encountered the Hilbert space and spin operators for the most common case of a spin 1/2. The algebra of the spin operators, given by the commutation relations in Eqs. (2.7), allows not only for a two-dimensional representation, but for a series of $(2S+1)$-dimensional representations in a Hilbert space $\mathbb{C}^{2S+1}$ for all integer and half-integer values $S = 0, 1/2, 1, 3/2, \ldots$. We define the basis states $\{|s\rangle\}$ with respect to the $\hat{S}^z$ operator with the quantum number s taking values in the range $-S, -S+1, \ldots S-1, S$. These basis states are connected through

$$\begin{aligned} \hat{S}^+ |s\rangle &= \sqrt{S(S+1) - s(s+1)}\,|s+1\rangle, \\ \hat{S}^- |s\rangle &= \sqrt{S(S+1) - s(s-1)}\,|s-1\rangle, \end{aligned} \tag{2.44}$$

and

$$\begin{aligned} \hat{S}^x &= \frac{1}{2}\left(\hat{S}^+ + \hat{S}^-\right), \\ \hat{S}^y &= \frac{1}{2i}\left(\hat{S}^+ - \hat{S}^-\right). \end{aligned} \tag{2.45}$$

Finally, the dynamics of a spin coupled to a magnetic field $\vec{h}$ is given through the Hamiltonian

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

3 Quantum one-body problem

3.1 Time-independent 1D Schrödinger equation

We start the numerical solution of quantum problems with the time-independent one-dimensional (1D) Schrödinger equation for a particle with mass m in a potential $V(x)$. In one dimension, the Schrödinger equation is just an ordinary differential equation

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

As we already mentioned in the previous chapter, $\psi\left(x\right)$ should be twice differentiable and square-integrable, such that the wave function is normalizable,

$$\int \mathrm{d}x\left|\psi\left(x\right)\right|^2=1. \tag{3.2}$$

To approach this problem, we start with a simple finite-difference scheme and discretize space into intervals of length Δx, denoting the space points by

$$x_n=n\Delta x, \qquad n\in\mathbb{Z}, \tag{3.3}$$

and the wave function at these points by

$$\psi_n=\psi\left(x_n\right). \tag{3.4}$$

3.1.1 Numerov algorithm

Efficient algorithms exist for the solution of first-order ordinary differential equations, or coupled sets thereof, such as the Runge-Kutta method. One strategy to solve the second-order Schrödinger equation could thus be to rewrite it as a coupled system of two first order differential equations, and then apply such a solver. However, there exist better methods, one of which we discuss here. For the special form

$$\psi''\left(x\right)+k\left(x\right)\psi\left(x\right)=0 \tag{3.5}$$

of the Schrödinger equation, with $k\left(x\right)=2m[E-V\left(x\right)]/\hbar^2$ [and $k_n=k(x_n)$], we can derive the so-called Numerov algorithm. It starts from the Taylor expansion of ψ_n,

$$\begin{aligned}\psi_{n\pm1}=\psi_n\pm\Delta x\,\psi_n'+\frac{(\Delta x)^2}{2}\psi_n''\pm\frac{(\Delta x)^3}{6}\psi_n^{(3)}+\frac{(\Delta x)^4}{24}\psi_n^{(4)}\\ \pm\frac{(\Delta x)^5}{120}\psi_n^{(5)}+O[(\Delta x)^6],\end{aligned} \tag{3.6}$$

where $\psi_n^{(m)}$ is the m-th derivative of $\psi(x)$ evaluated at x_n. Adding ψ_{n-1} and ψ_{n+1}, we obtain

$$\psi_{n-1}+\psi_{n+1}=2\psi_n+\left(\Delta x\right)^2\psi_n''+\frac{(\Delta x)^4}{12}\psi_n^{(4)}. \tag{3.7}$$

Replacing the fourth derivative by a finite-difference 'second derivative' of the second derivative

$$\psi_n^{(4)}=\frac{\psi_{n+1}''+\psi_{n-1}''-2\psi_n''}{\left(\Delta x\right)^2} \tag{3.8}$$

and substituting $-k_n\psi_n = \psi_n''$, we obtain the Numerov algorithm:

$$\left(1+\frac{(\Delta x)^2}{12}k_{n+1}\right)\psi_{n+1} = 2\left(1-\frac{5(\Delta x)^2}{12}k_n\right)\psi_n - \left(1+\frac{(\Delta x)^2}{12}k_{n-1}\right)\psi_{n-1} + O[(\Delta x)^6], \tag{3.9}$$

which is locally of sixth order in Δx.

To run the Numerov algorithm, we need the wave function not just at one point as would be the case for a first-order differential equation, but at *two* initial values. We will now present several ways to obtain these.

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

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

For general potentials, we need to use other approaches. If the potential vanishes identically for large distances, in other words $V(x) = 0$ for $|x| \geq a$, we can use the exact solution of the Schrödinger equation (for $E < 0$) at large distances to define starting points, namely

$$\begin{aligned}&\psi(-a) = 1,\\ &\psi(-a-\Delta x) = \exp\left(-\Delta x\sqrt{-2mE}/\hbar\right).\end{aligned} \tag{3.10}$$

In the case where the potential never vanishes, we need to begin with a single starting value $\psi(x_0)$ and obtain the second starting value $\psi(x_1)$ by performing an integration over the first space step Δx with an Euler or Runge-Kutta algorithm.

Finally, at the end of the calculation, the wave function needs to be normalized.

3.1.2 1D scattering problem

The scattering problem is characterized by a potential that vanishes at large distances, $V(x) \to 0$ for $|x| \to \infty$. Its most appealing property is that solutions are guaranteed to exist for all energies $E > 0$. The solution becomes particularly simple if the potential is nonzero only on a finite interval $x \in [0, a]$, since the asymptotic solution outside of this region is a plane wave. Specifically, for a particle approaching the potential barrier from the left we can make the following ansatz for the free propagation when $x < 0$:

$$\psi_{\mathrm{L}}(x) = Ae^{iqx} + Be^{-iqx}, \tag{3.11}$$

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

$$\psi_{\mathrm{R}}(x) = Ce^{iqx}. \tag{3.12}$$

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

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

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

$$R = \frac{|B|^2}{|A|^2}, \tag{3.13}$$

$$T = \frac{|C|^2}{|A|^2} = \frac{1}{|A|^2}. \tag{3.14}$$

3.1.3 Bound states and solution of the eigenvalue problem

While there exist scattering states for all energies $E > 0$, bound-state solutions of the Schrödinger equation with $E < 0$ exist only for discrete energy eigenvalues (if at all). Integrating the Schrödinger equation from $-\infty$ to $+\infty$ for a negative E, we will obtain a formal solution which diverges to $\pm\infty$ as $x \to \infty$ for almost all values. These functions cannot be normalized and thus do not constitute a physical bound-state solution to the Schrödinger equation. Only for discrete eigenvalues E will the solution go to zero as $x \to \pm\infty$.

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

- Start with an initial guess E.
- Integrate the Schrödinger equation for $\psi_E(x)$ from $x = 0$ to $x_\mathrm{f} \gg a$ and determine the value $\psi_E(x_\mathrm{f})$
- Use a root solver, such as a bisection method, to look for an energy E with $\psi_E(x_\mathrm{f}) \approx 0$.

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

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

- Pick a starting point b and choose as energy $E = V(b)$. Note that then $\psi''_E(b) = 0$.
- Integrate both from a to b and from 0 to b. Obtain $\psi_\mathrm{L}(b)$ and $\psi_\mathrm{R}(b)$. Obtain numerical estimations for the first derivatives: $\psi'_\mathrm{L}(b) = [\psi_\mathrm{L}(b) - \psi_\mathrm{L}(b - \Delta x)]/\Delta x$, $\psi'_\mathrm{R}(b) = [\psi_\mathrm{R}(b + \Delta x) - \psi_\mathrm{R}(b)]/\Delta x$.

- Match the two solutions and their first derivatives at point b (matching of the second derivative we get for free, since it is zero). Keeping in mind that we can multiply non-normalized wave functions by an arbitrary factor, we obtain the conditions

$$\begin{aligned} \psi_{\mathrm{L}}(b) &= \alpha \psi_{\mathrm{R}}(b), \\ \psi'_{\mathrm{L}}(b) &= \alpha \psi'_{\mathrm{R}}(b), \end{aligned} \tag{3.15}$$

  which reduces to

$$\frac{\psi'_{\mathrm{L}}(b)}{\psi_{\mathrm{L}}(b)} = \frac{\psi'_{\mathrm{R}}(b)}{\psi_{\mathrm{R}}(b)}. \tag{3.16}$$

- Solve the last equation using any root-solving method, such as a bisection algorithm, to find b for which $E = V(b)$ is indeed an eigenvalue.

3.2 Time-independent Schrödinger equation in higher dimensions

The time independent Schrödinger equation in more than one dimension is a partial differential equation and cannot, in general, be solved by a simple ordinary differential equation solver such as the Numerov algorithm. Before employing a partial differential equation solver, which is substantially more difficult to handle, we should, however, first try to use symmetries to reduce the problem to decoupled one-dimensional problems. This can be done if the equation factorizes.

3.2.1 Factorization along coordinate axes

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

$$\psi(\vec{r}) = \psi_x(x)\,\psi_y(y)\,\psi_z(z), \tag{3.17}$$

we can factorize the partial differential equation into three independent ordinary differential equations, which can be solved as above.

3.2.2 Potential with spherical symmetry

Another factorization trick applies to spherically symmetric potentials with $V(\vec{r}) = V(|\vec{r}|)$, where an ansatz using spherical harmonics

$$\psi_{l,m}(\vec{r}) = \psi_{l,m}(r, \theta, \phi) = \frac{u(r)}{r} Y_{lm}(\theta, \phi), \quad l \in \mathbb{N}_0, \quad m \in \mathbb{Z}, \quad |m| \leq l, \tag{3.18}$$

can be used to reduce the three-dimensional Schrödinger equation to a one-dimensional one for the radial wave function $u(r)$. In particular, we find

$$\left(-\frac{\hbar^2}{2\mu} \frac{\mathrm{d}^2}{\mathrm{d}r^2} + \frac{\hbar^2 l(l+1)}{2\mu r^2} + V(r) \right) u(r) = E u(r), \tag{3.19}$$

where we have called the particle mass μ (to avoid confusion with magnetic quantum number m in the spherical harmonics) and $r = |\vec{r}|$. This is again a one-dimensional Schrödinger equation with a modified effective potential

$$V_l(r) = V(r) + \frac{\hbar^2 l(l+1)}{2\mu r^2} \tag{3.20}$$

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

3.2.3 Finite difference methods

If we cannot use any symmetries or factorizations, we can still employ solvers for partial differential equations. One approach is to discretize the Schrödinger equation on a spatial mesh using a finite difference approximation. Replacing differentials by differences, we convert the Schrödinger equation into a system of coupled linear equations. Starting from the three-dimensional Schrödinger equation (we set $\hbar = 1$ from now on)

$$\boldsymbol{\nabla}^2 \psi(\vec{r}) + 2m\left[E - V(\vec{r})\right]\psi(\vec{r}) = 0, \tag{3.21}$$

we discretize space and obtain the system of linear equations

$$\begin{aligned} 0 = \frac{1}{(\Delta x)^2}\big[&\psi(x_{n+1}, y_n, z_n) + \psi(x_{n-1}, y_n, z_n) \\ &+\psi(x_n, y_{n+1}, z_n) + \psi(x_n, y_{n-1}, z_n) \\ &+\psi(x_n, y_n, z_{n+1}) + \psi(x_n, y_n, z_{n-1})\big] \\ &+\left\{2m\left[E - V(\vec{r})\right] - \frac{6}{(\Delta x)^2}\right\}\psi(x_n, y_n, z_n). \end{aligned} \tag{3.22}$$

Both the scattering problem and the calculation of bound states essentially reduce to eigenvalue problems. For small problems, Wolfram Mathematica or the `dsysv` function of the LAPACK library can be used. For larger problems, it is essential to realize that the matrices produced by the discretization of the Schrödinger equation are usually very sparse, meaning that only $O(N)$ of the N^2 matrix elements are nonzero. For these sparse systems of equations, optimized iterative numerical algorithms exist and are implemented in numerical libraries such as in the EIGEN library (C++) or in the SciPy module (Python). For big systems, sparse solvers such as the Lanczos algorithm (which will be discussed in detail in the following lectures) are the best and, again, there exist many efficient implementations of iterative algorithms for sparse matrices.

3.2.4 Variational approaches

There is a solution to the problem in the case of general potentials (or for more than two particles), if we are interested in the ground state and it is not possible to reduce the Schrödinger equation to a one-dimensional problem. It is based on expanding a variational wave functions in terms of a finite set of basis functions

$$|\phi\rangle = \sum_{i=1}^{N} a_i |u_i\rangle. \tag{3.23}$$

In order to estimate the ground-state energy, we want to minimize the energy E^* of the variational wave function

$$E^* = \frac{\langle\phi|\hat{H}|\phi\rangle}{\langle\phi|\phi\rangle}. \tag{3.24}$$

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

To perform the minimization, we denote by

$$H_{ij} = \langle u_i|\hat{H}|u_j\rangle = \int \mathrm{d}\vec{r}\, u_i^*(\vec{r}) \left(-\frac{\hbar^2}{2m}\boldsymbol{\nabla}^2 + V\right) u_j(\vec{r}) \tag{3.25}$$

the matrix elements of the Hamiltonian H and by

$$S_{ij} = \langle u_i|u_j\rangle = \int \mathrm{d}\vec{r}\, u_i^*(\vec{r})\, u_j(\vec{r}) \tag{3.26}$$

the overlap matrix. Note that for an orthogonal basis set, S_{ij} is the identity matrix δ_{ij}. Minimizing Eq. (3.24) with respect to a_i^*, we obtain a generalized eigenvalue problem

$$\sum_j H_{ij}a_j = E^* \sum_k S_{ik}a_k, \tag{3.27}$$

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

$$H\vec{a} = E^* S\vec{a}, \tag{3.28}$$

where H and S are matrices consisting of H_{ij} and S_{ij} respectively. If the basis set is orthogonal, this reduces to an ordinary eigenvalue problem.

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

$$\begin{aligned} H\vec{a} &= E^* S\vec{a}, \\ HU\vec{b} &= E^* SU\vec{b}, \\ U^T HU\vec{b} &= E^* U^T SU\vec{b} = E^*\vec{b}, \end{aligned} \tag{3.29}$$

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

Example: the anharmonic oscillator

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

$$\begin{aligned} \hat{H} &= \hat{H}_0 + \lambda\hat{q}^4, \\ \hat{H}_0 &= \frac{1}{2}\left(\hat{p}^2 + \hat{q}^2\right), \end{aligned} \tag{3.30}$$

where the harmonic oscillator H_0 was already discussed in Sec. 2.2.2.We use the N lowest harmonic-oscillator eigenvectors $|n\rangle$ as basis states of a finite basis and write the Hamiltonian as

$$\hat{H} = \frac{1}{2} + \hat{a}^\dagger \hat{a} + \lambda \hat{q}^4 = \frac{1}{2} + \hat{a}^\dagger \hat{a} + \frac{\lambda}{4} \left(\hat{a}^\dagger + \hat{a}\right)^4 . \tag{3.31}$$

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

3.2.5 Finite element method

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

3.3 Time-dependent Schrödinger equation

We are now switching to problems where we want to follow the time evolution of a given state $|\psi_0\rangle = |\psi(t_0)\rangle$, and thus study non-stationary quantum systems.

3.3.1 Spectral methods

By introducing a basis and obtaining 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 $|\psi_0\rangle$ from time t_0 to t, we first solve the stationary eigenvalue problem $\hat{H}|\phi\rangle = E|\phi\rangle$ and calculate the eigenvectors $|\phi_n\rangle$ with eigenvalues ε_n. Next, we represent the initial wave function $|\psi_0\rangle$ as a spectral decomposition

$$|\psi_0\rangle = \sum_n c_n |\phi_n\rangle . \tag{3.32}$$

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

$$|\psi(t)\rangle = \sum_n c_n e^{-i\varepsilon_n (t-t_0)/\hbar} |\phi_n\rangle . \tag{3.33}$$

This approach is, however, only useful for very small problems due to the effort of diagonalizing the Hamiltonian. A more broadly applicable method is direct numerical integration, discussed in the next two subsections.

3.3.2 Direct numerical integration

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

A forward Euler scheme

$$|\psi(t_{n+1})\rangle = |\psi(t_n)\rangle - \frac{i\Delta t}{\hbar}\hat{H}\,|\psi(t_n)\rangle \tag{3.34}$$

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

$$\psi(\vec{r}, t+\Delta t) = e^{-iH\Delta t/\hbar}\psi(\vec{r}, t) \tag{3.35}$$

is unitary and thus conserves the norm, we want to look for a unitary approximant as integrator. (Here and in the remainder of this chapter, we use the symbol H to denote the representation of the Hamiltonian in the chosen finite basis set.) Instead of using the forward Euler method (3.34), which is a first-order Taylor expansion of the exact time evolution

$$e^{-iH\Delta t/\hbar} = \mathbb{1} - \frac{i\Delta t}{\hbar}H + O[(\Delta t)^2], \tag{3.36}$$

we reformulate the time evolution operator as

$$e^{-iH\Delta t/\hbar} = \left(e^{iH\Delta t/2\hbar}\right)^{-1} e^{-iH\Delta t/2\hbar} = \left(\mathbb{1} + \frac{i\Delta t}{2\hbar}H\right)^{-1}\left(\mathbb{1} - \frac{i\Delta t}{2\hbar}H\right) + O[(\Delta t)^3], \tag{3.37}$$

which is unitary. This gives the simplest stable and unitary integrator algorithm

$$\psi(\vec{r}, t+\Delta t) = \left(\mathbb{1} + \frac{i\Delta t}{2\hbar}H\right)^{-1}\left(\mathbb{1} - \frac{i\Delta t}{2\hbar}H\right)\psi(\vec{r}, t)\,, \tag{3.38}$$

or equivalently

$$\left(\mathbb{1} + \frac{i\Delta t}{2\hbar}H\right)\psi(\vec{r}, t+\Delta t) = \left(\mathbb{1} - \frac{i\Delta t}{2\hbar}H\right)\psi(\vec{r}, t)\,. \tag{3.39}$$

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

3.3.3 Split-operator method

A simpler and explicit method is possible for a quantum particle in the real space picture with the Schrödinger equation in continuous space. Writing the Hamiltonian as

$$\hat{H} = \hat{T} + \hat{V} \tag{3.40}$$

with

$$\hat{T} = \frac{(\hat{\vec{p}})^2}{2m}, \qquad \hat{V} = V\left(\hat{\vec{r}}\right), \tag{3.41}$$

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 particle in d-dimensional space, its wave function in momentum space is obtained through the Fourier transform

$$\tilde{\psi}(\vec{k}) = \left(\frac{1}{\sqrt{2\pi}}\right)^d \int_{-\infty}^{+\infty} \psi\left(\vec{r}\right) e^{-i\vec{k}\cdot\vec{r}}\, \mathrm{d}\vec{r}, \tag{3.42}$$

and the inverse Fourier transform yields

$$\psi\left(\vec{r}\right) = \left(\frac{1}{\sqrt{2\pi}}\right)^d \int_{-\infty}^{+\infty} \tilde{\psi}(\vec{k}) e^{i\vec{k}\cdot\vec{r}}\, \mathrm{d}\vec{k}\,. \tag{3.43}$$

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

If we split the time evolution as

$$e^{-i\Delta t\hat{H}/\hbar} = e^{-i\Delta t\hat{V}/2\hbar} e^{-i\Delta t\hat{T}/\hbar} e^{-i\Delta t\hat{V}/2\hbar} + O[(\Delta t)^3], \tag{3.44}$$

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

$$\langle\vec{r}|\, e^{-i\Delta t\hat{V}/2\hbar}\, |\psi\rangle = e^{-i\Delta tV(\vec{r})/2\hbar}\psi\left(\vec{r}\right), \tag{3.45}$$

$$\langle\vec{k}|\, e^{-i\Delta t\hat{T}/\hbar}\, |\psi\rangle = e^{-i\Delta t\hbar\left|\vec{k}\right|^2/2m}\tilde{\psi}(\vec{k}) \tag{3.46}$$

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

$$\begin{aligned} e^{-it\hat{H}/\hbar} &= \left(e^{-i\Delta t\hat{V}/2\hbar} e^{-i\Delta t\hat{T}/\hbar} e^{-i\Delta t\hat{V}/2\hbar}\right)^N \\ &= e^{-i\Delta t\hat{V}/2\hbar}\left[e^{-i\Delta t\hat{T}/\hbar} e^{-i\Delta t\hat{V}/\hbar}\right]^{N-1} e^{-i\Delta t\hat{T}/\hbar} e^{-i\Delta t\hat{V}/2\hbar}. \end{aligned} \tag{3.47}$$

In practice, in order to obtain efficient representations of the wave functions both in real and momentum space, we still need to discretize 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 $k_n = \frac{2\pi}{P}n$ for each spatial direction, with $n = 0, 1, ..., P-1$. 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 $O\left(P\log\left(P\right)\right)$ operations.

The discretized algorithm then starts as

$$\begin{aligned}\psi_1\left(\vec{r}\right) &= e^{-i\Delta t V(\vec{r})/2\hbar}\psi_0\left(\vec{r}\right),\\ \tilde{\psi}_1(\vec{k}) &= \mathcal{F}\psi_1\left(\vec{r}\right),\end{aligned} \tag{3.48}$$

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

$$\begin{aligned}\tilde{\psi}_{2n}(\vec{k}) &= e^{-i\Delta t\hbar\left|\vec{k}\right|^2/2m}\tilde{\psi}_{2n-1}(\vec{k}),\\ \psi_{2n}\left(\vec{r}\right) &= \mathcal{F}^{-1}\tilde{\psi}_{2n}(\vec{k}),\end{aligned} \tag{3.49}$$

$$\begin{aligned}\psi_{2n+1}\left(\vec{r}\right) &= e^{-i\Delta t V(\vec{r})/\hbar}\psi_{2n}\left(\vec{r}\right),\\ \tilde{\psi}_{2n+1}(\vec{k}) &= \mathcal{F}\psi_{2n+1}\left(\vec{r}\right).\end{aligned} \tag{3.50}$$

Only in the last step, we finish with another half time step in real space,

$$\psi_{2N+1}\left(\vec{r}\right) = e^{-i\Delta t V(\vec{r})/2\hbar}\psi_{2N}\left(\vec{r}\right). \tag{3.51}$$

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

4 Introduction to the quantum many-body problem

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

$$\mathcal{H}^{(N)} = \mathcal{H}^{\otimes N} \equiv \underbrace{\mathcal{H} \otimes \mathcal{H} \otimes \cdots \otimes \mathcal{H}}_{\text{N times}} \equiv \bigotimes_{i=1}^{N} \mathcal{H} \tag{4.1}$$

with dimension d^N.

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

This exponential scaling of the Hilbert space dimension with the number of particles is a big (numerical) challenge. Even in the simplest case of a spin-1/2 particle with $d = 2$, the basis for $N = 30$ spins is already of size $d = 2^{30} \approx 10^9$. A single complex vector requires 16GB of memory and will hardly fit into the memory of your PC. This challenge will be addressed during this course by learning about

1. Brute-force (exact) methods solving a problem in a huge Hilbert space for modest numbers of particles and effective (spin) models.
2. Quantum state compression to variational wave functions in the relevant part of the Hilbert space.
3. Quantum Monte Carlo methods for bosonic and magnetic systems.
4. Approximative methods, reducing the many-particle problem to a single-particle problem.

4.1 Indistinguishable particles

4.1.1 Bosons and fermions

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

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

Upon exchanging the two particles, the wave function needs to be identical up to a phase factor $e^{i\phi}$, so that all observables are invariant. Exchanging the arguments twice brings back the original wave function and can be thought of as an actual physical process in which one particle moves around the other. This path can always be contracted to a point in three dimensions (the first homotopy group is trivial in 3D [2]) and thus

$$\psi\left(\vec{r}_1, \vec{r}_2\right) = e^{i\phi}\psi\left(\vec{r}_2, \vec{r}_1\right) = e^{2i\phi}\psi\left(\vec{r}_1, \vec{r}_2\right). \tag{4.3}$$

As a consequence, $e^{i\phi} = \pm 1$ and

$$\psi\left(\vec{r}_1, \vec{r}_2\right) = \pm\psi\left(\vec{r}_2, \vec{r}_1\right). \tag{4.4}$$

The many-body Hilbert space can be split into two orthogonal subspaces: one in which particles pick up a '$-$' sign upon exchange of any two particles and are called *fermions*, and the other where particles pick up a '$+$' sign and are called *bosons*.

Bosons

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

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

where the sum runs over all permutations p of N particles, and $\mathcal{N}_\mathrm{S}$ is a normalization factor.

Fermions

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

$$\Psi^{(\mathrm{A})} = \mathcal{S}_-\psi\left(\vec{r}_1, \ldots \vec{r}_N\right) \equiv \mathcal{N}_\mathrm{A} \sum_p \mathrm{sign}\left(p\right)\psi\left(\vec{r}_{p(1)}, \ldots, \vec{r}_{p(N)}\right), \tag{4.6}$$

where $\mathrm{sign}\left(p\right) = \pm 1$ is the sign of the permutation and $\mathcal{N}_\mathrm{A}$ again is a normalization factor.

A 'consequence' of the antisymmetrization is the *Pauli exclusion principle*, which states that no two fermions can occupy the same state $\phi(\vec{r})$,

$$\psi\left(\vec{r}_1, \vec{r}_2\right) = \phi\left(\vec{r}_1\right)\phi\left(\vec{r}_2\right), \tag{4.7}$$

since this vanishes under antisymmetrization,

$$\Psi\left(\vec{r}_1, \vec{r}_2\right) \propto \psi\left(\vec{r}_1, \vec{r}_2\right) - \psi\left(\vec{r}_2, \vec{r}_1\right) = \phi\left(\vec{r}_1\right)\phi\left(\vec{r}_2\right) - \phi\left(\vec{r}_2\right)\phi\left(\vec{r}_1\right) = 0. \tag{4.8}$$

Note that here and in the following, we have dropped the superscript A (or S), since it should always be clear what particles we are talking about.

[2] As a side remark we want to mention that in two dimensions, the first homotopy group is $\mathbb{Z}$ and not trivial: it matters whether we move the particles clockwise or anticlockwise when exchanging them, and two clockwise exchanges are not the identity anymore. Then, more generally, *anyonic* statistics are possible.

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 $r = (\vec{r}, \sigma)$ including both position $\vec{r}$ and spin σ has to be antisymmetric.

4.1.2 Fock space

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

$$\mathcal{F} = \bigoplus_{N=0}^{\infty} \mathcal{S}_{\pm} \mathcal{H}^{\otimes N}, \tag{4.9}$$

where $\mathcal{S}_+$ is the symmetrization operator used for bosons and $\mathcal{S}_-$ is the antisymmetrization operator used for fermions.

Occupation number basis

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

$$\frac{1}{\sqrt{2}} \left[\phi_1(r_1)\, \phi_2(r_2) \pm \phi_1(r_2)\, \phi_2(r_1)\right], \tag{4.10}$$

where again the + sign is for bosons and the − sign is for fermions.

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

Slater determinant

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

$$\Psi(r_1, \ldots, r_N) = \mathcal{S}_- \prod_{i_1}^{N} \phi_i(r_i) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \phi_1(r_1) & \ldots & \phi_N(r_1) \\ \vdots & & \vdots \\ \phi_1(r_N) & \ldots & \phi_N(r_N) \end{vmatrix}. \tag{4.11}$$

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

4.1.3 Creation and annihilation operators

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

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

$$\tilde{\Psi}(r_1, \ldots, r_{N-1}) = \mathcal{S}_{\pm} \int \mathrm{d}r_N \, \phi_i^*(r_N) \, \Psi(r_1, \ldots, r_N) \,. \tag{4.12}$$

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

$$\hat{a}_i |\phi_j\rangle = \delta_{ij} |0\rangle \,, \tag{4.13}$$

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

The creation operator $\hat{a}_i^\dagger$ is defined as the adjoint of the annihilation operator $\hat{a}_i$. Applying it to the vacuum "creates" a particle with the wave function

$$|\phi_i\rangle = \hat{a}_i^\dagger |0\rangle \,. \tag{4.14}$$

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

Commutation relations

The creation and annihilation operators satisfy certain canonical commutation relations, which we will discuss for an orthogonal set of basis functions.

For bosons, the commutation relations are the same as those of the ladder operators discussed for the harmonic oscillator, Eq. (2.39), namely

$$\begin{aligned} [\hat{a}_i, \hat{a}_j^\dagger] &= \hat{a}_i \hat{a}_j^\dagger - \hat{a}_j^\dagger \hat{a}_i = \delta_{ij}, \\ [\hat{a}_i, \hat{a}_j] &= [\hat{a}_i^\dagger, \hat{a}_j^\dagger] = 0. \end{aligned} \tag{4.15}$$

For fermions, on the other hand, the operators anticommute,

$$\begin{aligned} \{\hat{a}_i, \hat{a}_j^\dagger\} &= \hat{a}_i \hat{a}_j^\dagger + \hat{a}_j^\dagger \hat{a}_i = \delta_{ij}, \\ \{\hat{a}_i, \hat{a}_j\} &= \{\hat{a}_i^\dagger, \hat{a}_j^\dagger\} = 0. \end{aligned} \tag{4.16}$$

The anticommutation implies that

$$(\hat{a}_i^\dagger)^2 = \hat{a}_i^\dagger \hat{a}_i^\dagger = -\hat{a}_i^\dagger \hat{a}_i^\dagger, \tag{4.17}$$

and thus guarantees that

$$(\hat{a}_i^\dagger)^2 = 0, \tag{4.18}$$

as expected from the Pauli exclusion principle.

Fock basis in second quantization and normal ordering

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

$$|n_1, \ldots, n_N\rangle = \prod_{i=1}^{N} (\hat{a}_i^\dagger)^{n_i} |0\rangle = (\hat{a}_1^\dagger)^{n_1} (\hat{a}_2^\dagger)^{n_2} \ldots (\hat{a}_N^\dagger)^{n_N} |0\rangle . \tag{4.19}$$

For bosons, the ordering of the creation operators does not matter, since the operators commute. For fermions, however, the ordering matters since the fermionic creation operators anticommute and $\hat{a}_1^\dagger \hat{a}_2^\dagger |0\rangle = -\hat{a}_2^\dagger \hat{a}_1^\dagger |0\rangle$. We therefore need to agree on a specific ordering of the creation operators to define what we mean by the state $|n_1, \ldots, n_N\rangle$. The choice of ordering does not matter but we have to stay consistent and use the convention in Eq. (4.19) as one example.

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

$$\begin{aligned} \hat{a}_i |n_1, \ldots, n_i, \ldots, n_N\rangle &= \delta_{n_i,1} (-1)^{\sum_{j=1}^{i-1} n_i} |n_1, \ldots, n_i - 1, \ldots, n_N\rangle , \\ \hat{a}_i^\dagger |n_1, \ldots, n_i, \ldots, n_N\rangle &= \delta_{n_i,0} (-1)^{\sum_{j=1}^{i-1} n_i} |n_1, \ldots, n_i + 1, \ldots, n_N\rangle , \end{aligned} \tag{4.20}$$

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

4.2 Quantum spin models

In the study of many-body systems, we are often interested in complex phase diagrams or quantum dynamics. Of particular interest are quantum phase transitions, in other words phase transitions driven by quantum, not thermal, fluctuations. As only very few problems allow for an exact solution and the complexity of the Hilbert space severely restricts numerically tackling complicated models, we need to find model systems capturing a broad range of complex phenomena while being as simple as possible.

In (classical) statistical mechanics, the Ising model and other idealized (spin) models have allowed for insights into thermal phase transitions, their universality, and critical phenomena. In order for a model to exhibit quantum phenomena, an extensive number of operators of the model Hamiltonian must not commute with each other. Again, this can be realized in 'simple' spin models, since the spin operators defined in Sec. 2.1.2 indeed satisfy a non-trivial algebra. Quantum spin models are thus instrumental in investigating exotic quantum many-body states and quantum phase transitions. Finally, interacting fermions can in principle be mapped to spin models, though in practice, this is mainly restricted to one-dimensional systems.

4.2.1 Transverse field Ising model

The Drosophila of quantum spin models is the transverse field Ising model (TFIM), which exhibits a quantum phase transition between a spontaneously symmetry-broken and a disordered phase. It derives from the classical Ising model, a model

describing a lattice of spin-1/2 particles coupled only through their spins' z component, by adding a magnetic field in the x direction,

$$\hat{H} = \sum_{\langle ij \rangle} J_{ij} \hat{S}_i^z \hat{S}_j^z - \sum_i \frac{h_i}{2} \hat{S}_i^x. \tag{4.21}$$

Here, the symbol $\langle ij \rangle$ denotes a sum over all nearest-neighbor bonds in the lattice (without double-counting). Note that $\langle\langle ij \rangle\rangle$ would denote next-nearest neighbors. In this model, the spin operator for the x component does not commute with the z component, rendering this model inherently quantum.

For the translationally invariant one-dimensional case, in other words $J_{ij} = J$, $h_i = h$, this model is exactly solvable and exhibits a $T = 0$ phase diagram as shown in Fig. 3.

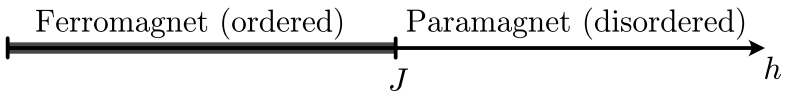


Figure 3: Zero temperature phase diagram of the Transverse field Ising model.

4.2.2 The Heisenberg model

The (quantum) Heisenberg model is the generalization of the classical Heisenberg model to non-commuting spin operators,

$$\hat{H} = \sum_{\langle ij \rangle} J_{ij} \hat{\vec{S}}_i \cdot \hat{\vec{S}}_j = \sum_{\langle ij \rangle} J_{ij} \left(\hat{S}_i^x \hat{S}_j^x + \hat{S}_i^y \hat{S}_j^y + \hat{S}_i^z \hat{S}_j^z \right). \tag{4.22}$$

We can introduce raising and lowering operators $\hat{S}^+$ and $\hat{S}^-$ as in Eq. (2.45) to find

$$\hat{H} = \sum_{\langle ij \rangle} J_{ij} \left[\frac{1}{2} \left(\hat{S}_i^+ \hat{S}_j^- + \hat{S}_i^- \hat{S}_j^+ \right) + \hat{S}_i^z \hat{S}_j^z \right]. \tag{4.23}$$

The three terms in Eq. (4.23) clearly do not change the *total* magnetization $\hat{M}^z = \sum_i \hat{S}_i^z$, in other words $\hat{M}^z$ is a symmetry of the Heisenberg Hamiltonian. Note that the first two terms can change the magnetization locally. What is more, the Heisenberg Hamiltonian possesses SU(2) symmetry, meaning it commutes with the operator $\exp\left(i\vec{\theta} \cdot \hat{\vec{M}}\right)$, where $\vec{\theta}$ is an arbitrary vector and $\hat{\vec{M}} = (\hat{M}^x, \hat{M}^y, \hat{M}^z) = \left(\sum_i \hat{S}_i^x,\ \sum_i \hat{S}_i^y,\ \sum_i \hat{S}_i^z \right)$ [3].

Using the basis $\{|\uparrow\uparrow\rangle, |\uparrow\downarrow\rangle, |\downarrow\uparrow\rangle, |\downarrow\downarrow\rangle\}$ for two spins, the matrix representation of the two-site operator is

$$\begin{pmatrix} J_{ij}/4 & 0 & 0 & 0 \\ 0 & -J_{ij}/4 & J_{ij}/2 & 0 \\ 0 & J_{ij}/2 & -J_{ij}/4 & 0 \\ 0 & 0 & 0 & J_{ij}/4 \end{pmatrix}. \tag{4.24}$$

The block-diagonal form reflects the conservation of the total magnetization.

[3] This is equivalent to the Hamiltonian commuting with $\hat{M}^x, \hat{M}^y, \hat{M}^z$.

4.2.3 The XXZ model

Typically, magnetic materials do not have the full SU(2) symmetry, but possess an easy axis or an easy plane for the spins. Such preferred directions are captured by the so-called XXZ model described by the Hamiltonian

$$\hat{H} = J \sum_{\langle ij \rangle} \left(\hat{S}_i^x \hat{S}_j^x + \hat{S}_i^y \hat{S}_j^y + \Delta \hat{S}_i^z \hat{S}_j^z \right). \tag{4.25}$$

For an *anisotropy* $\Delta = 0$, we find the quantum XY model, for $\Delta = 1$ we have the Heisenberg model, while for $\Delta \to \infty$, we recover the Ising model. Note that all these models conserve the total magnetization $\hat{M}^z$.

4.2.4 Mapping of spin models to (spinless) fermions in 1D

The raising and lowering operators $\hat{S}^{\pm} = \hbar\sigma^{\pm}$ satisfy $(\sigma_i^+)^2 = 0$, in other words a spin can be flipped only once; attempting to flip it twice annihilates the state. This property resembles the Pauli principle for fermions, where acting with the same operator twice anihiliates any state. Even though there is a fundamental difference in the operators acting on different sites of a spin system and a fermionic system, namely the commutation relations of the operators, we can map the above introduced spin models to (spinless) fermions in one dimension. The mapping that achieves this and takes care of the right commutation relations is the Jordan-Wigner transformation and is given by

$$\hat{c}_i = \prod_{j<i} \left(\sigma_j^z\right) \sigma_i^+, \tag{4.26}$$

and

$$\hat{c}_i^\dagger = \prod_{j<i} \left(\sigma_j^z\right) \sigma_i^-. \tag{4.27}$$

One checks that the so-defined operators satisfy

$$\{\hat{c}_i, \hat{c}_j^\dagger\} = \delta_{ij}, \qquad \{\hat{c}_i, \hat{c}_j\} = \{\hat{c}_i^\dagger, \hat{c}_j^\dagger\} = 0, \tag{4.28}$$

meaning they furnish a Hilbert space of a chain with one fermionic degree of freedom per site. Notice that the fermionic operators, when written in terms of the spin operators are *non-local*. This non-locality is necessary to build in the fermionic anticommutation relations and caused by the "string" operator

$$\hat{K}_i = \prod_{j<i} \left(\sigma_j^z\right), \tag{4.29}$$

which accounts for commuting through the operators to the position i, see also Eq. (4.20).

Using this transformation, the XXZ model becomes (up to chemical potential and constant terms)

$$\hat{H} = \frac{J}{2} \sum_{\langle ij \rangle} \left(\hat{c}_i^\dagger \hat{c}_j + h.c. + 2\Delta \hat{n}_i \hat{n}_j \right) \tag{4.30}$$

with $\hat{n}_i = \hat{c}_i^\dagger \hat{c}_i$, which describes (spinless) fermions hopping on a chain subject to a nearest-neighbor interation $U = J\Delta$. Note how the conserved magnetization $\hat{M}^z$ of the XXZ model is mapped to a conserved particle number $\hat{N} = \sum_i \hat{n}_i$.

Note that we are often interested in simulating fermions. In order to avoid dealing with the anticommutating operators of the fermionic model, we usually map the Hamiltonian to a spin model. For a 1D system with operators that have only local support, the resulting Hamiltonian is always also local.

5 Brute-force methods

After we got acquainted with various (spin-1/2) model systems in the last chapter, we now want to calculate their properties. In this chapter, we learn about brute-force methods for finding the ground state (or a few lowest states) and for the time evolution of a given initial state. For a system of size N, the Hilbert space has dimension 2^N and thus, the Hamiltonian is a matrix of dimension $2^N \times 2^N$. Therefore, only very small system sizes—usually in one dimension—can be tackled. While brute-force approaches are very useful for benchmarking more elaborate methods and sometimes are even the only method available, care has to be taken when interpreting the results in terms of a thermodynamic limit.

5.1 Exact Diagonalization

The most accurate method is completely diagonalizing the Hamiltonian matrix. If we want to obtain the full spectrum, only system sizes $N \sim 20$ are reachable. If we are only interested in the ground state, we can use iterative solvers, in particular the Lanczos algorithm, which allow system sizes $N \sim 40$. This is usually referred to as *exact diagonalization* or ED.

5.1.1 Lanczos Algorithm

To motivate the Lanczos algorithm, we will first take a look at the power method for a matrix A. For concreteness, we will use the Hamiltonian H as the matrix. Then, we start from a random vector

$$\vec{v} = \sum_n c_n \vec{n} \tag{5.1}$$

with $H\vec{n} = \lambda_n \vec{n}$. Applying the Hamiltonian to $\vec{v}$, we find

$$H\vec{v} = \sum_n c_n \lambda_n \vec{n} \tag{5.2}$$

and repeating this process M times, we obtain vectors

$$\vec{v}_k := H^k \vec{v} = \sum_n c_n \lambda_n^k \vec{n}, \qquad k = 0, 1, 2, \ldots, M. \tag{5.3}$$

If λ_{n_0} is the largest eigenvalue (with non-zero coefficient c_{n_0}), we can rewrite

$$\vec{v}_M = c_{n_0} (\lambda_{n_0})^M \left[\vec{n_0} + \sum_{n \neq n_0} \frac{c_n}{c_{n_0}} \left(\frac{\lambda_n}{\lambda_{n_0}} \right)^M \vec{n} \right]. \tag{5.4}$$

For $M \to \infty$, this procedure, thus, converges (slowly) to the state with the largest eigenvalue. The Lanczos algorithm optimizes this crude method.

The (normalized) vectors $\vec{v}_k$ span the *Krylov* space

$$K_M = \text{span} \left\{ \frac{\vec{v}}{\|\vec{v}\|}, \frac{H\vec{v}}{\|H\vec{v}\|}, \frac{H^2\vec{v}}{\|H^2\vec{v}\|}, \ldots, \frac{H^M\vec{v}}{\|H^M\vec{v}\|} \right\}. \tag{5.5}$$

Next, we orthogonalize the basis vectors employing a *Gram-Schmidt orthogonalization*

$$\begin{aligned}
&\vec{r}_0 = \vec{v}/\|\vec{v}\|,\\
&\beta_1\vec{r}_1 = H\vec{r}_0 - \alpha_0\vec{r}_0,\\
&\beta_2\vec{r}_2 = H\vec{r}_1 - \alpha_1\vec{r}_1 - \beta_1\vec{r}_0,\\
&\dots\\
&\beta_{n+1}\vec{r}_{n+1} = H\vec{r}_n - \alpha_n\vec{r}_n - \beta_n\vec{r}_{n-1},\\
&\dots\\
&\beta_M\vec{r}_M = H\vec{r}_{M-1} - \alpha_{M-1}\vec{r}_{M-1} - \beta_{M-1}\vec{r}_{M-2},
\end{aligned} \tag{5.6}$$

where

$$\begin{aligned}
\alpha_n &= \vec{r}_n^\dagger H\vec{r}_n,\\
\beta_n &= \left|\vec{r}_n^\dagger H\vec{r}_{n-1}\right|.
\end{aligned} \tag{5.7}$$

This yields an orthonormal basis with $\vec{r}_n^\dagger\vec{r}_m = \delta_{nm}$. As the orthogonality condition does not determine the phases of the basis vectors, β_n can be chosen to be real and positive. Expressed in this basis, the Hamiltonian H is a tridiagonal matrix

$$H^{(M)} \equiv \begin{pmatrix}
\alpha_0 & \beta_1 & 0 & \dots & 0\\
\beta_1 & \alpha_1 & \ddots & \ddots & \vdots\\
0 & \ddots & \ddots & \ddots & 0\\
\vdots & \ddots & \ddots & \ddots & \beta_M\\
0 & \dots & 0 & \beta_M & \alpha_M
\end{pmatrix}. \tag{5.8}$$

The eigenvalues of $H^{(M)}$ provide a good approximation of the eigenvalues of the Hamiltonian H, with the extremal eigenvalues converging very fast, such that $M \ll 2^N$ steps are required. Note that $H^{(M)}$ is only a $(M+1)\times(M+1)$ matrix, so it is easy to diagonalize and check for convergence. Furthermore, for the construction of the Hamiltonian, we only need to store three vectors of size 2^N in memory, which makes the Lanczos algorithm very efficient when compared to dense matrix eigensolvers which requires storage of order $(2^N)^2$. Furthermore, note that we do not need the explicit form of the Hamiltonian; the Lanczos algorithm only needs a linear operator, in other words the rules for matrix multiplication with H, $H : \vec{v} \mapsto H\vec{v}$ as an input. This allows for an extremely efficient implementation of matrix-vector multiplication for spin Hamiltonians, as we will discuss below.

It is also straight forward to compute the eigenvectors of $H^{(M)}$. They are, however, given in the Lanczos basis $\{\vec{r}_n\}$. 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 $\vec{r}_n$, 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 $\vec{r}_0$ we construct the vectors $\vec{r}_n$ iteratively and perform the basis transformation as we go along.

One problem that we need to be aware of: in principle all Lanczos vectors $\vec{r}_n$ are orthogonal to each other. But round-off errors spoil this property, and "new" low-lying eigenvalues appear throughout the iteration. These are called *ghost states* and have to be removed.

Numerically stable and efficient implementations of the Lanczos algorithm can be obtained as part of open-source packages (EIGEN library for C++, scipy.linalg module for Python).

5.1.2 Spin-1/2 Hamiltonians

A straight forward implementation of a spin Hamiltonian can be achieved using direct products of spin operators. For this purpose, we add identity matrices on all sites outside the operator's support. For the example of an Ising term, we can write in matrix form

$$\hat{S}_i^z \hat{S}_{i+1}^z = \underbrace{\mathbb{1}}_{n=1} \otimes \cdots \otimes \underbrace{\mathbb{1}}_{n=i-1} \otimes \underbrace{\hat{S}^z}_{n=i} \otimes \underbrace{\hat{S}^z}_{n=i+1} \otimes \underbrace{\mathbb{1}}_{n=i+2} \cdots . \tag{5.9}$$

As noted above, for a system of length N, the total Hamiltonian matrix has $O(2^N \times 2^N)$ terms, a prohibitively large number for "large" N. The good news is that all (spin) Hamiltonians introduced in the previous section contain only $O(N^2)$ two-body terms. As a consequence, most of the matrix elements of the Hamiltonian matrix are zero. However, even the storage of a sparse matrix with $O(N^2 2^N)$ matrix elements is prohibitive for large N. Here, the Lanczos algorithm again provides a route forward: For the algorithm's application, the full matrix is never necessary, but only the action of H on a state $\vec{v}$. What we need then are (1) a representation of the basis states suited for the problem and (2) an efficient way of calculating $H\vec{v}$.

The 2^N basis states of a quantum spin-1/2 system can simply be represented as N-bit strings. To do so, we represent every basis state by a single integer between $0, \ldots, 2^N - 1$. The value of the i-th bit of the integer corresponds to the orientation of the i-th spin in that configuration: A value of 0 denotes a $\downarrow$ spin, while 1 corresponds to $\uparrow$. For $N = 3$, the basis states then are

$$\{|\{s_i\}\rangle\} = \{\underset{0=000_2}{|\downarrow\downarrow\downarrow\rangle}, \underset{1=001_2}{|\downarrow\downarrow\uparrow\rangle}, \underset{2=010_2}{|\downarrow\uparrow\downarrow\rangle}, \underset{3=011_2}{|\downarrow\uparrow\uparrow\rangle}, \underset{4=100_2}{|\uparrow\downarrow\downarrow\rangle}, \underset{5=101_2}{|\uparrow\downarrow\uparrow\rangle}, \underset{6=110_2}{|\uparrow\uparrow\downarrow\rangle}, \underset{7=111_2}{|\uparrow\uparrow\uparrow\rangle}\}. \tag{5.10}$$

Using these basis states, an efficient implementation of the Hamiltonian requires only bitwise operations. As an example, we discuss the Heisenberg model. We can implement the diagonal term $\hat{S}_i^z \hat{S}_{i+1}^z$ using an XOR between the original state s and the state with all bits shifted by one, $\tilde{s}$. In Python code this would be `s_tilde = s ^ (s>>1)`. For each aligned pair of spins, this will set a bit to 0, while for antialigned spins, this sets a bit to 1. For example, using the state $s = 3 = 011_2$, we find for $\tilde{s} = 011_2 \,\hat{}\, 001_2 = 010_2$. Now we can simply count the number of bits 0 and 1 in $\tilde{s}$ to determine the diagonal matrix value of the state s. Similarly, for the $\hat{S}_i^+ \hat{S}_{i+1}^- + \text{h.c.}$ term, we can use $\tilde{s}$ and if a bit is set at position r, then we flip the two neighboring spins by an XOR operation of s with `s_new = s ^ (3<<r)`. The Hamiltonian then has a matrix element between s and s_{new} with matrix element $J/2$.

5.1.3 Symmetries

If a Hamiltonian obeys some symmetries, such as total magnetization, particle number, or momentum, we can block diagonalize the Hamiltonian and solve within the

symmetries' eigenspaces. As an example, we discuss in the following the Transverse Field Ising Model, for which we can define a parity operation $\hat{P}$ through

$$\hat{P} = \prod_i \sigma_i^x, \tag{5.11}$$

which is a symmetry of $\hat{H}$, meaning

$$[\hat{P}, \hat{H}] = 0. \tag{5.12}$$

Furthermore, applying parity twice yields the original state,

$$\hat{P}^2 = \mathbb{1}. \tag{5.13}$$

From Eq. (5.13), it follows that the parity operator has eigenvalues $P = \pm 1$. Working in an eigenbasis $|\alpha_\pm\rangle$ of $\hat{P}$, all matrix elements between the two eigenspaces vanish, since for any two states $|+\rangle$ and $|-\rangle$,

$$\langle +|\hat{H}|-\rangle = \langle +|\hat{P}\hat{H}|-\rangle = \langle +|\hat{H}\hat{P}|-\rangle = -\langle +|\hat{H}|-\rangle = 0. \tag{5.14}$$

Implementation

To make use of a symmetry $\hat{S}$, we need to:

1. Find the eigenspaces of the symmetry operator $\hat{S}$ and an appropriate basis. For this purpose, we start from each basis state and apply $\hat{S}$ exactly M times until we find the initial state again. The set of states related by the symmetry is called an orbit. Note that for $\hat{P}$, $M = 2$. For each eigenspace of the symmetry, we construct eigenfunctions out of these orbits (if possible) and denote them by a representative and the eigenvalue. For the case of $N = 3$ and parity, for example, we find the orbits

   $$\{|\uparrow\uparrow\uparrow\rangle, |\downarrow\downarrow\downarrow\rangle\}, \{|\uparrow\uparrow\downarrow\rangle, |\downarrow\downarrow\uparrow\rangle\}, \{|\uparrow\downarrow\uparrow\rangle, |\downarrow\uparrow\downarrow\rangle\}, \{|\downarrow\uparrow\uparrow\rangle, |\uparrow\downarrow\downarrow\rangle\} \tag{5.15}$$

   and we can construct eigenstates such as

   $$|\downarrow\downarrow\downarrow, \pm\rangle = \frac{1}{\sqrt{2}}\big(|\downarrow\downarrow\downarrow\rangle \pm |\uparrow\uparrow\uparrow\rangle\big). \tag{5.16}$$

2. Construct the Hamiltonian in the eigenspaces. Note that if we want to only define a linear operator (instead of an explicit Hamiltonian matrix), fast look-up tables that map an index to a representative, find the representative of a given state in an orbit, and find the index of a representative are essential.

5.2 Time evolution

As for the single-particle case, we want to solve the time evolution of some initial state, governed by

$$i\hbar\partial_t |\psi(t)\rangle = \hat{H}(t) |\psi(t)\rangle \tag{5.17}$$

both for constant Hamiltonians $\hat{H}(t) \equiv \hat{H}$ and for time-dependent ones.

5.2.1 Time-independent Hamiltonians

The exact solution to Eq. (5.17) can be obtained by integration, namely

$$|\psi(t+\Delta t)\rangle = e^{-i\hat{H}\Delta t/\hbar}\,|\psi(t)\rangle\,. \tag{5.18}$$

However, the exponential of the sparse matrix is dense, rendering this computationally too costly. Unfortunately, as we have seen in the case of the single-particle time evolution, simply expanding the exponential leads to non-unitary time evolution. In the case of single-particle time evolution with $\hat{H} = \hat{T} + \hat{V}$, we discussed the split-operator method, where $e^{\hat{O}}$ for some operator $\hat{O}$ can be applied exactly. Here, we go along a similar line.

Trotter-Suzuki decomposition

We split the Hamiltonian into a sum of K terms that mutually don't commute,

$$\hat{H} = \sum_{k=1}^{K} \hat{h}_k, \tag{5.19}$$

but which we know how to exponentiate explicitly. We then decompose the time-evolution operator for a small time step Δt into multiple products of the non-commuting terms in the Hamiltonian using the Baker-Campbell-Hausdorff formula. To first order, we have

$$e^{-i\hat{H}\Delta t/\hbar} = \prod_{k=1}^{K} e^{-i\hat{h}_k\Delta t/\hbar} + O\left((\Delta t)^2\right). \tag{5.20}$$

The second-order version reads

$$e^{-i\hat{H}\Delta t/\hbar} = \hat{S}\left(\frac{\Delta t}{2}\right) + O\left((\Delta t)^3\right), \tag{5.21}$$

where

$$\hat{S}\left(\Delta t\right) = \left(\prod_{k=1}^{K} e^{-i\hat{h}_k\Delta t/\hbar}\right)\left(\prod_{k=K}^{1} e^{-i\hat{h}_k\Delta t/\hbar}\right). \tag{5.22}$$

For $K = 2$, this simplifies to

$$e^{-i\hat{H}\Delta t/\hbar} = e^{-i\hat{h}_1\Delta t/2\hbar}e^{-i\hat{h}_2\Delta t/\hbar}e^{-i\hat{h}_1\Delta t/2\hbar}, \tag{5.23}$$

which, as in the split-operator case, leads to half-time steps at the beginning and at the end of the time evolution.

Examples

A: Transverse Field Ising Model

For the case of the TFIM, we split the Hamiltonian into

$$\hat{H} = \underbrace{\sum_{\langle ij\rangle} J_{ij}\sigma_i^z\sigma_j^z}_{\hat{h}_1} \; - \; \underbrace{\sum_i h_i\sigma_i^x}_{\hat{h}_2}. \tag{5.24}$$

Both $\hat{h}_1$ and $\hat{h}_2$ can be explicitly exponentiated. First, the Ising term $\hat{h}_1$ is diagonal in the z basis and can be written as

$$e^{-i\hat{h}_1 \Delta t/\hbar} = \prod_{\langle ij \rangle} e^{-i\Delta t J_{ij} s_i^z s_j^z/\hbar}, \tag{5.25}$$

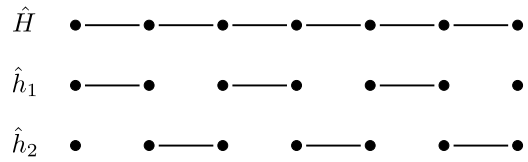
where s_i^z is the eigenvalue of σ_i^z. The local field in $\hat{h}_2$, on the other hand, is not diagonal, but can be exponentiated for each site individually,

$$e^{i\Delta t h_i \sigma^x/\hbar} = \begin{pmatrix} \cos(\Delta t h_i/\hbar) & i\sin(\Delta t h_i/\hbar) \\ i\sin(\Delta t h_i/\hbar) & \cos(\Delta t h_i/\hbar) \end{pmatrix}. \tag{5.26}$$

Note that, as in the case of exact diagonalization, both terms $\hat{h}_1$ and $\hat{h}_2$ can be either written as a direct product or calculated on the fly given a state in the z basis; we never have to store the full evolution operator.

B: XXZ model

For the XXZ model—and thus, the Heisenberg model—we split the interaction bonds into disjoint sets



$$\hat{H} = \hat{h}_1 + \hat{h}_2 = \hat{H}_{\text{even}} + \hat{H}_{\text{odd}} \tag{5.27}$$

with

$$\begin{aligned} \hat{H}_{\text{even}} &= J \sum_i \left[\frac{1}{2} \left(\hat{S}_{2i}^+ \hat{S}_{2i+1}^- + \hat{S}_{2i}^- \hat{S}_{2i+1}^+ \right) + \Delta \hat{S}_{2i}^z \hat{S}_{2i+1}^z \right], \\ \hat{H}_{\text{odd}} &= J \sum_i \left[\frac{1}{2} \left(\hat{S}_{2i-1}^+ \hat{S}_{2i}^- + \hat{S}_{2i-1}^- \hat{S}_{2i}^+ \right) + \Delta \hat{S}_{2i-1}^z \hat{S}_{2i}^z \right]. \end{aligned} \tag{5.28}$$

Both $\hat{H}_{\text{even}}$ and $\hat{H}_{\text{odd}}$ contain only mutually commuting terms that we can individually exponentiate, yielding for two neighboring sites

$$\exp\left(i\Delta t \, \Delta \frac{J}{4} \right) \begin{pmatrix} e^{-i\Delta t \, \Delta \frac{J}{2}} & 0 & 0 & 0 \\ 0 & \cos(\Delta t J/2) & -i\sin(\Delta t J/2) & 0 \\ 0 & -i\sin(\Delta t J/2) & \cos(\Delta t J/2) & 0 \\ 0 & 0 & 0 & e^{-i\Delta t \, \Delta \frac{J}{2}} \end{pmatrix}. \tag{5.29}$$

Imaginary-time evolution

We can use the time-evolution protocol discussed above to obtain the ground state by replacing

$$it \to \tau. \tag{5.30}$$

The *imaginary-time evolution* reads

$$|\psi(\tau)\rangle = e^{-\tau\hat{H}}\,|\psi(0)\rangle\,. \tag{5.31}$$

Note an important difference with respect to the real (or physical) time evolution: in the imaginary-time case, the evolution is no longer unitary, it will change the norm of the initial state. However, we can use imaginary-time evolution as an alternative scheme to find the ground state of a given Hamiltonian $\hat{H}$.

To see this, we start from a random state $|\psi_0\rangle = \sum_i c_i\,|i\rangle$, where $\hat{H}\,|i\rangle = E_i\,|i\rangle$. Then,

$$e^{-\tau\hat{H}}\,|\psi(0)\rangle = \sum_i c_i e^{-E_i\tau}\,|i\rangle = e^{-E_0\tau}\left[c_0\,|0\rangle + \sum_{k>0} e^{-\Delta E_k\tau} c_k\,|k\rangle\right], \tag{5.32}$$

where $\Delta E_k = E_k - E_0 \geq 0$. Thus, provided that the initial state has not been chosen orthogonal to the exact ground state, i.e. $|c_0| \neq 0$, the imaginary-time evolution converges to the exact ground state by suppressing the amplitudes of excited states exponentially fast in the product $\Delta E_k\tau$.

5.2.2 Time-dependent Hamiltonians

Up to now, we have only considered the time evolution of static, time-independent Hamiltonians. If the Hamiltonian is time dependent, as is typically the case in simulations of e.g. optical-trap loading, transport-property studies, or the Floquet-engineered Hamiltonians, we have to modify our numerical schemes accordingly. The fundamental difference stems from the fact that, for time steps Δt larger than the characteristic time scale of the Hamiltonian,

$$|\psi(t+\Delta t)\rangle \neq e^{-i\hat{H}(t)\Delta t/\hbar}\,|\psi(t)\rangle\,, \tag{5.33}$$

since typically the Hamiltonian does not commute with itself at different times, i.e. $\left[\hat{H}(t_1), \hat{H}(t_2)\right] \neq 0$ for $t_1 \neq t_2$. Thus, we need to consider the time-evolution operator

$$\hat{U}(t,t') = \hat{T}_s \exp\left(-\frac{i}{\hbar}\int_t^{t'} \hat{H}(s)\,\mathrm{d}s\right) \tag{5.34}$$

instead, where $\hat{T}_s$ is the time-ordering operator.

The Magnus expansion

There are various ways of dealing with the time-ordering operator. It is often convenient to use the Magnus expansion

$$\hat{U}(\Delta t) = \exp\left(-\frac{i}{\hbar}\Delta t\bar{H}_t\right), \tag{5.35}$$

where

$$\bar{H}_t = \bar{H}_t^{(1)} + \bar{H}_t^{(2)} + \ldots, \tag{5.36}$$

and each of the terms is given by

$$\begin{aligned}\bar{H}_t^{(1)} &= \frac{1}{\Delta t}\int_t^{t+\Delta t} \hat{H}(s)\,\mathrm{d}s\,,\\ \bar{H}_t^{(2)} &= -\frac{i}{\Delta t}\int_t^{t+\Delta t}\mathrm{d}s\int_t^{s}\mathrm{d}l\,\Big[\hat{H}(s),\hat{H}(l)\Big],\end{aligned} \tag{5.37}$$

and so on. Notice that the first term $\bar{H}_t^{(1)}$ corresponds to the average value of the Hamiltonian in the interval Δt and the second term contains a correction proportional to the commutator of the Hamiltonian at different times. Taking only the first term in this expansion, one finds

$$\hat{U}(\Delta t) = \exp\left(-\frac{i}{\hbar}\Delta t \bar{H}_t^{(1)}\right) + O\left((\Delta t)^2\right), \tag{5.38}$$

for which we can apply all of the previously discussed techniques, e.g. the Trotter-Suzuki expansion, by simply considering an effective static Hamiltonian $\bar{H}_t^{(1)}$ at each time step. Notice that if one does not consider (at least) the time-averaged Hamiltonian $\bar{H}_t^{(1)}$, a large $O(1)$ error can be accumulated during the integration of the Schrödinger equation.

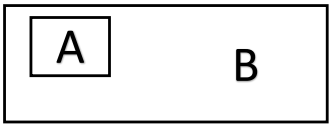
6 Matrix Product States

We have seen that the exponential growth of the Hilbert space dimension is a major difficulty even when only representing a state. In this chapter, we will learn how tensor networks, in particular Matrix Product States, can compress the state representation in the relevant part of the Hilbert space and allow for efficient calculations. The crucial insight for this compression comes from quantum information theory and concerns the entanglement in ground states.

6.1 Bipartite entanglement

6.1.1 Reduced density matrix

Imagine a system composed of two parts, A and B, with $|\psi\rangle \in \mathcal{H} = \mathcal{H}_\mathrm{A} \otimes \mathcal{H}_\mathrm{B}$.



If we assume that we only have access to subsystem A, in other words we want to measure $M_\mathrm{A} \otimes \mathbb{1}_\mathrm{B}$, how can we characterize such a measurement? Labeling states in $\mathcal{H}$ by $|i\rangle \otimes |j\rangle = |i,j\rangle$, where $|i\rangle \in \mathcal{H}_\mathrm{A}$, $|j\rangle \in \mathcal{H}_\mathrm{B}$, we find for a pure state $|\psi\rangle$,

$$\langle\psi|M_\mathrm{A} \otimes \mathbb{1}|\psi\rangle = \sum_{i,j}\sum_{i',j'} \psi^*_{ij}\psi_{i'j'} \langle i,j|M_\mathrm{A} \otimes \mathbb{1}|i',j'\rangle \tag{6.1}$$

$$= \sum_{i,i',j} \psi^*_{ij}\psi_{i'j} \langle i|M_\mathrm{A}|i'\rangle = \mathrm{Tr}_\mathrm{A}\left(\rho_\mathrm{A} M_\mathrm{A}\right), \tag{6.2}$$

where

$$\mathrm{Tr}_X(\cdot) = \sum_n {}_X\langle n| \cdot |n\rangle_X \tag{6.3}$$

denotes a trace over the degrees of freedom of subsystem X and we have introduced the *reduced density operator*

$$\rho_\mathrm{A} = \sum_{i,i',j} \psi^*_{ij}\psi_{i'j} |i'\rangle_\mathrm{A} \langle i|_\mathrm{A} = \mathrm{Tr}_\mathrm{B}\left(|\psi\rangle\langle\psi|\right). \tag{6.4}$$

It follows from the definition in Eq. (6.4) that

1. $\rho_\mathrm{A} = \rho_\mathrm{A}^\dagger$,
2. $\rho_\mathrm{A} \geq 0$,
3. $\mathrm{Tr}_\mathrm{A}\,\rho_\mathrm{A} = 1$.

In other words, ρ_A is indeed a density matrix as introduced in Sec. 2.1.4.

6.1.2 Entanglement entropy and Schmidt decomposition

The (von Neumann) *entanglement entropy* between subsystems A and B is defined as

$$S = -\operatorname{Tr}(\rho_{\mathrm{A}} \log \rho_{\mathrm{A}}) = -\operatorname{Tr}(\rho_{\mathrm{B}} \log \rho_{\mathrm{B}}) . \tag{6.5}$$

An *entangled state* is a state, where ρ_{A} and ρ_{B} describe mixed states, which then implies $S \neq 0$.

Singular Value decomposition and Schmidt decomposition

A matrix M of dimension $m \times n$ allows for a decomposition of the form

$$M = USV^{\dagger}, \tag{6.6}$$

where U is of dimension $m \times \min(m,n)$ with $U^{\dagger}U = \mathbb{1}$, V is of dimension $\min(m,n) \times n$ with $V^{\dagger}V = \mathbb{1}$ and S is diagonal of dimension $\min(m,n) \times \min(m,n)$. The diagonal elements of S are called *singular values* and the number of non-zero elements is the *rank* of M. An important property of the SVD is that the matrix M can be well approximated by *truncating* the smallest singular values, leading to a new matrix M' with smaller rank.

The high practical importance of SVD for quantum-state representation becomes clear when looking at a state $|\psi\rangle = \sum_{i,j} \psi_{ij} |i\rangle \otimes |j\rangle$, which now can be written as

$$|\psi\rangle = \sum_{\alpha=1}^{\min(d_{\mathrm{A}},d_{\mathrm{B}})} \lambda_\alpha |\phi_\alpha\rangle_{\mathrm{A}} \otimes |\phi_\alpha\rangle_{\mathrm{B}} , \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq 0, \tag{6.7}$$

where the Schmidt states $|\phi_\alpha\rangle_{\mathrm{A(B)}}$ are orthonormal states of A (B) and d_A and d_B are the corresponding Hilbert space dimensions. If A and B are both of length $N/2$, then $d_A = d_B = d^{N/2}$. The λ_α's are called Schmidt values, and normalization of $|\psi\rangle$ implies $\sum_\alpha \lambda_\alpha^2 = 1$.

Using the Schmidt decomposition, we immediately find $\rho_{\mathrm{A}} = \sum_\alpha \lambda_\alpha^2 |\phi_\alpha\rangle_{\mathrm{A}} \langle\phi_\alpha|_{\mathrm{A}}$, $\rho_{\mathrm{B}} = \sum_\alpha \lambda_\alpha^2 |\phi_\alpha\rangle_{\mathrm{B}} \langle\phi_\alpha|_{\mathrm{B}}$ and thus, the entanglement entropy is $S = -\sum_\alpha \lambda_\alpha^2 \log \lambda_\alpha^2$. Examples of Schmidt values and corresponding entanglement entropies for specific states are (for a cut dividing the system into two equal parts of length $N/2$:

- product state: $\lambda_1 = 1, \lambda_{\alpha>1} = 0 \quad \Rightarrow \quad S = 0$,
- maximally entangled state: $\lambda_i = 1/\sqrt{d^{N/2}} \ \forall i \quad \Rightarrow \quad S = S_{\max} = \frac{N}{2} \log d$,
- for a random state, the entanglement entropy is close to $S_{\max}$ ($S \approx \frac{N}{2} \log d - 1/2$) .

Area law of entanglement

The ground state of a gapped (local) Hamiltonian obeys the *area law* of entanglement: for a subsystem A of linear dimension L, the entanglement entropy scales as $S = L^{D-1}$. Thus, for a one-dimensional system, the entanglement does not depend on the size L of A, $S(L) \sim const$ (for $L > \xi$, some characteristic length scale). This should be contrasted with the entanglement entropy of a gapless system, $S \sim \log L$. While a random state requires an exponentially increasing number of Schmidt values (with the system size), ground states only require a finite number of them, and are therefore "close" to product states and can be efficiently represented.

6.2 Matrix Product States

We can generalize product states with $\psi_{i_1,\ldots,i_N} = \phi^{[1]i_1} \cdot \ldots \cdot \phi^{[N]i_N}, \quad \phi^{[n]i_n} \in \mathbb{C}$ to *matrix product states* (MPS):

$$\psi_{i_1,\ldots,i_N} = A^{[1]i_1} \cdot \ldots \cdot A^{[N]i_N}, \tag{6.8}$$

where $\{A^{[n]}\}$ is a set of rank-3 tensors, in other words $A^{[n]i_n}$ are matrices. The matrices have dimension $\chi_n \times \chi_{n+1}$ and χ_n is the *bond dimension* of the bond n. Note that for Eq. (6.8) to yield a scalar, we have $\chi_1 = \chi_{N+1} = 1$. For simplicity of notation, we will drop in the following the index $[n]$, i.e., we will write A^{i_n}.

A simple example is the GHZ or 'cat' state,

$$|\psi\rangle = \frac{1}{\sqrt{2}} \left(|\uparrow\uparrow \ldots \uparrow\rangle + |\downarrow\downarrow \ldots \downarrow\rangle \right), \tag{6.9}$$

which can be written as a MPS with

$$A^{\uparrow} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad A^{\downarrow} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \tag{6.10}$$

Another famous, but more involved, example is the Affleck, Kennedy, Lieb, Tasaki (AKLT) state. It is the exact ground state of the spin-1 Hamiltonian

$$\hat{H} = \sum_j \hat{\vec{S}}_j \cdot \hat{\vec{S}}_{j+1} + \frac{1}{3} \left(\hat{\vec{S}}_j \cdot \hat{\vec{S}}_{j+1} \right)^2. \tag{6.11}$$

The state is constructed by interpreting each spin-1 as two spins-1/2 projected onto the triplet sector. Neighboring spins are then in a spin-singlet configuration, see Fig. 4. The MPS representation of this state is

$$A^{+} = \sqrt{\frac{2}{3}}\sigma^{+}, \quad A^{0} = -\sqrt{\frac{1}{3}}\sigma^{z}, \quad A^{-} = -\sqrt{\frac{2}{3}}\sigma^{-}. \tag{6.12}$$

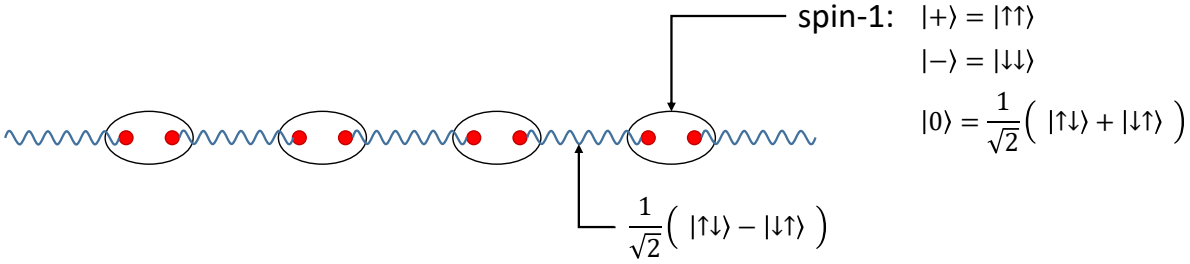
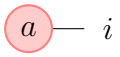


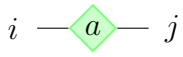
Figure 4: AKLT state. Each black ellipse denotes a single spin-1, which can be in one of three states: $|+\rangle, |-\rangle, |0\rangle$. Each spin-1 is represented as two spins-1/2 (red dots), such that spins-1/2 algebra, corresponding to a triplet state, is projected onto the algebra of a single spin-1. At the same time neighboring spins-1/2 from different spin-1's are always in a singlet state.

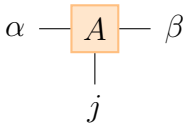
6.2.1 Tensor network notation

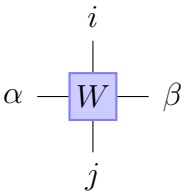
To simplify the following discussion, we introduce a useful diagrammatic representation of tensor networks. In particular, we introduce:

scalar: $a =$ 

vector: $a_i =$ 

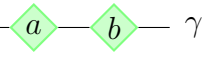
matrix: $a_{ij} =$ 

tensor of rank 3: $A^j_{\alpha\beta} =$ 

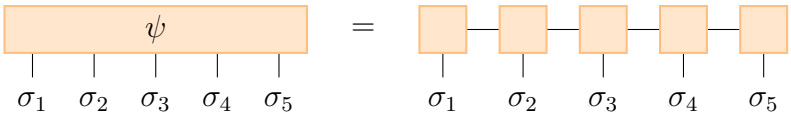
tensor of rank 4: $W^{ij}_{\alpha\beta} =$ 

Note that the shapes and colors used are arbitrary, the number of legs coming out, however, defines the object.

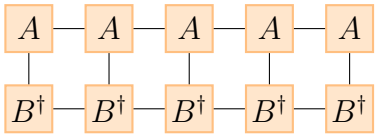
Using this notation, we can simply write for a matrix multiplication

$c_{\alpha\gamma} = \sum_\beta a_{\alpha\beta} b_{\beta\gamma} =$ 

The decomposition of a state into its MPS form can then be written as

$\psi_{\sigma_1 \ldots \sigma_5} =$ 

and the overlap of two states $|\psi\rangle$ and $|\phi\rangle$ is

$\langle\phi|\psi\rangle =$ 

6.2.2 Constructing an MPS from a generic state

Starting from a general state with $\psi_{\sigma_1 \ldots \sigma_N}$, Fig. 5 (a), we first flatten this N-rank $d \times d \times \cdots \times d$ tensor into a $d \times d^{N-1}$ matrix $\psi_{(\sigma_1, \sigma_2 \ldots \sigma_N)}$, to perform a SVD of the left-most site,

$$\begin{aligned} \psi_{\sigma_1 \sigma_2 \ldots \sigma_N} &= \sum_{a_1=1}^{d} U_{\sigma_1, a_1} S_{a_1, a_1} (V^\dagger)_{a_1, \sigma_2 \ldots \sigma_N} \\ &= \sum_{a_1=1}^{d} A^{\sigma_1}_{a_1} \psi_{(a_1, \sigma_2 \ldots \sigma_N)}, \end{aligned} \tag{6.13}$$

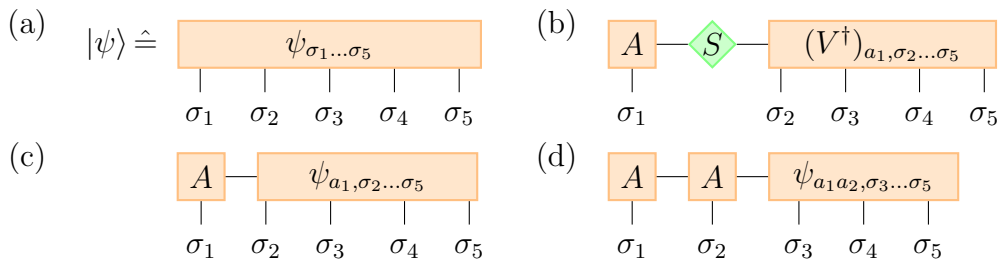


Figure 5: Construction of a MPS form for a general state Ψ for a system of size N=5.

where we interpreted the rows of U as column vectors $A^{\sigma_1}_{a_1}$ [Fig. 5 (b)] and we multiplied $S_{a_1,a_1}(V^\dagger)_{a_1,\sigma_2 \ldots \sigma_N}$ [Fig. 5 (c)] . Next, we group $\psi_{a_1\sigma_2,\sigma_3 \ldots \sigma_N}$ to decompose the second bond

$$\begin{aligned} \psi_{\sigma_1\sigma_2 \ldots \sigma_N} &= \sum_{a_1=1}^{d} \sum_{a_2=1}^{d^2} A^{\sigma_1}_{a_1} U_{a_1\sigma_2,a_2} S_{a_2,a_2} (V^\dagger)_{a_2,\sigma_3 \ldots \sigma_N} \\ &= \sum_{a_1=1}^{d} \sum_{a_2=1}^{d^2} A^{\sigma_1}_{a_1} A^{\sigma_2}_{a_1,a_2} \psi_{(a_2\sigma_3,\sigma_4 \ldots \sigma_N)} \end{aligned} \tag{6.14}$$

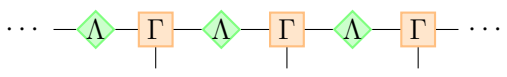
see Fig. 5 (d). We can then continue this procedure until we reach the right end of the chain. Note that by construction, the A matrices are (left) normalized

$$\sum_{\sigma_l} A^{\sigma_l \dagger} A^{\sigma_l} = \mathbb{1}. \tag{6.15}$$

Looking at these first two steps, we already note a problem: the bond dimension of the matrices χ_n increases exponentially $\chi_n \sim d^n$ for $n < N/2$. This is, however, no surprise, as so far, we have simply rewritten the state, thus we need an exponential number of parameters. For area law states, we should be close to a product state ($\chi_n = 1, \forall n$) and we thus expect the singular values to decrease rapidly. Therefore, we can compress the state by discarding (truncating) all Schmidt values below some threshold.

6.2.3 Canonical Form

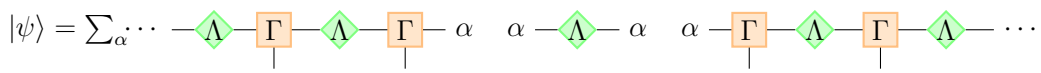
A MPS form for a given state is not uniquely defined. In particular, we can insert an identity $\mathbb{1} = X^{-1}X$ given by an invertible $\chi_n \times \chi_n$ matrix X at bond n, yielding new matrices at site $n-1$ and n with $\tilde{A}^{\sigma_{n-1}} = A^{\sigma_{n-1}} X^{-1}$ and $\tilde{A}^{\sigma_n} = X A^{\sigma_n}$. A particularly useful 'gauge' was introduced by Vidal, where the Schmidt values for each bond are stored instead of incorporating them into a neighboring site matrix. In particular, in the procedure outlined in 6.2.2, we keep the singular values $S_{a,a}$ and save them in a bond matrix Λ, such that $A = \Lambda\Gamma$,



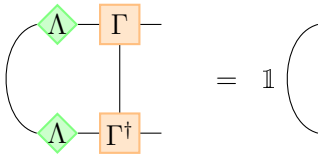
Using this notation, we can find the Schmidt decomposition

$$|\psi\rangle = \sum_{\alpha} |\alpha\rangle_{\mathrm{L}} \, \lambda_{\alpha} \, |\alpha\rangle_{\mathrm{R}} \tag{6.16}$$

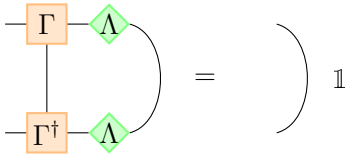
for any cut immediately: in diagrammatic notation, the decomposition is simply given by



Another very useful feature of this notation is the freedom to choose the normalization: while in the above constructed MPS decomposition the $A = \Lambda\Gamma$ matrices are always *left normalized*, see Eq. (6.15), which here can be written as

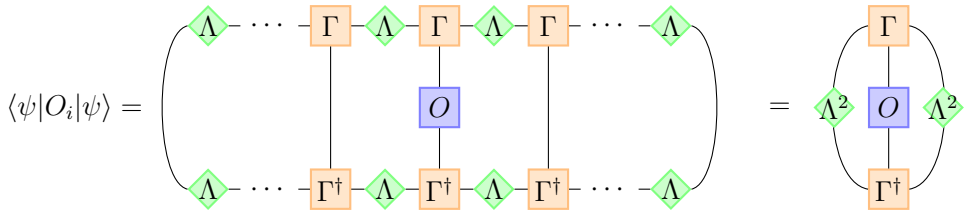


we can easily construct *right normalized* matrices $B = \Gamma\Lambda$, such that



Note that for a more symmetric notation and to simplify the implementation, we usually add trivial 1×1 bond matrices at the beginning and the end of an MPS.

Finally, we can conveniently evaluate expectation values of local operators,

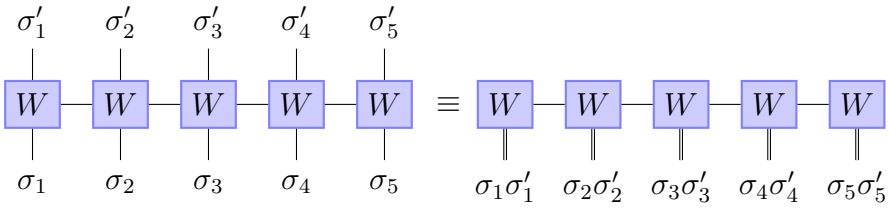


6.2.4 Matrix product operators (MPO)

The idea of factorizing a state into a product of matrices can be readily extended to operators. For this purpose, we write

$$\hat{O} = \sum_{\{\sigma_i, \sigma_i'\}} \left[(W^{[1]})^{\sigma_1 \sigma_1'} (W^{[2]})^{\sigma_2 \sigma_2'} \dots (W^{[N]})^{\sigma_N \sigma_N'} \right] |\sigma_1 \sigma_2 \dots \sigma_N\rangle \langle \sigma_1' \sigma_2' \dots \sigma_N'| \,, \tag{6.17}$$

where $\{W^{[i]}\}$ now represents a set of rank-4 tensors, each of which carries two indices σ_i, σ_i' corresponding to the physical degrees of freedom at site i and two (horizontal) virtual indices. In the case of open boundary conditions, the tensors at the edges are again 'vectors'. Any operator acting on a many-body Hilbert space may be written in the form above and can be constructed in the same way as an MPS by grouping the physical indices of the same site together:



However, note that we usually are not given a random operator to decompose. More commonly, we have local operators that can be constructed directly. The following operators exemplify the general approach:

- Single site operator:

$$\hat{O}_j = \mathbb{1} \otimes \mathbb{1} \otimes \ldots \mathbb{1} \otimes \underbrace{\hat{O}}_{\text{site } j} \otimes \mathbb{1} \cdots \otimes \mathbb{1} \quad \rightarrow \quad \left(W^{[j]}\right)^{\sigma_j \sigma'_j} = \left\langle \sigma_j \middle| \hat{O} \middle| \sigma'_j \right\rangle ,$$
$$\left(W^{[i]}\right)^{\sigma_i \sigma'_i} = \delta_{\sigma_i \sigma'_i} \text{ for } i \neq j. \tag{6.18}$$

  Note that here each matrix $(W^{[i]})^{\sigma_i\sigma'_i}$ is actually a 1×1 matrix (scalar). So, virtual indices are dummy one-dimensional indices (the bond dimension is 1).

- Paramagnetic system: $\hat{H} = -\sum_i h\hat{S}^z_i$

$$\hat{H} = \left(-h\hat{S}^z \otimes \mathbb{1} \otimes \cdots \otimes \mathbb{1}\right) + \left(\mathbb{1} \otimes -h\hat{S}^z \otimes \mathbb{1} \otimes \ldots\right) + \ldots \\ \cdots + \left(\mathbb{1} \otimes \cdots \otimes \mathbb{1} \otimes -h\hat{S}^z\right). \tag{6.19}$$

$$W^{[1]} = \begin{pmatrix} -hS^z & \mathbb{1} \end{pmatrix}, \\ W^{[i]} = \begin{pmatrix} \mathbb{1} & 0 \\ -hS^z & \mathbb{1} \end{pmatrix}, \\ W^{[N]} = \begin{pmatrix} \mathbb{1} \\ -hS^z \end{pmatrix}, \tag{6.20}$$

  where S^z are matrices with matrix elements $(S^z)^{\sigma\sigma'} = \langle\sigma|\hat{S}^z|\sigma'\rangle$ and $\mathbb{1}$ are identity matrices with the corresponding dimension. One can see that in this example the bond dimension is 2.

- Transverse field Ising model: $\hat{H} = -\sum_i \hat{S}^z_i \hat{S}^z_{i+1} + h\sum_i \hat{S}^x_i$ [4]

$$W^{[1]} = \begin{pmatrix} hS^x & -S^z & \mathbb{1} \end{pmatrix}, \\ W^{[i]} = \begin{pmatrix} \mathbb{1} & 0 & 0 \\ S^z & 0 & 0 \\ hS^x & -S^z & \mathbb{1} \end{pmatrix}, \\ W^{[N]} = \begin{pmatrix} \mathbb{1} \\ S^z \\ hS^x \end{pmatrix}, \tag{6.21}$$

  where S^z and S^x are matrices with matrix elements $\langle\sigma|\hat{S}^z|\sigma'\rangle$ and $\langle\sigma|\hat{S}^x|\sigma'\rangle$ correspondingly. Here the bond dimension is 3.

[4] Note that the local Hilbert space is $\mathbb{C}^2$. So, spin operators are represented by Pauli matrices. However, we will denote them as $\hat{S}$ rather than σ to avoid notational ambiguity with physical spin indices σ_i.

6.2.5 Computing expectation values

We would like to compute the expectation value $\langle\psi|\hat{O}|\psi\rangle$ of an operator $\hat{O}$ given in MPO form. The algorithm for such an evaluation can be done starting from the left side and adding sites to the right. Note that we use for simplicity again A matrices instead of Vidal notation. In particular, the algorithm now proceeds as follows:

1. We start by building the *left environment*,

$$L^{b_1}_{a_1,a_1'} = \sum_{\sigma_1,\sigma_1'} \left(A^{\sigma_1}_{a_0,a_1}\right)^\dagger W^{\sigma_1,\sigma_1'}_{b_0,b_1} \left(A^{\sigma_1'}_{a_0',a_1'},\right) \tag{6.22}$$

   where $a_0 = a_0' = b_0 = 1$ are dummy variables.

2. Then, we add the matrix of state $|\psi\rangle$ at position $i = 2$:

$$^{(2)}L^{\sigma_2',b_1}_{a_1,a_2'} = \sum_{a_1'} L^{b_1}_{a_1,a_1'} A^{\sigma_2'}_{a_1',a_2'}. \tag{6.23}$$

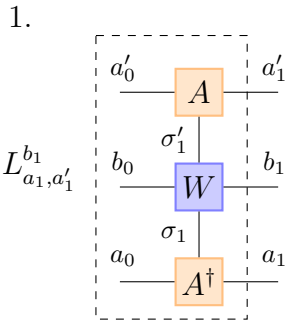
3. Next, we compute the action of $W^{[2]}$:

$$^{(3)}L^{\sigma_2,b_2}_{a_1,a_2'} = \sum_{b_1,\sigma_2'} W^{\sigma_2,\sigma_2'}_{b_1,b_2} {}^{(2)}L^{\sigma_2',b_1}_{a_1,a_2'}. \tag{6.24}$$

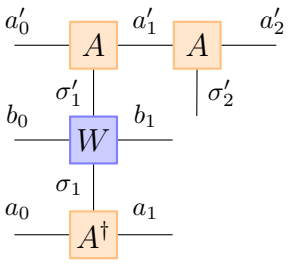
4. Finally, we add the matrix of state $\langle\psi|$ at position $i = 2$ to find the new left environment

$$L^{b_2}_{a_2,a_2'} = \sum_{\sigma_2,a_1} \left(A^{\sigma_2}_{a_1,a_2}\right)^\dagger {}^{(3)}L^{\sigma_2,b_2}_{a_1,a_2'}. \tag{6.25}$$

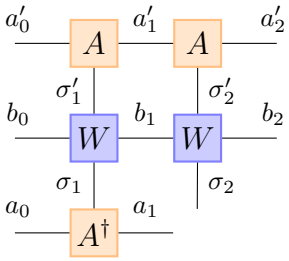
1.



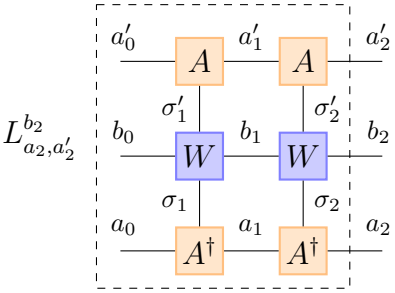
2.



3.



4.



We can then repeat this until we reach the right end of the MPS. The very last step then yields a number, the expectation value of the operator. Note that alternatively, we could have started from the right side, constructing the *right environment* $R^{b_N}_{a_N,a'_N}$. Then, the right environment is moved left by adding sites in the same fashion as above.

We are now ready to tackle the two main tasks we face in many-body quantum physics: finding the ground state of a given Hamiltonian $\hat{H}$ or calculate the time-evolution governed by $\hat{H}$.

6.3 The density matrix renormalization group

With matrix product states, we have a way of efficiently representing one-dimensional (ground) states and evaluating overlaps or expectation values. However, we are usually given a Hamiltonian $\hat{H}$ and we need to find its ground state, in other words find $|\psi\rangle$, which minimizes the energy expectation value

$$E_0 = \min \frac{\langle\psi|\hat{H}|\psi\rangle}{\langle\psi|\psi\rangle}. \tag{6.26}$$

While such an optimization problem can in principle be attacked head on, the corresponding algorithms tend to be rather unstable. Instead, we will use an iterative approach in the following, the density matrix renormalization group or DMRG.

As the name suggests, DMRG is based on renormalization group (RG) theory, in other words the idea that the microscopical characteristics of a system are of minor importance near a critical point or in a bulk phase. To make use of this *scale invariance*, we want to integrate out the high-energy degrees of freedom and explore the behavior of the system under resizing. The DMRG achieves this resizing through an iterative variational method depicted in Fig. 6 that reduces the effective degrees of freedom to those most important for a target (often the ground) state. In its original (*infinite-system*) form, the method grows the system by splitting it into two blocks and adding two sites in between. Then, a candidate for the ground state of the system "left block + two sites + right block", which is a reduced version of the full system, may be found. The candidate ground state is projected onto the Hilbert

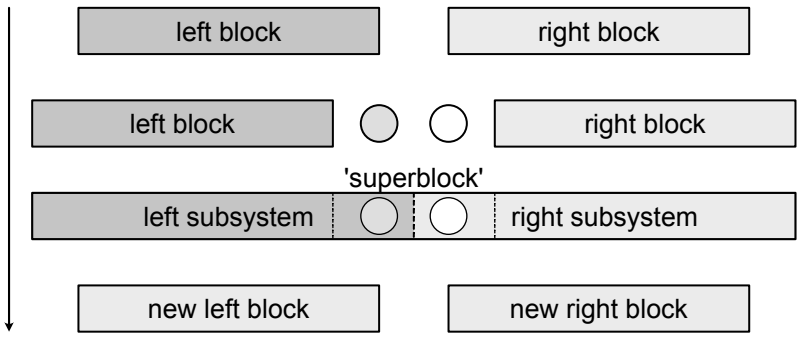


Figure 6: Basic procedure of *infinite-system* dmrg, where the system is grown by adding two sites in the middle, which are then absorbed into new right and left blocks.

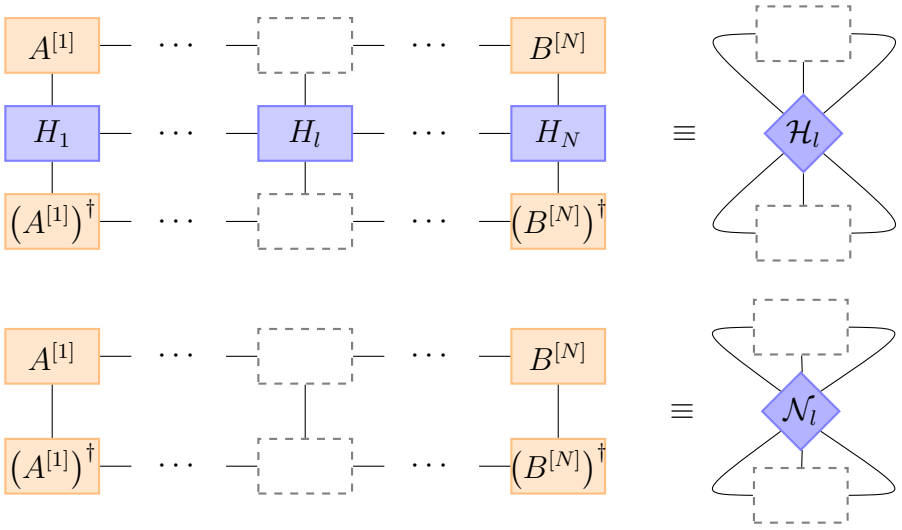
subspace for each "block + one site" using the reduced density matrix. Thus, the relevant states for each new block are updated and we regain the form of the system we started with. This procedure is then repeated until the energy converges. At this point, a 'sweeping' procedure is employed to minimize the energy further. In the following, we will focus on this latter optimization step.

Only later it was realized that DMRG can be understood as a variational method in the space of (fixed-bond-dimension) MPS wave functions to find the ground state. While DMRG comes in various flavors, we will in the following first sketch the general approach for a *single-site* version before giving the details of an actual implementation. Note that what all flavors of DMRG have in common is that the Hamiltonian needs to be provided in MPO form.

First, instead of minimizing Eq. (6.26) with respect to all the Γ and Λ matrices, we can minimize the expectation value $\langle\psi|\hat{H}|\psi\rangle$ under the condition that $\langle\psi|\psi\rangle = 1$. For this purpose, we introduce a Lagrange multiplier to extremize the functional

$$f\left[|\psi\rangle\right] = \langle\psi|\hat{H}|\psi\rangle + \lambda\left(1 - \langle\psi|\psi\rangle\right). \tag{6.27}$$

As mentioned, it is computationally most stable to split up the optimization process and optimize one site l at a time, while keeping the others fixed. We then introduce the following graphical notation:



Here, we contracted all tensors around the site l into $\mathcal{H}_l$ and $\mathcal{N}_l$. These operators act on the combined space of left and right virtual and physical indices, so that their (matrix) dimensions are $\chi_l\chi_{l+1}d \times \chi_l\chi_{l+1}d$. If we write all the parameters of site l as a vector $\vec{a}^{[l]}$, Eq. (6.27) now becomes a quadratic functional of the form

$$f_l\left[\vec{a}^{[l]}\right] = \left(\vec{a}^{[l]}\right)^\dagger \mathcal{H}_l \vec{a}^{[l]} + \lambda\left[1 - \left(\vec{a}^{[l]}\right)^\dagger \mathcal{N}_l \vec{a}^{[l]}\right]. \tag{6.28}$$

To proceed, we group the Γ and Λ matrices in such a way that all the matrices on the left of l are written as $A^{[i<l]} = \Lambda^{[i]}\Gamma^{[i]}$ and all the matrices to the right of l are $B^{[i>l]} = \Gamma^{[i]}\Lambda^{[i+1]}$. Finally, we introduce the matrix $M^{[l]} = \Lambda^{[l]}\Gamma^{[l]}\Lambda^{[l+1]}$, such that the vector $\vec{a}^{[l]}$ is given by the elements $M^{\sigma_l}_{a_l,a_{l+1}} \hat{=} M_{(\sigma_l a_l a_{l+1})}$. The form of the MPS is called a (mixed) canonical form or, equivalently, the MPS is said to be in its unitary gauge. The advantage of absorbing the Λ matrices into A and B matrices

now becomes obvious looking at the above expression for the norm: Since all the A's (B's) are left (right) normalized, the operator $\mathcal{N}_l = \mathcal{N} = \mathbb{1}$.

Minimizing Eq. (6.28) with respect to $\left(\vec{a}^{[l]}\right)^\dagger$ now leads to a regular eigenvalue problem

$$\mathcal{H}_l \vec{a}^{[l]} = \lambda \vec{a}^{[l]}, \tag{6.29}$$

which can be solved using standard numerical libraries (e.g. Lanczos). Note that for reasons of efficiency, one should not actually constructs the object $\mathcal{H}_l$, which has a complexity $O(\chi^4)$, but instead rely on an iterative eigenvalue solver which allows to exploit the structure of the tensor network to reduce the computational cost to $O(\chi^3)$.

To optimize the entire MPS, we start from the left edge of the system working our way to the right edge and then back. This right-left moving defines one *sweep*. By sweeping until the energy λ converges to a predefined accuracy, we obtain an excellent approximations of the ground state in case of a one-dimensional gapped system.

DMRG algorithm

After introducing the general procedure above, let us look at the actual algorithm. The procedure of single-site DMRG is as follows:

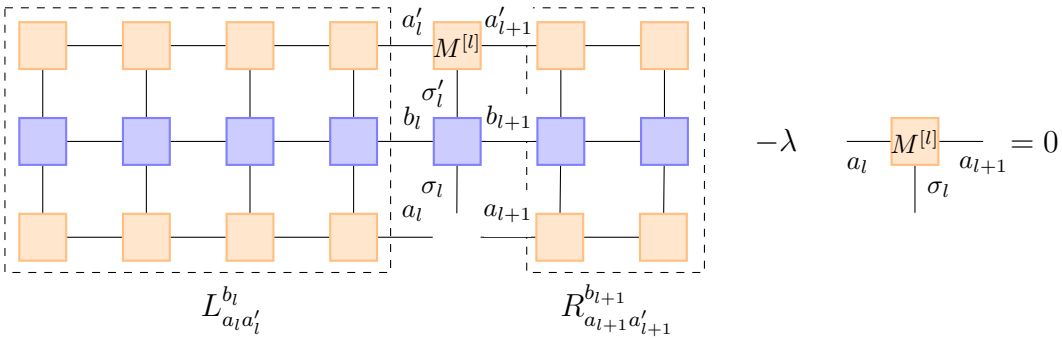
1. Start from an initial guess for $|\psi\rangle$ which is right-normalized, i.e. consists of B matrices only. This is often simply a random state. Note that there should be a non-zero overlap with the ground state.

2. Calculate the right environments, or R-expression, for all site positions L through 2 iteratively as discussed in 6.2.5.

3. Right sweep through sites $l = 1, \ldots, L-1$:

   (a) Solve the standard eigenvalue problem by an iterative eigensolver for $M^{[l]}$. Explicitly, we need to solve

   $$\sum_{\sigma_l', a_l', a_{l+1}', b_l, b_{l+1}} \left(L^{b_l}_{a_l, a_l'} H^{\sigma_l' \sigma_l}_{b_l b_{l+1}} R^{b_{l+1}}_{a_{l+1} a_{l+1}'} \right) M^{\sigma_l'}_{a_l' a_{l+1}'} - \lambda M^{\sigma_l}_{a_l a_{l+1}} = 0. \tag{6.30}$$

   We only need to compute the lowest eigenvalue λ. It provides an estimate for the ground-state energy.

   (b) We perform a SVD on the new $M^{[l]}$ to left normalize the site tensor into $A^{[l]}$. We then multiply the remaining matrices from SVD to $B^{[l+1]}$, which yields the new $M^{[l+1]}$ as a starting guess for the next site.

(c) Build $L^{[l]}$ iteratively.

(d) Move by one site $l \to l+1$ and repeat.

4. Left sweep through sites $l = L, \ldots, 2$

   (a) Solve eigenvalue problem for $M^{[l]}$.

   (b) Right normalize $M^{[l]}$.

   (c) Build $R^{[l]}$.

5. Repeat left and right sweeps until the energy is converged, or better, until $\mathrm{var}(H) < \varepsilon$.

In the discussion so far, we have only updated a single site at a time. While this allows for a straight forward implementation of DMRG, it tends to get stuck in local minima. A more stable implementation, the so-called *two-site* DMRG, groups two sites together to update them simultaneously. An advantage, but also a further difficulty, of this approach is that it allows for dynamically adjusting the bond dimension χ on the fly as one optimizes the wavefunction. This flexibility comes about because in splitting the optimized two-site tensor into single-site tensors using SVD, we can tune the number of states $\chi_{\max}$ kept. Note, however, that keeping a fixed $\chi_{\max}$ in this approach requires finding the MPS with bond dimension $\chi_{\max}$ that best approximates the MPS after SVD, which in general has bond dimension $d\chi_{\max}$. Finally, the single-site version tends to achieve slightly better results than the two-site version for a fixed χ. Thus, a common way of proceeding corresponds to starting with a two-site simulation until convergence and then a final single-site sweep which can slightly improve the results.

6.4 Time-evolving block decimation

In addition to the iterative ground state search of DMRG, the MPS formalism further allows for an efficient implementation of the time evolution $|\psi(t)\rangle = e^{-i\hat{H}t}\,|\psi(t=0)\rangle$ governed by a short-ranged Hamiltonian $\hat{H}$. Furthermore, we can use imaginary-time evolution to find the ground state,

$$|\psi_0\rangle = \lim_{\tau\to\infty} \frac{e^{-\hat{H}\tau}\,|\psi_i\rangle}{\left\| e^{-\hat{H}\tau}\,|\psi_i\rangle \right\|}. \tag{6.31}$$

This latter algorithm does, however, not converge as fast as DMRG.

Time-evolving block decimation (TEBD) is based on two concepts we have already encountered before in the context of single-particle and brute-force time evolution: the split-operator method and Trotter-Suzuki decomposition. Recall that we assume that our Hamiltonian can be split into non-commuting parts,

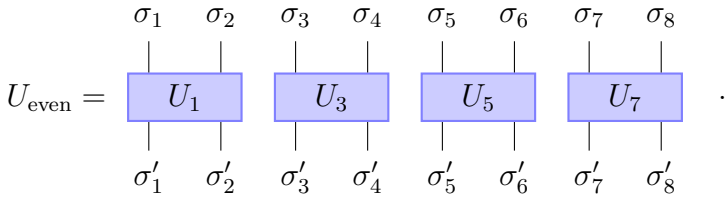
$$\hat{H} = \sum_{k=1}^{K} \hat{h}_k, \tag{6.32}$$

where, in the following, we make the additional assumption that the terms in $\hat{h}_k$ are short ranged. This in particular holds for the spin models we have discussed so far.

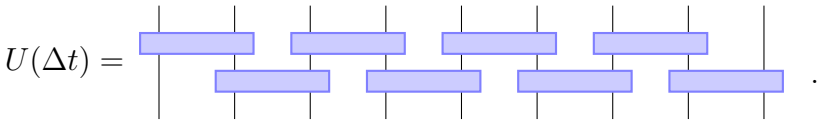
For the XXZ model, we have seen that we can split the Hamiltonian into an even and an odd part, see Eq. (5.27). Each of these two then consists of commuting two-site terms $\hat{H}_i$ [(5.28)], such that the time-evolution operator for the odd part, as an example, takes the form

$$U_{\text{odd}}(\Delta t) = \prod_{i \in \mathcal{O}} U_i(\Delta t), \tag{6.33}$$

where the product runs over odd bond indices i and the individual time-evolution operators are given by Eq. (5.29). In tensor-network notation, this corresponds to



Note that this is *not* an MPO of the indiviual U's, but rather they are applied individually, in other words, we are applying two-site *gates* to the MPS. The total time-evolution operator then can be written as



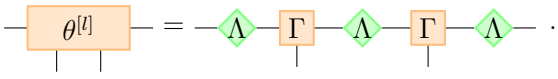
We could now in principle apply all the individual gates to the MPS and decouple the resulting tensor into its MPS form including truncation of the bond dimension—if we would like to work at fixed bond dimension. In time-evolving block decimation, we apply the individual gates one-by-one, each time decoupling the resulting two-site tensor using SVD and truncating to the desired bond dimension. The individual gate operations can be performed very efficiently within the MPS formalism and importantly, the unitary evolution operators preserve the canonical form. In the following, we will again explicitly go through the algorithm.

TEBD algorithm

Time-evolving block decimation is an iterative algorithm, much like DMRG, where we start from the left side and work through the chain to the right side. We will use in the following Vidal's notation, which is well suited for TEBD. Given all the two-site terms $\hat{H}_l$ of the Hamiltonian, we first construct the two-site time-evolution gates. We can store these gates in a list in the order we want to apply them for the time evolution. As we are using the split-operator method, we alternatively apply all the gates on odd bonds and all the gates on even bonds. Like in the case discussed in Sec. 5.2, we can use different orders of the Trotter-Suzuki decomposition and apply the same gates several times in one time step.

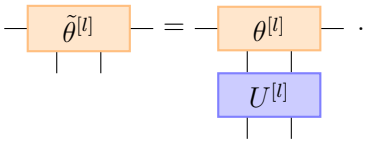
We then iterate through the list of gates and:

1. Calculate the two-site tensor $\theta^{[l]}$ around bond l by contracting the diagram

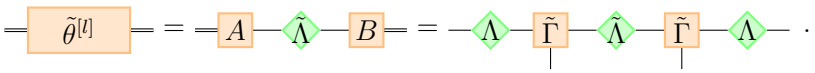


In particular, we contract here the tensors $\Lambda^{[l]}$, $\Gamma^{[l]}$, $\Lambda^{[l+1]}$, $\Gamma^{[l+1]}$, and $\Lambda^{[l+2]}$.

2. Apply the gate $U^{[l]}$ at bond l to find the new two-site tensor $\tilde{\theta}^{[l]}$



3. Reshape the two-site tensor $\tilde{\theta}^{[l]}$ and perform an SVD to split it into single-site tensors



In the last step, we have reshaped the matrices into tensors and separated out the Λ matrices by multiplying A (B) from the left (right) with Λ^{-1} to recover the canonical form.

4. Finally, to avoid a growth of the bond dimension to $d \cdot \chi_{\max}$, we discard the smallest Schmidt values in $\Lambda^{[l+1]}$ keeping only the $\chi_{\max}$ largest ones. In order for our state to stay normalized, we renormalize $\Lambda^{[l+1]}$ such that $\sum_i \Lambda_{ii}^2 = 1$ after truncation. We have thus recovered an MPS state in canonical form with desired bond dimension.

Computational errors in TEBD

The TEBD algorithm has several sources of errors. In particular:

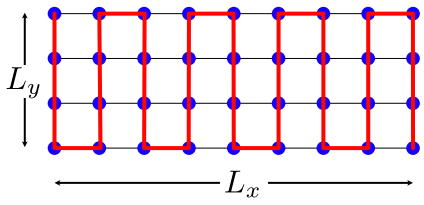
- Truncation error: this error is usually the main error when performing real-time evolution as the required bond dimension for exact time evolution grows exponentially.
- Trotter error: relatively harmless and can be reduced by reducing Δt or performing a higher-order expansion.
- To recover the canonical form, we multiply both A and B in step 3 with Λ^{-1}. For small singular values, this operation is not stable. However, this sort of instability can be avoided with little extra computational cost [5].
- If we perform imaginary-time evolution, we need to keep in mind that the canonical form is only retained when $\Delta\tau \to 0$.

[5] M. Hastings, J. Math. Phys. **50**, 095207 (2009)

6.5 Further topics

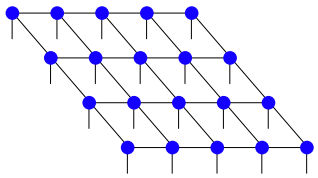
6.5.1 Two-dimensional systems

So far, our discussion of tensor networks has focused exclusively on one-dimensional systems. Indeed, matrix product states are by construction one dimensional tensor networks. Here, the MPS formalism allows for efficient simulations of ground states, even in the case of fermions, where sign-problem-free calculations can be performed. But can we move beyond one dimension? A common tactic is to consider strip geometries or cylinders and unwrap the transverse direction:



While this allows in principle to write the system as a one-dimensional chain, we have introduced long-range interaction into our Hamiltonian. Furthermore, the entanglement expected from the area law now scales as L_y and thus, the bond dimension required is exponential, $\chi \sim \exp(L_y)$. Nevertheless, systems of width up to $L_y \sim 12$ can be simulated in this way.

An alternative is to study the natural extension of the MPS, a two-dimensional tensor network:



Such two-dimensional arrays of tensors are known as a projected entangled pair states (PEPS). While these states satisfy the area law of entanglement in two dimension like MPS in one dimension, they are computationally much harder to handle. An example is the contraction (scalar product) of two PEPS, which is very inefficient. Furthermore, there is no canonical form for PEPS: Having loops in the tensor network means we can not split the network into left and right pieces for a Schmidt decomposition.

6.5.2 Mixed states and open-quantum-systems dynamics

A system that is not in a pure state can be described with the help of the density matrix (operator) ρ, whose unitary time evolution is governed by the system Hamiltonian $\hat{H}$,

$$\partial_t \hat{\rho}\,(t) = -i[\hat{H}, \hat{\rho}\,(t)]\,, \tag{6.34}$$

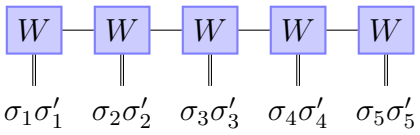
see Sec. 2.1.4, where we set $\hbar = 1$. If the system we are describing is itself a subsystem of a larger system, in other words it is coupled to an environment or bath, we speak of an *open quantum system*. If the environments dynamics is much faster than the characteristic time scale of the system and there is no back action

onto the system, the environment can be considered to have no memory, meaning it acts locally in time. The resulting time evolution is then *Markovian* and is described by a *Lindblad equation*

$$\partial_t \hat{\rho} = \mathcal{L}[\hat{\rho}] = -i[\hat{H}, \hat{\rho}] + \sum_i \gamma_i \Big(\hat{L}_i \hat{\rho} \hat{L}_i^\dagger - \frac{1}{2} \{ \hat{L}_i^\dagger \hat{L}_i, \hat{\rho} \} \Big), \tag{6.35}$$

where the first term again governs the unitary time evolution, while the second term describes the coupling of the system to the environment after tracing out the bath degrees of freedom. $\hat{L}_i$ are called jump operators, the system operators directly coupled to the bath. Examples include creation or annihilation operators if the system can exchange particles with the bath, or a number operator for dephasing.

One way of treating the time evolution in Eq. (6.35) is to recall how we matrix-product decoupled general operators: By rearranging the legs of the $d^N \times d^N$ operator matrix, we arrived at a *superstate* which now lives in a $d^N \cdot d^N$-dimensional space and can be written as

$|\rho\rangle_\sharp =$ 

.

The Lindbladian $\mathcal{L}$ is then a linear *superoperator* $\hat{\mathcal{L}}_\sharp$ acting on $|\rho\rangle_\sharp$, such that we can formally integrate the time evolution as before, $|\rho(t)\rangle_\sharp = \exp\Big(-i\hat{\mathcal{L}}_\sharp t\Big)\,|\rho_0\rangle_\sharp$. To simulate the dynamics, we can again resolve to TEBD.

6.5.3 Symmetries

As we have seen in our discussion of brute-force methods, exploiting symmetries of the Hamiltonian allows for (sometimes massive) computational speed ups. Furthermore, implementing symmetries guarantees the conservation of the corresponding quantities irrespective of numerical errors. This is indeed also the case within the MPS formalism. A condition is that the symmetries are built from locally measurable quantities to accommodate the local construction of an MPS wavefunction. In

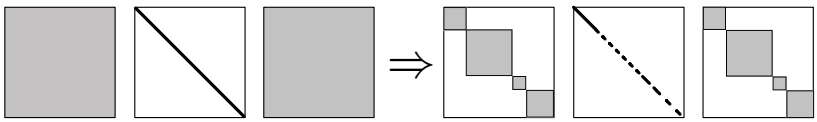


Figure 7: Employing symmetries for the Schmidt decomposition: Instead of ordering the Schmidt values according to $\lambda_0 \geq \lambda_1 \cdots$, we can force the Schmidt eigenstates to belong to a fixed magnetization sector.

particular, we can ensure a block structure of each individual tensor of the MPS: For a state with definite total magnetization, we can force Schmidt eigenstates to correspond to fixed magnetization, too and group the Schmidt values accordingly, see Fig. 7. We can then perform a full state decomposition, always grouping Schmidt values that corresponds to a fixed magnetization to the left and to the right of the bond. This procedure guarantees the block structure of the MPS tensors, which, given a Hamiltonian that preserves magnetization, can be further conserved during

optimization or time evolution. In addition to the just mentioned U(1) symmetries of magnetization or, equivalently, charge conservation, a symmetry that is regularly used is SU(2) symmetry.

6.5.4 Time-dependent variational principle

Another time-evolution algorithm for MPS is based on the time-dependent variational principle (TDVP) in quantum mechanics. Introducing the action functional

$$S = \int_{t_1}^{t_2} dt L[\psi(t), \bar{\psi}(t)] \tag{6.36}$$

with

$$L[\psi(t), \bar{\psi}(t)] = \langle \psi(t)| \, i\partial_t - \hat{H} \, |\psi(t)\rangle \,, \tag{6.37}$$

yields upon extremizing the time-dependent Schrödinger equation. This variational principle can be used to simulate the time evolution of a state also in a subspace of the full Hilbert space, for our purposes, the space of fixed-bond-dimension matrix product states [6].

The TDVP algorithm for MPS can be implemented almost exactly analogue to the DMRG algorithm. As such, it is very useful for simulations with long-ranged Hamiltonians given in MPO form, for which TEBD is not available. Another useful property of the TDVP algorithm is that by construction, it conserves both the norm and the energy of the state under real-time evolution.

6.5.5 Packages

There are excellent libraries for simulating quantum systems with tensor network, so there is no need to implement everything yourself. Examples are iTensor, a C++ and Julia library, or TenPy, a python library.

[6] J. Haegeman, *et al.*, Phys. Rev. B **94**, 165116 (2016)

7 Monte Carlo computations for spin systems

In this and the following chapter, we continue the numerical study of quantum many-body systems with a set of techniques subsumed under the term "Monte Carlo", which is a reference to the famous casino in Monaco and was picked as a code in the nuclear weapons research in the Los Alamos program. The term broadly refers to numerical algorithms with a random (sampling) component for evaluating large sums or multi-dimensional integrals. As such, the techniques have widespread applications in science and engineering. In the context of this lecture, we are mostly interested in applications to quantum many-body systems. More specifically, there are Monte-Carlo techniques that aim to *compute amplitudes or observables of a system using the partition function* and so-called variational Monte-Carlo techniques that try to *optimize*, for instance, a parametrized wave function to find a good approximation to a ground state.

We will start with the former type of problem. Naturally, we are mostly interested in quantum many-body systems, which can be broadly classified into spin systems on a lattice (this chapter) and systems of particles in continuous space (next chapter).

While our primary interest is the hard problem of *quantum* spin systems, there is an equivalence between the computation for a d-dimensional quantum spin system and a $(d+1)$-dimensional classical spin system. Therefore, and because they are of interest in their own right, we will start with a discussion of Monte Carlo techniques for *classical* spin systems, using the example of the Ising model. Note that a more detailed discussion of classical Monte Carlo methods is part of the lecture 'Computational Statistical Physics'.

7.1 Monte Carlo essentials

Monte Carlo is a powerful stochastic technique to estimate (ratios of) high-dimensional integrals. Examples include

- calculating an area, such as that of a circle (see Fig. 8).
- calculating the thermal average of a classical observable A at inverse temperature $\beta = 1/k_{\mathrm{B}}T$. In statistical physics, this average is given by

$$\langle A \rangle_\beta = \frac{1}{Z} \sum_{\text{states } X} A(X) \exp(-\beta E[X]) = \sum_X A(X) W_\beta[X] \tag{7.1}$$

  with the partition function

$$Z = \sum_X \exp(-\beta E[X]) \tag{7.2}$$

  and $E[X]$ denotes the energy of the state X.

Note that we usually separate out a probability distribution and interpret the integral as an expectation value of a function under that distribution. This is explicitly

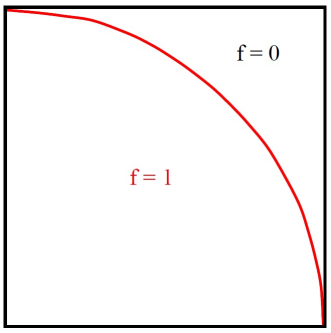


Figure 8: Elementary Monte Carlo experiment: draw two random numbers x, y uniformly distributed in $[0, 1]$. If $x^2 + y^2 < 1$, assign the function $f = 1$, otherwise $f = 0$. Take the average value of the function $\langle f \rangle$. It will converge to $\pi/4$ upon increasing the number of samples. This is the most naive implementation, much more advanced, and hence better converging options exist.

the case in the second example, Eq. (7.1). The sum is then evaluated by sampling the given probability distribution.

The convergence of the algorithm is governed by the central limit theorem: When drawing a random variable from a distribution with mean $\bar{x}$ then after a finite number of draws, the samples' mean will in general deviate by some number δx from the mean of the distribution,

$$\frac{1}{N} \sum_{i=1}^{N} x_i = \bar{x} + \delta x, \tag{7.3}$$

in other words, the estimate is noisy. If the sampling is unbiased, the expected value of the deviation is zero, $\langle \delta x \rangle = 0$, that is, as N tends to infinity, δx goes to 0 as $\delta x \sim 1/\sqrt{N}$. More precisely, $\delta x \sim \sqrt{\langle (x - \bar{x})^2 \rangle / N}$. Variance reduction techniques can further reduce the prefactor in the $\frac{1}{\sqrt{N}}$ error.

Most commonly, we perform sampling in a Monte Carlo simulation using a Markov chain: A Markov chain—or Markov process in discrete time—is a sequence in which each element is picked probabilistically and the probability (distribution) for picking the next element depends only on the current element (not any earlier ones). In physics, we think of the elements as states of a system, see Eq. (7.1). At each step, the system may change from state X to state Y with fixed probability P_{XY}. The matrix elements of the so-called Markov matrix P satisfy

$$P_{XY} \geq 0, \tag{7.4}$$

$$\sum_{Y} P_{XY} = 1. \tag{7.5}$$

These two properties ensure that what determines the move from state X to Y is actually a probability distribution. If we denote the probability distribution of states by a vector w, in other words $w[X]$ describes the probability for the system to be in the state X, then each step corresponds to multiplying w by P^{T} (T denoting transposition). Equation (7.5) implies that P has a right eigenvector with eigenvalue 1 ($w[X] = 1 \ \forall X$). Under the assumption of ergodicity, a Markov matrix has a *unique* left eigenvector with eigenvalue 1, in other words a *stationary distribution*

W. All other eigenvalues are < 1, ensuring (for finite state space) exponential convergence to this distribution W [7]. A Markov chain is called *ergodic* if there is a number N, such that any state can be reached from any other state in any number of steps less than or equal to N.

How do we design a Markov matrix with the desired stationary distribution W? One can check explicitly that a sufficient condition is that of *detailed balance*, which has to be met by the matrix elements

$$\frac{P_{XY}}{P_{YX}} = \frac{W[Y]}{W[X]}. \tag{7.6}$$

A simple algorithm which satisfies detailed balance is the *Metropolis algorithm*:

- **Step 1.** Starting from state X, *propose* a new candidate state Y with probability $T(X \to Y)$. T should satisfy the following properties:
  1. **Ergodicity.** Any state Y can be reached after a finite number of steps.
  2. **Normalization.** $\sum_Y T(X \to Y) = 1$.
  3. **Reversibility.** $T(X \to Y) = T(Y \to X)$. (This condition is not strictly required, but it simplifies the conditions below.)
- **Step 2.** *Accept* the candidate state Y as the next state in the Markov chain with probability

$$A(X \to Y) = \min\left(1, \frac{W[Y]}{W[X]}\right). \tag{7.7}$$

That is, if the proposed state is more likely in the stationary distribution it is always accepted [8]. The total probability of a Markov chain is then

$$P_{XY} = T(X \to Y)A(X \to Y), \tag{7.9}$$

satisfying ergodicity, as well as Eqs. (7.4) and (7.5). Finally, if the new state Y is rejected, we set X again as the next state in the Markov chain.

7.2 Classical Ising model

The classical Ising model describes a collection of degrees of freedom σ_i that can be in one of two states,‘ $\sigma_i = \pm 1$. We call them spins here, but the model is very powerful in describing many statistical processes. The (classical) Hamiltonian of the model is given by

$$H[\{\sigma_i\}] = -\sum_{i,j} J_{ij}\sigma_i\sigma_j - \sum_i h_i\sigma_i = E[C], \tag{7.10}$$

[7] These powerful properties are ensured by the Perron-Frobenius theorem.

[8] As an aside, if T is not satisfying reversibility, the acceptance probability is given by

$$A(X \to Y) = \min\left(1, \frac{W[Y]T(Y \to X)}{W[X]T(X \to Y)}\right) \tag{7.8}$$

in order to ensure detailed balance.

where J_{ij} are the coupling constants and h_i is an external field. Very often, J_{ij} is a sparse matrix, for instance in the case of a lattice model with non-zero interactions only for nearest neighbors, which we denote by $\langle i, j \rangle$. Here and in the following, we denote *configurations* of spins by $C = \{\sigma_i\}$, which is nothing but a vector containing $+1$ or -1 entries at element i depending on the state of the spin i [9]. Finally, in classical mechanics, the Hamiltonian corresponds to the energy $E[C]$ of Eq. (7.1) for the configuration C.

The interest in the Ising model arises because it can describe a symmetry-breaking phase transition at a finite temperature (depending on the dimensionality and couplings J_{ij}). If $J_{ij} \geq 0$, the symmetry-broken state is ferromagnetic and consequently, all non-ferromagnetic spin configurations are suppressed at low temperature. At infinite temperature, the paramagnetic state is an equal mixture of all spin configurations. As the temperature is lowered to near the phase transition, however, large ferromagnetic domains form in an overall inhomogeneous state. Consequently, many states still contribute to the partition sum in this case. Finally, at very low temperatures, the partition sum is dominated by two states, the two ferromagnetic states with all spins up ($\sigma_i = 1$) and all spins down ($\sigma_i = -1$).

We can use Monte Carlo techniques to evaluate the sum over configurations in Eqs. (7.1) and (7.2). A naive approach, analogous to the computation of π in the introductory example above, would be to randomly pick configurations C and add $A(C)$ to the sum. While this approach, a form of so-called direct sampling, would work at high temperatures, where all configuarations are equally probable, it would be very inefficient at low temperatures with much fewer states contributing. Instead, we want to use Markov-chain Monte Carlo and propose the next configuration to be evaluated for the sum based on the previous one. The central question then is how to generate this sequence of configurations.

A straight-forward Markov-chain approach is to propose flipping a single spin in the configuration. Roughly speaking, the larger the contribution of this new configuration to the summand, in other words, the smaller the energy of the new state relative to the previous one, the more likely should it be that we accept the new configuration [10]. However, local single-spin flips are very inefficient for moving between configurations with large domains. Rather, entire clusters of spins should be flipped. At very low temperatures, it becomes almost impossible to move between the two ferromagnetic states using single spin flips.

To overcome these shortcomings of the single spin update algorithm, we will now study algorithms that work with such "cluster updates". In the following, we discuss two algorithms using such cluster updates, namely the Swendsen-Wang algorithm and the Wolff algorithm.

7.2.1 Cluster updates for the Ising model

We consider in the following the ferromagnetic Ising model with the same coupling constant $J > 0$ for all bonds and no field $h_i = 0$, $\forall i$. We will build clusters such that spins pointing in the same direction can be in the same cluster, while spins pointing in the opposite directions will never be in the same cluster (see Fig. 9). Importantly, both algoritms satisfy detailed balance.

[9] The configurations C thus correspond to the states X of the general discussion above.
[10] We will study this single-spin-flip algorithm in the exercises.

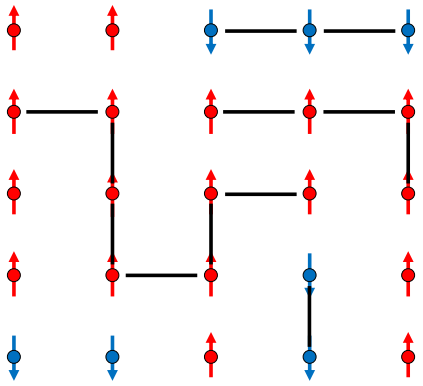


Figure 9: Ising model. Configuration C is a set of spins on sites. For the Swendsen-Wang algorithm, all spins are either connected or not, forming clusters—four in the above example. For the Wolff algorithm, only one cluster is built.

Swendsen-Wang algorithm

The Swendsen-Wang algorithm comprises the following steps:

1. Any two neighboring spins, which are aligned, are connected with probability $p = 1 - e^{-2\beta J}$. This choice builds large cluster, while providing a simple acceptance probability. Spins pointing in the opposite direction are never connected.

2. A cluster labeling algorithm (e.g., Hoshen-Kopelman algorithm) is used to find clusters of connected spins.

3. Measurements are performed using improved estimators, see below.

4. Each cluster is flipped with probability 1/2.

Wolff algorithm

The Swendsen-Wang algorithm is inefficient in dimensions higher than two, because the procedure results in creating many small clusters instead of the desired large ones. The Wolff algorithm is similar to the Swendsen-Wang algorithm, but builds only *one* cluster from a randomly chosen point. As the probability of this point being on a cluster of size S is proportional to S, it builds preferably larger clusters to be flipped. This results in more efficient non-local moves on the energy landscape. It proceeds as follows:

1. Choose a random site.

2. If a neighboring spin is parallel, add to the cluster with $p = 1 - e^{-2\beta J}$. This choice again provides a large cluster and furthermore, the Metropolis acceptance rate is 1.

3. Repeat the second step for all points newly added to the cluster, thus growing it, until no more spins can be added.

4. Measurements are performed using improved estimators, see below.

5. Flip all spins in the cluster.

We will see below that the linear cluster size diverges with the correlation length ξ and that the average number of spins in a cluster scales $\propto \chi = \langle m^2 \rangle - \langle m \rangle^2$. In other words, the closer we are to a phase transition, where $\chi \to \infty$, the more spins we flip per step, thereby addressing the problem of critical slowing down. Close to criticality, the cluster update algorithms can be many orders of magnitude more efficient than the local update methods that flip individual spins.

7.2.2 Improved estimators

Here, we present a neat trick to do "smart measurements" in a cluster algorithm that reduces the statistical error of the Markov chain using *improved estimators*. We illustrate them (mostly) on the Swendsen-Wang algorithm. Remember that this algorithm divides the lattice into N_C clusters, where all spins in each cluster are aligned. The next possible C' is any of the 2^{N_C} configurations arising from flipping (or not) each of the N_C clusters. The idea of improved estimators is to measure expectation values in *all* of these 2^{N_C} *equally probable* configurations.

Let us start with the magnetization $\langle m \rangle_{C'} = 1/N \sum_i \langle \sigma_i \rangle_{C'}$, where the expectation value is over all possible next configuartions C'. If we consider the expectation value of a single spin $\langle \sigma_i \rangle$, we find

$$\langle \sigma_i \rangle_{C'} = \frac{1}{2^{N_C}} \sum_{C'} \sigma_i = \frac{1}{2} \sum_{a_i = \pm 1} \sigma_i = \frac{1}{2}(1-1) = 0, \tag{7.11}$$

where we first perform the sum over all clusters that do not contain site i and a_i labels the cluster containing i, which can be all up or all down. While this result is obvious from the symmetries of the model, the principle underlying this improvement also works in less trivial situations. We can straight-forwardly compute two-point correlation functions to find

$$\langle \sigma_i \sigma_j \rangle_{C'} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are in the same cluster,} \\ 0 & \text{otherwise.} \end{cases} \tag{7.12}$$

Using this simple result for the correlation function, the mean square of the magnetization is

$$\langle m^2 \rangle_{C'} = \frac{1}{N^2} \sum_{i,j} \langle \sigma_i \sigma_j \rangle_{C'} = \frac{1}{N^2} \sum_a \sum_{i \in a} \sum_{j \in a} 1 = \frac{1}{N^2} \sum_a S(a)^2, \tag{7.13}$$

where a labels the N_C clusters in a configuration and $S(a)$ is the number of spins in the cluster. In the Wolff algorithm only a single cluster is built. The above sum can be rewritten to yield

$$\langle m^2 \rangle_{C'} = \frac{1}{N^2} \sum_i \sum_{j \in a_i} = \frac{1}{N} S(a_0) \tag{7.14}$$

with a_0 the one cluster built in the Wolff algorithm. Note that in all these calculations, we perform an additional average over the configurations C in the Markov chain.

7.3 Thermodynamics of quantum spin systems

We now turn to the Monte Carlo methods for *quantum* spin systems. On the level of statistical physics, the main change is that given a set of degrees of freedom, the partition sum Z is not just a sum over their possible configurations, but a trace over an operator on the Hilbert space. Specifically

$$Z = \mathrm{Tr}\left(e^{-\beta \hat{H}}\right). \tag{7.15}$$

While the dimension of the Hilbert space is equal to the number of classical spin configurations, the added complexity comes from the fact that we cannot straightforwardly exponentiate (or even store) $\hat{H}$, since it is an exponentially large matrix [11]. Furthermore, the expression for an observable, like the magnetization $\hat{m}$,

$$\langle \hat{m} \rangle = \frac{1}{Z} \mathrm{Tr}\left(\hat{m} e^{-\beta \hat{H}}\right) \tag{7.16}$$

cannot be evaluated by sampling over configurations, as $\hat{m}$ might induce transitions between different quantum states: $\hat{m}$ is also a matrix. If we knew an eigenbasis of $\hat{H}$ (which we cannot hope for in general) with $\hat{H}\,|\lambda\rangle = E_\lambda\,|\lambda\rangle$, then

$$Z = \mathrm{Tr}\left(e^{-\beta \hat{H}}\right) = \sum_\lambda \langle \lambda | e^{-\beta \hat{H}} | \lambda \rangle = \sum_\lambda e^{-E_\lambda \beta} \tag{7.17}$$

would be straightforward to compute. However, this basis is in general not simultaneously an eigenbasis of $\hat{m}$, but rather $\hat{m}\,|\lambda\rangle = \sum_k a_k^\lambda\,|\lambda_k\rangle$, such that computing expectation values retains the same level of complexity.

The way forward with this complexity is a mapping of the quantum system to an equivalent classical one, the first step in any quantum Monte Carlo procedure. Once such a mapping is established, we can rely on the powerful classical Monte Carlo algorithms that we discussed already. Formally, we want to reexpress

$$Z = \mathrm{Tr}\left(e^{-\beta \hat{H}}\right) = \sum_C W(C), \tag{7.18}$$

$$\langle \hat{m} \rangle = \frac{1}{Z} \mathrm{Tr}\left(\hat{m} e^{-\beta \hat{H}}\right) = \frac{1}{Z} \sum_C m(C) W(C), \tag{7.19}$$

where the sums run over configurations C of an artificial classical system. We will see such a mapping for particles in free space in the framework of path-integral quantum Monte Carlo in the next chapter. As an introduction to applications to lattice spin problems, we discuss a single quantum-mechanical spin-1/2 first.

7.3.1 Spin-1/2 in a magnetic field

The Hamiltonian for a single quantum-mechanical spin-1/2 in a longitudinal field h and a transverse field Γ is given by

$$\hat{H} = \hat{H}_z + \hat{H}_x = -h\hat{S}^z - \Gamma \hat{S}^x. \tag{7.20}$$

[11] In the classical problem, that matrix is diagonal and hence exponentiation is straight-forward by taking the exponential of each diagonal element.

Using a representation in which the z component of the spin operator $\hat{S}^z$ is diagonal, $\{|\uparrow\rangle, |\downarrow\rangle\}$, the spin operators $\hat{S}^x$ and $\hat{S}^z$ are represented by the Pauli matrices

$$\begin{aligned} \hat{S}^x &= \frac{\hbar}{2}\sigma^x, \\ \hat{S}^z &= \frac{\hbar}{2}\sigma^z, \end{aligned} \tag{7.21}$$

see Sec. 2.1.2. For simplicity, we set $\hbar = 1$, such that the Hamiltonian reads

$$\hat{H} = \begin{pmatrix} -h/2 & -\Gamma/2 \\ -\Gamma/2 & h/2 \end{pmatrix}. \tag{7.22}$$

7.3.2 Discrete-time path integral

We now introduce a path-integral representation of the partition function. Mathematically, it relies on the rather elementary trick of rewriting $\beta = \Delta\tau M$, where M is an integer, which we will refer to as the number of imaginary time steps, and $\Delta\tau$ is the width of such a step. For each imaginary time step, we approximate $e^{-\Delta\tau\hat{H}}$ by the *transfer matrix* $\hat{U}$,

$$e^{-\Delta\tau\hat{H}} = \hat{U} + O[(\Delta\tau)^2], \tag{7.23}$$

where

$$\hat{U} = \mathbb{1} - \Delta\tau\hat{H} = \begin{pmatrix} 1 + \frac{\Delta\tau h}{2} & \frac{\Delta\tau\Gamma}{2} \\ \frac{\Delta\tau\Gamma}{2} & 1 - \frac{\Delta\tau h}{2} \end{pmatrix}. \tag{7.24}$$

Note that we just use the lowest-order Taylor expansion (versus a more sophisticated Trotter-Suzuki scheme). We anticipate taking the limit of large M and small $\Delta\tau$ in such a way that the inverse temperature β is held constant.

In the discrete-time path-integral formulation, we evaluate the operator exponential $e^{-\beta\hat{H}}$ by writing the trace as a sum over a basis of states $|i\rangle$ of the Hilbert space—in our case, the Hilbert space is two-dimensional with $|i\rangle \in \{|\uparrow\rangle, |\downarrow\rangle\}$.) Inserting $M-1$ resolutions of the identity $\mathbb{1} = \sum_i |i\rangle\langle i|$, we obtain

$$\begin{aligned} Z = \mathrm{Tr}\left(e^{-\beta\hat{H}}\right) &= \mathrm{Tr}\left[(e^{-\Delta\tau\hat{H}})^M\right] = \mathrm{Tr}\left\{\left[\hat{U} + O[(\Delta\tau)^2]\right]^M\right\} \\ &= \sum_{i_1}\sum_{i_2,\ldots,i_M} \langle i_1|\hat{U}|i_2\rangle\ \langle i_2|\hat{U}|i_3\rangle\langle i_3|\ldots|i_M\rangle\ \langle i_M|\hat{U}|i_1\rangle + O(\Delta\tau) \\ &\approx \sum_{C=\{i_1,\ldots,i_M\}} W(C), \end{aligned} \tag{7.25}$$

where the approximation of the last line is from dropping the terms $O(\Delta\tau)$. The key observation is that this expression is identical to the partition sum of a 1D chain of classical Ising spins $\sigma_i = \pm 1$ of length M at inverse temperature β_{cl},

$$H_{\mathrm{cl}} = -J_{\mathrm{cl}}\sum_{i=1}^{M}\sigma_i\sigma_{i+1} - h_{\mathrm{cl}}\sum_i \sigma_i + E_0 \tag{7.26}$$

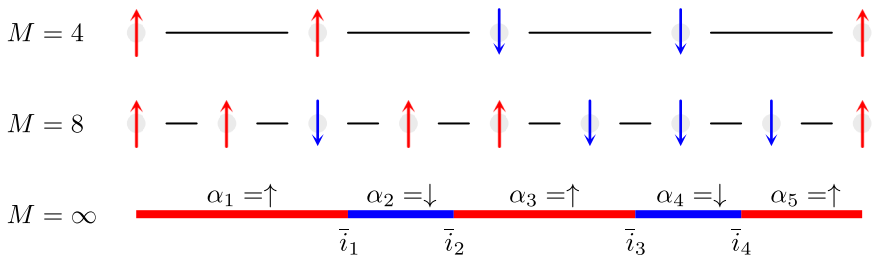


Figure 10: 0D transverse field Ising model, equivalent to a 1D classical Ising model. Taking $M \to \infty$ brings us to the continuous limit, where the number of domain walls stays finite. We label the locations of domain walls by $\bar{i}_k$, $k = 1, 2, 3, \ldots$ and the value of each domain by $\alpha_k = \pm 1$, $k = 1, 2, 3, \ldots$

with periodic boundary conditions $\sigma_{M+1} = \sigma_1$ and the following relations:

$$\beta_{\mathrm{cl}} J_{\mathrm{cl}} = -\frac{1}{2} \log(\Delta\tau\Gamma/2), \tag{7.27}$$

$$\beta_{\mathrm{cl}} h_{\mathrm{cl}} = \log(1 + \Delta\tau h/2) \approx \Delta\tau h/2, \tag{7.28}$$

$$\beta_{\mathrm{cl}} E_0 = M \beta_{\mathrm{cl}} J_{\mathrm{cl}}, \tag{7.29}$$

provided we work in the range of validity of the second relation's approximation. Therefore, any Monte Carlo algorithm for the classical Ising model, including cluster updates, can be used for the simulation. Finally, we can calculate expectation values as

$$\langle \hat{A} \rangle = \frac{1}{Z} \sum_C \frac{\langle i_1 | \hat{A}\hat{U} | i_2 \rangle}{\langle i_1 | \hat{U} | i_2 \rangle} W(C) = \frac{1}{Z} \sum_C A(C) W(C). \tag{7.30}$$

Note that this expression simplifies significantly for A diagonal in the basis $|i\rangle$.

7.3.3 Continuous-time path integral

To reduce the error in Eq. (7.23), we need to take the limit $\Delta\tau \to 0$, or, equivalently, $M \to \infty$. Consequently, we need to store larger and larger spin configurations. Nevertheless, we will see in the following that we can explicitly take the limit $\Delta\tau \to 0$ and work directly in the continuum. The key insight is that while the number of spins diverges, the number of domain walls remains finite. To see this, we first consider the coupling between neighboring spins in the time direction,

$$\beta_{\mathrm{cl}} J_{\mathrm{cl}} = -\frac{1}{2} \log(\Delta\tau\Gamma/2) \to \infty \quad \text{for} \quad \Delta\tau \to 0, \tag{7.31}$$

which diverges. Consequently, the probability for any pair of spins to be antiparallel rather than parallel, given by the Boltzmann factor

$$e^{-2\beta_{\mathrm{cl}} J_{\mathrm{cl}}} = \frac{\Delta\tau\Gamma}{2}, \tag{7.32}$$

is proportional to $\Delta\tau$ and hence vanishes for $M \to \infty$. The average number of domain walls, however, is proportional to

$$M e^{-2\beta_{\mathrm{cl}} J_{\mathrm{cl}}} = M \frac{\Delta\tau\Gamma}{2} = \frac{\beta\Gamma}{2}, \tag{7.33}$$

and remains finite if the $\Delta\tau \to 0$ and $M \to \infty$ limit is taken such that β is held constant. Instead of keeping track of spin configurations, it is thus sufficient to store the times τ, at which the configuration changes.

Specifically, we store the orientation $\alpha_j \in \{+1, -1\}$ of the spins in the j-th domain, $j = 1, \ldots, 2n$ [12], and the location of the last spin of the j-th domain $\bar{i}_j$ (see Fig. 10). In this notation, the weight of the configuration reads

$$\begin{aligned} W(C) &= \langle i_1|\hat{U}|i_2\rangle \, \langle i_2|\hat{U}|i_3\rangle \ldots \langle i_M|\hat{U}|i_1\rangle \\ &= (1+\Delta\tau h\alpha_1/2)^{\bar{i}_1-1} \, \langle \alpha_1|\hat{U}|\alpha_2\rangle \, (1+\Delta\tau h\alpha_2/2)^{\bar{i}_2-\bar{i}_1-1} \times \cdots \times \\ &\quad \times (1+\Delta\tau h\alpha_n/2)^{\bar{i}_{2n}-\bar{i}_{2n-1}-1} \, \langle \alpha_{2n}|\hat{U}|\alpha_1\rangle \, (1+\Delta\tau h\alpha_1/2)^{M-\bar{i}_{2n}}. \end{aligned} \tag{7.34}$$

Further,

$$\langle \alpha_j|\hat{U}|\alpha_{j+1}\rangle = -\Delta\tau \, \langle \alpha_j|\hat{H}_x|\alpha_{j+1}\rangle = \Delta\tau\Gamma/2, \tag{7.35}$$

as $|\alpha_j\rangle \neq |\alpha_{j+1}\rangle$. Finally, we introduce $\tau_j = \bar{i}_j\Delta\tau$, $E_j = -h\alpha_j/2$ and now formally take the limit $\Delta\tau \to \mathrm{d}\tau \to 0$ to find

$$\begin{aligned} W(C) &= e^{-\tau_1 E_1}\frac{\Gamma}{2}e^{-(\tau_2-\tau_1)E_2}\ldots e^{-(\tau_{2n}-\tau_{2n-1})E_{2n}}\frac{\Gamma}{2}e^{-(\beta-\tau_{2n})E_1}\mathrm{d}\tau_1\cdots\mathrm{d}\tau_{2n} \\ &= \exp\left[-(\beta-2\sum_{k=1}^{n}(\tau_{2k}-\tau_{2k-1}))E_1 + \log\left(\prod_{k=1}^{n}\frac{\Gamma}{2}\mathrm{d}\tau_{2k-1}\frac{\Gamma}{2}\mathrm{d}\tau_{2k}\right)\right]. \end{aligned} \tag{7.36}$$

To evaluate the final continuous time representation of the partition function

$$Z = \sum_{\alpha_1=\pm1} e^{-\beta E_1} + \sum_{n=1}^{\infty}\sum_{(\alpha_1,\ldots,\alpha_{2n})}\int_0^\beta \mathrm{d}\tau_1\cdots\int_{\tau_{2n-1}}^\beta \mathrm{d}\tau_{2n}e^{-\tau_1 E_1}\frac{\Gamma}{2}\ldots\frac{\Gamma}{2}e^{-(\beta-\tau_{2n})E_1}, \tag{7.37}$$

we can sample using a Markov chain, where new configurations are proposed by either adding or removing two domain walls

7.3.4 More complicated models: quantum XY model

In the same fashion, any d-dimensional *quantum* spin model can be mapped to a $(d+1)$-dimensional *classical* Ising model. For this purpose, we start again by discretizing $\beta = M\Delta\tau$:

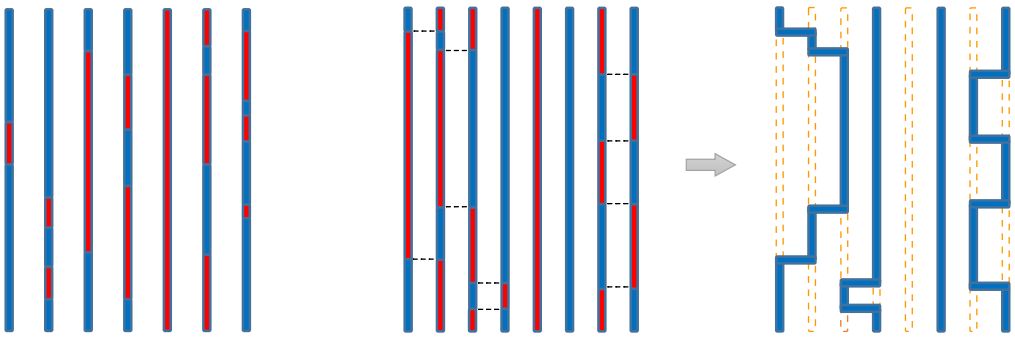
$$e^{-\beta\hat{H}} = \left(e^{-\Delta\tau\hat{H}}\right)^M = (\mathbb{1} - \Delta\tau\hat{H})^M + O(\Delta\tau), \tag{7.38}$$

a rewriting that is again exact in the $\Delta\tau \to 0$, $M \to \infty$ limit. As before, we write, by inserting resolutions of the identity,

$$\begin{aligned} Z &= \mathrm{Tr}\left(e^{-\beta\hat{H}}\right) \\ &= \sum_{i_1,\ldots,i_M} \langle i_1|\mathbb{1} - \Delta\tau\hat{H}|i_2\rangle \, \langle i_2|\mathbb{1} - \Delta\tau\hat{H}|i_3\rangle \ldots \langle i_M|\mathbb{1} - \Delta\tau\hat{H}|i_1\rangle \, . \end{aligned} \tag{7.39}$$

Note that now, the states $|i\rangle$ are the 2^N classical configurations, with N the total number of spins. As before, we could sample this d-dimensional quantum spin-1/2

[12] Due to the periodic boundary conditions of the trace, the number of domains has to be even.



no spin conservation
(1D TFIM)

with spin conservation
(XY model)

Figure 11: Models with and without spin (magnetization) conservation with red and blue denoting spin-up and spin-down domains. If the magnetization is conserved, spin flips are only allowed on neighboring sites and we only have to store information about one of the two types of world lines.

problem over all classical configurations $i_1, \ldots, i_M$. For the transverse field Ising model, the world lines are just "colored" segments of straight lines: one color for $|\uparrow\rangle$ and another for $|\downarrow\rangle$, and we store the location of the domain walls. This is a conceptually easy generalization of the above case, see Fig. 11, where each site has a sequence of configurations independent of that of the other sites.

In models, where the total magnetization is conserved, the lines from neighboring sites become dependent on each other: They have to change color together, as spin flips always happen pairwise (see Fig. 11) [13]. We consider as an example the Hamiltonian of the XY model

$$\hat{H} = -\sum_{\langle i,j \rangle} \frac{J_{xy}}{2} (\hat{S}_i^+ \hat{S}_j^- + \hat{S}_i^- \hat{S}_j^+). \tag{7.40}$$

Looking at the (partial) weights $\langle i_n | \mathbb{1} - \Delta\tau \hat{H} | i_{n+1} \rangle$, we realize that $\mathbb{1}$ conserves all spins, so $|i_n\rangle = |i_{n+1}\rangle$. Moreover, $\Delta\tau\hat{H}$ only mediates matrix elements between states, where $|i_n\rangle$ and $|i_{n+1}\rangle$ differ by one spin flip $\hat{S}_i^+ \hat{S}_j^-$. All classical configurations $i_1, \ldots, i_M$ that do not fulfill this condition have exactly zero weight. Therefore, we only need to consider *connected world lines* representing spins of the same orientation, as shown in Fig. 11.

Finally, in the continuum limit $\Delta\tau \to 0$, instead of storing all spins at all M time slices, we again store only the minimal information of where and in which direction the blue lines bend, in other words, where spin flip-flops occur. As for the simple case discussed above, the probability of $\hat{H}$ acting at a given time is proportional to $\frac{1}{M}$, and hence the total number of spin flip-flops will stay finite under $M \to \infty$ and $\Delta\tau \to 0$ with β constant.

[13] Note that this naturally allows for a cluster-update algorithm, the so-called loop algorithm: H. G. Evertz, et al., PRL **70**, 875 (1993); B. B. Beard and U-J. Wiese, PRL **77**, 5130 (1996); B. Ammon, et al., PRB **58**, 4304 (1998)

7.3.5 The negative sign problem

While one might think that we now have an extremely powerful method to tackle any spin problem, the *negative sign problem* destroys our dreams. In the algorithms discussed here, we need to interpret $\langle i_n|\mathbb{1} - \Delta\tau\hat{H}|i_{n+1}\rangle$ as probabilities—more exactly the product of such terms—which requires them to be positive. However, positive off-diagonal elements of $\hat{H}$ will lead to negative probabilities. The simplest example for this is an antiferromagnetic exchange term $J_{xy} < 0$ in the XXZ model. While for bipartite lattices (such as chains, square lattices, or hexagonal lattices), spin flips come in pairs due to the periodic boundary conditions imposed by the trace, in so-called *frustrated* lattices this is not the case and they remain very hard to simulate. One way to still use Monte Carlo techniques is to sample over $|W(C)|$ and incorporate the sign of the weight into the observable, in other words writing

$$\begin{aligned}\langle\hat{A}\rangle_W &= \frac{\sum_C A(C)W(C)}{\sum_C W(C)} = \frac{\sum_C A(C)\,\mathrm{sign}(W)|W(C)|/\sum_C |W(C)|}{\sum_C \mathrm{sign}(W)|W(C)|/\sum_C |W(C)|} \\ &= \frac{\langle\hat{A}\,\mathrm{sign}\rangle_{|W|}}{\langle\mathrm{sign}\rangle_{|W|}}.\end{aligned} \tag{7.41}$$

Unfortunately, this will not solve the exponential scaling of the algorithms, as for most interesting cases, the denominator goes to zero. Correspondingly, the relative error increases (exponentially) with inverse temperature and system size.

7.4 Variational Monte Carlo

We have seen in the last section, that for frustrated spin systems, the negative sign problem prevents us from a full (quantum) statistical treatment with Monte Carlo methods. If we are interested in the ground-state properties of a system, we can combine Monte Carlo methods with the variational principle to approximate the ground-state wave function of a given Hamiltonian $\hat{H}$. For this purpose, we consider a parametrization of a wave function

$$|\psi(\boldsymbol{\theta})\rangle = \sum_n \psi_n(\boldsymbol{\theta})|n\rangle, \tag{7.42}$$

where $|n\rangle$ is some (orthonormal) basis of the Hilbert space, such as the occupation basis introduced in Sec. 4. Using the variational principle, we want to find parameters $\tilde{\boldsymbol{\theta}}$, which minimize the energy,

$$E_{\tilde{\boldsymbol{\theta}}} = \min_{\boldsymbol{\theta}}\langle\hat{H}\rangle_{\boldsymbol{\theta}} = \min_{\boldsymbol{\theta}}\frac{\langle\psi(\boldsymbol{\theta})|\hat{H}|\psi(\boldsymbol{\theta})\rangle}{\langle\psi(\boldsymbol{\theta})|\psi(\boldsymbol{\theta})\rangle} \geq E_{\mathrm{g.s.}}. \tag{7.43}$$

In Sec. 6, we have seen that for the special case of matrix product states, we can use the DMRG algorithm to minimize this variational energy iteratively for each site. Here, we will treat the general case, where the Hilbert space dimension is too large for direct evaluation, but we can evaluate the expectation value with Monte Carlo. Inserting a resolution of identity in both the numerator and denominator of the above expectation value, we can write the energy expectation value as

$$E_{\boldsymbol{\theta}} \equiv \frac{\sum_n\langle\psi(\boldsymbol{\theta})|n\rangle\langle n|\hat{H}|\psi(\boldsymbol{\theta})\rangle}{\sum_n\langle\psi(\boldsymbol{\theta})|n\rangle\langle n|\psi(\boldsymbol{\theta})\rangle} = \frac{\sum_n|\psi_n(\boldsymbol{\theta})|^2 E_{\mathrm{l}}(n)}{\sum_n|\psi_n(\boldsymbol{\theta})|^2}, \tag{7.44}$$

where we have introduced the *local energy*

$$E_{\mathrm{l}}(n) = \sum_m H_{nm}\psi_m(\boldsymbol{\theta})/\psi_n(\boldsymbol{\theta}) \tag{7.45}$$

with $H_{nm} = \langle n|\hat{H}|m\rangle$ the matrix elements of the Hamiltonian in the chosen basis. Interpreting $W(n) \equiv |\psi_n(\boldsymbol{\theta})|^2/\sum_{n'}|\psi_{n'}(\boldsymbol{\theta})|^2 \geq 0$ as a weight, we can calculate the expectation value in Eq. (7.44) using a (Markov chain) Monte Carlo method.

7.4.1 Stochastic Gradient Descent

In order to find the variational parameters that minimize the energy in Eq. (7.43), a first attempt would be to employ a gradient descent, in other words we update

$$\boldsymbol{\theta} \to \boldsymbol{\theta} - \lambda \boldsymbol{\nabla}_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}} \tag{7.46}$$

with λ some learning rate. We now want to write the gradient in the above expression in a way that we can again calculate it stochastically within a Monte Carlo approach. For this purpose, we write

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}} = \frac{\sum_n [\psi_n^*(\boldsymbol{\theta})\boldsymbol{\nabla}_{\boldsymbol{\theta}}\psi_n(\boldsymbol{\theta}) + \psi_n(\boldsymbol{\theta})\boldsymbol{\nabla}_{\boldsymbol{\theta}}\psi_n^*(\boldsymbol{\theta})]E_{\mathrm{l}}(n)}{\sum_n |\psi_n(\boldsymbol{\theta})|^2} \tag{7.47}$$

$$- E_{\boldsymbol{\theta}} \frac{\sum_n [\psi_n^*(\boldsymbol{\theta})\boldsymbol{\nabla}_{\boldsymbol{\theta}}\psi_n(\boldsymbol{\theta}) + \psi_n(\boldsymbol{\theta})\boldsymbol{\nabla}_{\boldsymbol{\theta}}\psi_n^*(\boldsymbol{\theta})]}{\sum_n |\psi_n(\boldsymbol{\theta})|^2}. \tag{7.48}$$

Introducing the logarithmic wave-function derivative $\boldsymbol{O}(n) = \boldsymbol{\nabla}_{\boldsymbol{\theta}}\psi_n(\boldsymbol{\theta})/\psi_n(\boldsymbol{\theta})$, we can reexpress the gradient as a sum with weights $W(n)$, namely

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}}\langle E\rangle_{\boldsymbol{\theta}} = \sum_n W(n)[E_{\mathrm{l}}(n)\boldsymbol{O}(n) + \mathrm{h.c.}] - E_{\boldsymbol{\theta}} \sum_n W(n)[\boldsymbol{O}(n) + \mathrm{h.c.}] \tag{7.49}$$

$$= 2\mathrm{Re}\Big\{ \sum_n W(n)[E_{\mathrm{l}}(n) - E_{\boldsymbol{\theta}}]\boldsymbol{O}(n)\Big\}. \tag{7.50}$$

As we estimate the gradient in the above expression with Monte Carlo sampling, this approach is referred to as Stochastic Gradient Descent (SGD).

7.4.2 Stochastic Reconfiguration and the Metric Tensor

The stochastic gradient descent, while commonly used, has some drawbacks: On the one hand, with the variational ansatz wave functions potentially highly non-linear, small changes in some parameters might significantly change the wave function, while changes to others have almost no effect. On the other hand, the derivatives of the wave functions are not orthogonal, resulting in unwanted interference effects. We now discuss a different approach, relying on imaginary time evolution of the variational wave function, which does not suffer from these problems.

Instead of changing the variational parameters in a gradient-descent scheme, we evolve the wave function for a short time step $\delta\tau$,

$$e^{-\delta\tau\hat{H}}|\psi(\boldsymbol{\theta})\rangle \approx (1 - \delta\tau\hat{H})|\psi(\boldsymbol{\theta})\rangle = |\psi(\boldsymbol{\theta}')\rangle \tag{7.51}$$

and ask what small change $\boldsymbol{\theta}' = \boldsymbol{\theta} + \delta\boldsymbol{\theta}$ would best approximate this evolved wave function [14]. For this purpose, we first expand the wave function

$$|\psi(\boldsymbol{\theta} + \delta\boldsymbol{\theta})\rangle \approx (1 + \delta\boldsymbol{\theta} \cdot \hat{\boldsymbol{O}})|\psi(\boldsymbol{\theta})\rangle, \tag{7.52}$$

where we introduced the operator $\hat{\boldsymbol{O}} = \sum_n \boldsymbol{O}(n)|n\rangle\langle n|$. Then, we determine $\delta\boldsymbol{\theta}$ by maximizing the overlap of the two wave-function expansions given in Eqs. (7.51) and (7.52) through maximizing the fidelity, which for two states ϕ and ψ is defined as

$$F(\psi, \phi) = \frac{|\langle\psi|\phi\rangle|^2}{\langle\psi|\psi\rangle\langle\phi|\phi\rangle}. \tag{7.53}$$

Keeping only terms up to quadratic order in $\delta\tau$ and $\delta\boldsymbol{\theta}$, we find

$$F(\delta\boldsymbol{\theta}) = 1 + \delta\tau^2(\langle\hat{H}^2\rangle - E_{\boldsymbol{\theta}}^2) - \delta\tau\delta\boldsymbol{\theta} \cdot \boldsymbol{\nabla}_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}} + \delta\boldsymbol{\theta}^T G \delta\boldsymbol{\theta} \tag{7.54}$$

with the *metric tensor*

$$G_{kl} = \langle\hat{O}_k^* \hat{O}_l\,\rangle_{\boldsymbol{\theta}} - \langle\hat{O}_k^*\rangle_{\boldsymbol{\theta}}\langle\hat{O}_l\,\rangle_{\boldsymbol{\theta}}. \tag{7.55}$$

Finally, maximizing the fidelity with respect to $\delta\boldsymbol{\theta}$, we find the update rule for $\boldsymbol{\theta}$

$$\boldsymbol{\theta} \to \boldsymbol{\theta} - \delta\tau G^{-1} \boldsymbol{\nabla}_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}. \tag{7.56}$$

Comparing this update to the update of the stochastic gradient descent, we see that the gradients are modified with the inverse of the metric tensor, a change that is known to lead to better convergence.

One potential issue concerns estimating the metric tensor with Monte Carlo: Since we have to take the inverse, noisy estimates for small values can lead to instability. One way of mitigating such unstable behavior is to regularize the metric tensor, for example by using

$$G' = \sqrt{\beta^2 \mathbb{I} + G^\dagger G} \tag{7.57}$$

with $\beta \in \mathbb{R}$. For large β, the regularized metric tenser is close to the identity matrix, such that this regularization interpolates between stochastic gradient descent with $\lambda \approx \delta\tau/\beta$ and the stochastic reconfiguration.

Next, we need to introduce variational wave functions that can be used in the above schemes. An important property, which any such variational ansatz needs to provide, is (easy) access to the logarithmic derivative $O(n)$ and the local energies $E_l(n)$. We discuss in the following three different variational wave functions, which are well suited for quantum spin systems: Jastrow states and neural quantum states, which define directly the wave function components, as well as mean-field projected wave functions.

7.4.3 Jastrow States

One of the best established variational ansatz wave functions, which comes in a host of implementations, are so-called Jastrow states. The main physical motivation for

[14] Note that the same approach also works for real time evolution and is equivalent to the time-dependent variational principle

their formulation, first used for continuum models for particles, is to allow for (pairwise) correlations on top of some (non-interacting) reference state. In the context of spin systems, we can define a Jastrow state on the level of the wave function components, namely

$$\psi_n(\boldsymbol{\theta}) = \exp\left(\sum_i a_i\sigma_i + \sum_{i,j} J_{ij}\sigma_i\sigma_j\right) \tag{7.58}$$

with the variational parameters $\boldsymbol{\theta} = \{a_i, J_{ij}\}$. The basis states above are labeled by $n = \{\sigma_1, \cdots, \sigma_N\}$ and N is the number of sites of the system.

7.4.4 Neural Quantum States

The second example we discuss here, neural quantum states (NQS), make use of the recent advances in artificial neural networks. While the ansatz is not physically motivated [15], it relies on and takes advantage of the numerical methods developed for such neural networks. In the following, we give a brief introduction to the general structure of neural networks before commenting on NQS [16].

The structure of (feed-forward) neural networks
Feed-forward neural networks are a class of variational functions $\mathbf{F}(\mathbf{x})$ with a layered structure. The input of the function is a vector of real numbers $\mathbf{x}$ and the output is a vector (in general of different dimension) $\mathbf{y}$. A standard *layer* ℓ is given by a combination of a linear and a nonlinear function, which combine into a vector-valued function

$$f_i^{(\ell)}(\mathbf{x}) = g\left(\sum_{j=1}^{N_{\ell-1}} W_{ij}^{(\ell)} x_j + b_i^{(\ell)}\right), \qquad i = 1, \cdots, N_\ell, \tag{7.59}$$

where $W_{ij}^{(\ell)}$ and $b_i^{(\ell)}$ are the real variational parameters of the network, called *weight matrix* and *bias vector*. The number N_ℓ gives the number of *neurons* in the layer ℓ, which may differ from layer to layer. The matrix $W_{ij}^{(\ell)}$ and the vector b_i are of dimension $N_\ell \times N_{\ell-1}$ and N_ℓ, respectively. The non-linear function g is called the *activation function*. There are several standard choices which can be made for it. For instance, the sigmoid activation function is given by

$$g(x) = \frac{1}{1+e^{-x}}. \tag{7.60}$$

The neural network is then the concatenation of several layer functions. For instance, the structure

$$\mathbf{y} = \mathbf{F}(\mathbf{x}) = \mathbf{f}^{(2)}\left(\mathbf{f}^{(1)}(\mathbf{x})\right), \tag{7.61}$$

is referred to as a neural network with one *hidden layer*, $\mathbf{f}^{(1)}$, while $\mathbf{f}^{(2)}$ is the *output layer*. One also refers to the bare input $\mathbf{x}$ of dimension N_0 as the *input layer* (see Fig. 12).

A network in which the matrix elements $W^{(\ell)}$ can all be nonzero is called *fully connected*. This is in contrast to the situation where one neuron is only connected to the output from a selected subset of neurons in the previous layer. A special

[15] This can sometimes be an advantage, as it has no (physical) bias
[16] For a more detailed introduction, see for example the lecture notes at arXiv:2102.04883

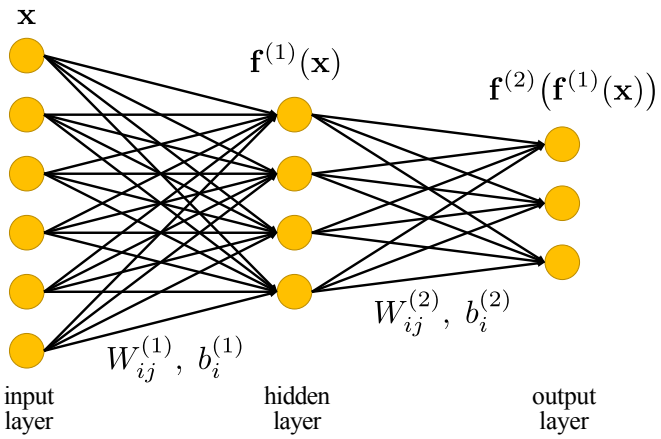


Figure 12: Neural network with one hidden layer. Note that while, in general, the output layer can have many neurons as above, for the NQS we only have a single output, the wave function component $\psi_n(\boldsymbol{\theta}) = F_{\boldsymbol{\theta}}(n)$.

example of such a constrained network is a *convolutional network*, where the weight matrix is a *filter* that is swept over the input. It is particularly useful in image recognition or other situations where only a small local region should go into the information fed into the next layer's neuron.

Neural quantum states
We can now introduce the neural quantum state, which provides another representation of quantum states through a neural network. In particular, the output is a single (complex) number y [17] and the weight matrices W and biases b are now the variational parameters of our wave function, $\boldsymbol{\theta} = \{W, b\}$. As in the above definition of Jastrow states, we use as input to the network the quantum numbers that label a basis of the Hilbert space $|n\rangle$, such as a spin configuration $n = \{\sigma_1, \cdots, \sigma_N\} \equiv \mathbf{x}$, where N is the number of sites of the system. The wave function component is then directly given by

$$\psi_n(\boldsymbol{\theta}) = F_{\boldsymbol{\theta}}(\{\sigma_1, \cdots, \sigma_N\}). \tag{7.62}$$

Note that while we only introduced the (dense) feed-forward neural network above, any network architecture can serve as variational wave function [18]. NetKet is a software package with implementations for neural quantum states.

As a final comment, note that another application of NQS is quantum state tomography, the process of optimizing the wave function given measurements on a real quantum system, such as a quantum computer. The state $|\psi(\boldsymbol{\theta})\rangle$ is required to reproduce the same values of the observables that have been measured. This way, $|\psi(\boldsymbol{\theta})\rangle$ is a likely representation of the (unknown) actual state of the quantum computer.

7.4.5 Mean-field projected wave function

Unlike the Jastrow states and NQS, the definition of the mean-field projected wave function (MFPWF) is based on the representation of spins in terms of pseud-

[17] or two real numbers for the amplitude and phase of the wave function component
[18] Indeed, the original proposal to use neural networks for quantum mechanical simulations used a so-called restricted Boltzmann machine (RBM), see Carleo and Troyer, Science **355**, 602 (2017).

ofermions [19]. Namely

$$\hat{S}_i^\alpha = \frac{1}{2}\sum_{ss'} \hat{c}_{i,s}^\dagger \sigma_{ss'}^\alpha \hat{c}_{i,s'} \tag{7.63}$$

with the fermionic operators $\hat{c}_{i,s}^\dagger$ and $\hat{c}_{i,s}$ and $\alpha = x, y, z$. As an example, the Heisenberg model is given by the Hamiltonian

$$\hat{H} = \sum_{\langle i,j\rangle} J_{ij} \hat{\vec{S}}_i \cdot \hat{\vec{S}}_j \equiv \sum_{\langle i,j\rangle} J_{ij} \sum_{ss'} \left(\hat{c}_{i,s}^\dagger \hat{c}_{j,s} \hat{c}_{j,s'}^\dagger \hat{c}_{i,s'} + \frac{1}{2} \hat{c}_{i,s}^\dagger \hat{c}_{i,s} \hat{c}_{j,s'}^\dagger \hat{c}_{j,s'} \right). \tag{7.64}$$

Importantly, in order to describe the same Hilbert space, we have to restrict the pseudofermions to uniform single occupancy.

We can now introduce the variational wave function in these pseudofermions,

$$|\psi(\boldsymbol{\theta})\rangle = \mathcal{P}_\mathrm{G} \Big[\sum_{i,j} \sum_{s,s'} F_{ij}^{ss'} \hat{c}_{i,s}^\dagger \hat{c}_{j,s'}^\dagger \Big]^{N/2} |0\rangle, \tag{7.65}$$

where $\boldsymbol{\theta} \equiv F_{ij}^{ss'}$ are the $2N \times 2N$ variational parameters, N is the system size and $|0\rangle$ the completely empty state. Further, $\mathcal{P}_\mathrm{G}$ is the so-called Gutzwiller projection operator, enforcing single occupancy, such that the Hilbert spaces of the pseudofermionic model and the spin model are equivalent. Finally, the power $N/2$ ensures the right filling. Note that this variational ansatz also provides a fast access to the wave function components $\psi_n(\boldsymbol{\theta}) = \langle n|\psi(\boldsymbol{\theta})\rangle$. In particular, labeling the basis states by $n = (i_1, s_1; i_2, s_2; \ldots; i_N, s_N)$ and building the matrix $X_{m,n} = F_{i_m i_n}^{s_m s_n} - F_{i_n i_m}^{i_n i_m}$, the components are given by

$$\psi_n(\boldsymbol{\theta}) = (N/2)!\mathrm{Pf}(X) \tag{7.66}$$

with $\mathrm{Pf}(X)$ the Pfaffian of the skew-symmetric matrix X which satisfies $\mathrm{Pf}(X)^2 = \det X$.

Finally, note that, as usually, there are software packages with all necessary functions implemented. An example for MFPWF can be found here:
https://issp-center-dev.github.io/mVMC/docs/index.html

[19] This representation, an example of a parton construction, is known as 'Abrikosov representation' of the spins.

8 Quantum Monte Carlo for particles

This chapter is dedicated to quantum Monte Carlo calculations for systems of interacting particles. In contrast to the spin systems with spatial discretization and continuous time, we will work in discretized imaginary time but in continuous space. In the context of particles, some notions, like a 'path integral' and a 'world line', which we came across already for spins, have a much more natural meaning. We will also learn about the limitations of the method, the most important one being the infamous fermionic sign problem.

8.1 Path integrals in quantum statistical mechanics

Let us introduce the concept of path integrals for the description of quantum many-body systems of particles. We consider the first-quantized and time-independent Hamiltonian of a N-particle system,

$$\hat{H} = \hat{T} + \hat{V}, \tag{8.1}$$

where $\hat{T}$ and $\hat{V}$ are the kinetic and potential terms. In coordinate representation, the two terms are given by

$$\hat{T} = -\frac{\hbar^2}{2m} \sum_i \hat{\boldsymbol{\nabla}}_i^2, \tag{8.2}$$

$$\hat{V} = V(\hat{\vec{r}}^1, \ldots, \hat{\vec{r}}^N). \tag{8.3}$$

In most cases, the potential energy is a sum of single-particle and two-particle terms

$$\hat{V} = \sum_{i=1}^{N} v_{\text{ext}}(\hat{\vec{r}}^i) + \sum_{i<j} v(\hat{\vec{r}}^i - \hat{\vec{r}}^j), \tag{8.4}$$

which may correspond to the ionic potential and the Coulomb-repulsion that electrons experience in a crystal, for instance. In the following, we first assume that the particles are identical but distinguishable, therefore obeying Boltzmann statistics. Later, we will extend our considerations to Bose and Fermi statistics.

All the (statistical) properties of the system in thermal equilibrium are encoded in the thermal density matrix

$$\hat{\rho} = e^{-\beta \hat{H}} \tag{8.5}$$

with $\beta = 1/k_{\text{B}}T$. A path integral representation of the partition function can be obtained from its coordinate representation,

$$\rho(\vec{R}, \vec{R}', \beta) = \langle \vec{R} | e^{-\beta \hat{H}} | \vec{R}' \rangle \,, \tag{8.6}$$

where $\vec{R} = (\vec{r}^1, \ldots, \vec{r}^N)$ is a vector containing all particle coordinates. The partition sum is then given by

$$Z(N, \beta, \Omega) = \text{Tr}\left(e^{-\beta \hat{H}}\right) = \int \rho(\vec{R}, \vec{R}, \beta) \, \mathrm{d}\vec{R} \,, \tag{8.7}$$

where Ω denotes the volume of the system. Note that $\vec{R}' = \vec{R}$, as we trace to calculate Z.

Similar to the quantum spin systems, we proceed by discretizing the 'imaginary time' β,

$$e^{-\beta\hat{H}} = \left(e^{-\Delta\tau\hat{H}}\right)^M, \tag{8.8}$$

where $\Delta\tau = \beta/M$. Note that in coordinate representation, this product is a product of density matrices, explicitly,

$$\begin{aligned}\rho(\vec{R}_1, \vec{R}_{M+1}, \beta) = \int\cdots\int \mathrm{d}\vec{R}_2 \ldots \mathrm{d}\vec{R}_M\, \rho(\vec{R}_1, \vec{R}_2, \Delta\tau)\rho(\vec{R}_2, \vec{R}_3, \Delta\tau)\ldots \\ \ldots\rho(\vec{R}_M, \vec{R}_{M+1}, \Delta\tau).\end{aligned} \tag{8.9}$$

This equation is not particularly useful, as we don't know $\rho(\vec{R}, \vec{R}', \Delta\tau)$. However, in the limit $\Delta\tau \to 0$ ($M \to \infty$), we can try to approximate ρ. At fixed, large M, such an approximation will be most accurate in the limit $T \to \infty$, in other words the high-temperature limit of the density matrix. We will see in the following that the above equation becomes a high-dimensional integral over positive definite functions: the perfect use-case for Monte Carlo.

To approximate ρ, we employ a trick in analogy to the split-operator method of Sec. 3.3.3 and write

$$e^{-\Delta\tau(\hat{T}+\hat{V})+\frac{\Delta\tau^2}{2}\left[\hat{T},\hat{V}\right]+\ldots} = e^{-\Delta\tau\hat{T}}e^{-\Delta\tau\hat{V}}. \tag{8.10}$$

In the limit $\Delta\tau \to 0$, we can neglect the term with $\Delta\tau^2$. In coordinate representation, the potential term is again diagonal

$$\langle\vec{R}_i|e^{-\Delta\tau\hat{V}}|\vec{R}_{i+1}\rangle = e^{-\Delta\tau V(\vec{R}_i)}\delta(\vec{R}_i - \vec{R}_{i+1}). \tag{8.11}$$

The kinetic part, however, requires more work and is given by a multi-dimensional Gaussian integral. For simplicity, we first look at one particle in one dimension. Then,

$$\hat{T} = -\frac{\hbar^2}{2m}\partial_x^2 \tag{8.12}$$

and we define

$$\rho_{\text{free}}(x, x', \Delta\tau) = \langle x|e^{-\Delta\tau\hat{T}}|x'\rangle\,. \tag{8.13}$$

We insert resolutions of identity $\mathbb{1} = \int |p\rangle\langle p|\,\mathrm{d}p$, where $|p\rangle$ are the momentum eigenstates, to evaluate

$$\begin{aligned}\rho_{\text{free}}(x, x', \Delta\tau) &= \int \mathrm{d}p \int \mathrm{d}p'\, \langle x|p\rangle\, \langle p|e^{-\Delta\tau\hat{T}}|p'\rangle\, \langle p'|x'\rangle \\ &= \frac{1}{2\pi}\int \frac{\mathrm{d}p}{\hbar} e^{i(x-x')p/\hbar} e^{-\frac{\Delta\tau p^2}{2m}} \\ &= \sqrt{\frac{m}{2\pi\Delta\tau\hbar^2}} e^{-\frac{m}{2\Delta\tau\hbar^2}(x-x')^2}.\end{aligned} \tag{8.14}$$

Here, we have used $\langle x|p\rangle = e^{ipx/\hbar}/\sqrt{2\pi\hbar}$ and the fact that $\hat{T}$ is diagonal in $|p\rangle$. The same calculation also works in the many-particle, higher-dimensional problem, and we find

$$\rho_{\text{free}}(\vec{R}, \vec{R}', \Delta\tau) = \left(\frac{2\pi\hbar^2\Delta\tau}{m}\right)^{-Nd/2} e^{-\frac{|\vec{R}-\vec{R}'|^2}{2\hbar^2\Delta\tau/m}}. \tag{8.15}$$

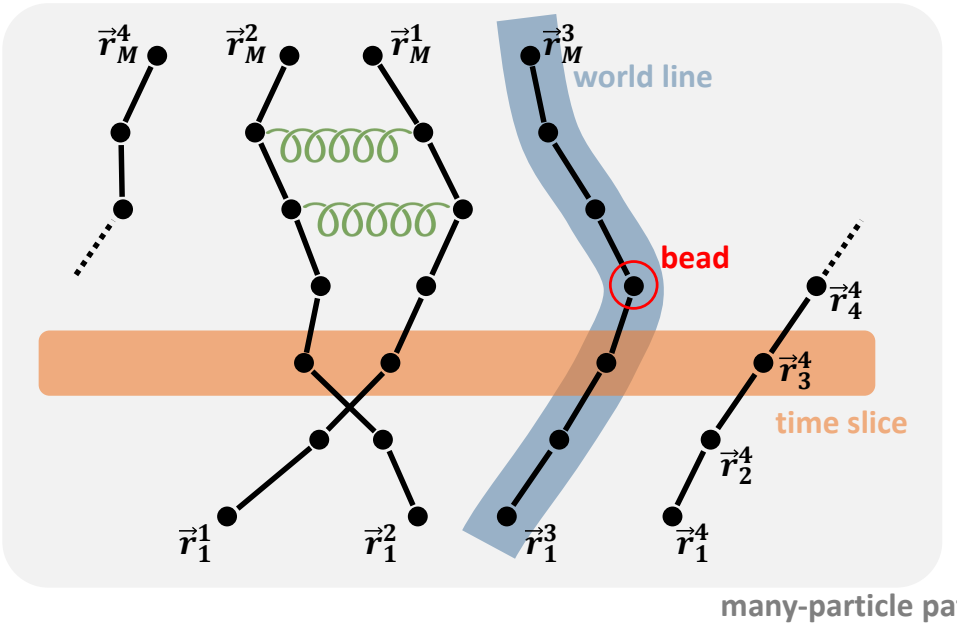


Figure 13: Many-particle path. The blue line denotes a world line of a particle. The orange line denotes a time slice (coordinates of all particles at a certain time step). Coordinates of one particle at one time step are called a bead. Each bead experiences two-body interactions with all other beads at the same time slice (denoted by green springs). To calculate a path integral, one has to sample over such many-particle paths (efficiently).

Finally, combining Eqs. (8.11) and (8.15) in the limit of large M and taking the trace, the partition function is given by

$$\begin{aligned} Z &= \int (d\vec{R}) \rho(\vec{R}, \vec{R}, \beta) \\ &= \int \left(\prod_{j=1}^{M} \mathrm{d}\vec{R}_j \right) \prod_{j=1}^{M} \left[\left(\frac{2\pi\hbar^2 \Delta\tau}{m} \right)^{-Nd/2} e^{-\frac{\left|\vec{R}_j - \vec{R}_{j+1}\right|^2}{2\hbar^2 \Delta\tau/m} - \Delta\tau V(\vec{R}_j)} \right] \end{aligned} \tag{8.16}$$

with $R_{M+1} = R_1$. The Trotter formula ensures that for $M \to \infty$, the equation for the partition function is exact and the dNM-dimensional integral can be computed with *classical Monte Carlo*. The key challenge is to determine the minimal value of M such that the statistical error inherent to Monte Carlo dominates over the Trotter error (this additional Trotter error does not exist in pure classical Monte Carlo).

Before we continue, it is useful to introduce a few useful definitions (see Fig. 13):

- **Many-particle path**: also called "system configuration", is the set of NMd coordinates $X = \{\vec{R}_1, \ldots, \vec{R}_M\}$, where d is the spatial dimension.
- **Time slice**: the j-th term of a system configuration, $\vec{R}_j$, which contains Nd coordinates of the N particles at "time" slice $(j-1)\Delta\tau$.
- **World line**: the set of all Md coordinates of one particle i $\{\vec{r}_1^{\,i}, \vec{r}_2^{\,i}, \ldots, \vec{r}_M^{\,i}\}$ at all time slices.

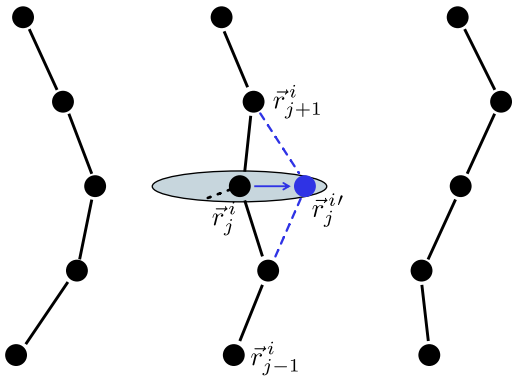


Figure 14: Single-bead update. We displace a bead $\vec{r}^{\,i}_j$ into $\vec{r}^{\,i\prime}_j$. The probability to accept this move is determined by Eq. (8.19).

- **Beads**: a bead is the set of d coordinates of one particle at one time slice $\vec{r}^{\,i}_j$.

The partition function in Eq. (8.16) corresponds to the classical partition sum of N interacting polymers (or "necklaces") of beads in $(d+1)$ dimensions [20]. This is the famous quantum to classical mapping which we encountered already for spin systems. In the current approximation, two polymers interact through V, as if they were connected with springs at every time slice.

8.1.1 Path sampling methods

We can again sample over the many-particle paths using a Markov chain. As in the case for spin systems, we want to employ the *Metropolis algorithm*. This algorithm consists of two steps: starting in X, (1) propose a new state X' with probability $T(X \to X')$, then accept/reject this proposal with the probability

$$A(X \to X') = \min\left(1, \frac{V_{X'} T(X' \to X)}{V_X T(X \to X')}\right). \tag{8.18}$$

We are free to choose the proposed update steps, as long as the probabilities possess the properties discussed in Sec. 7.1. The most elementary option is to displace a single randomly chosen bead of a randomly chosen particle's worldline, say $\vec{r}^{\,i}_j$, to another point $\vec{r}^{\,i\prime}_j$, which we uniformly sample from a sphere around the old position (see Fig. 14). By construction, $T(X \to X') = T(X' \to X)$, and we find the acceptance probability

$$A(X \to X') = \min\left\{1, \frac{e^{-m\left[(\vec{r}^{\,i}_{j-1}-\vec{r}^{\,i\prime}_j)^2+(\vec{r}^{\,i\prime}_j-\vec{r}^{\,i}_{j+1})^2\right]/2\hbar^2\Delta\tau}}{e^{-m\left[(\vec{r}^{\,i}_{j-1}-\vec{r}^{\,i}_j)^2+(\vec{r}^{\,i}_j-\vec{r}^{\,i}_{j+1})^2\right]/2\hbar^2\Delta\tau}} \cdot e^{-\Delta\tau\left[V(\vec{R}'_j)-V(\vec{R}_j)\right]}\right\}. \tag{8.19}$$

This defines a perfectly valid Monte Carlo scheme. However, for a large number of time slices such *single-bead* updates become extremely inefficient, as moving an

[20] The energy of the polymers is given by a tension term (between time slices) and an interaction term (within time slices),

$$H = \sum_j \sum_i \frac{m}{2(\hbar\Delta\tau)^2}\left(\vec{r}^{\,i}_j - \vec{r}^{\,i}_{j+1}\right)^2 + \sum_j V(\vec{R}_j). \tag{8.17}$$

Note that there is no kinetic energy in the expression, meaning that the beads are very heavy and hence, their movement is very slow.

entire world-line takes many steps. To increase efficiency, one can rigidly move an entire world line. With such moves, the rejection has to be done only with respect to the interaction part.

8.1.2 Calculating properties

The expectation value of a general operator $\hat{O}$ corresponding to a physical observable can be written as

$$\langle \hat{O} \rangle = \frac{1}{Z} \int O(X) W(X) \, \mathrm{d}X \,, \tag{8.20}$$

with X again a shorthand for a many-particle path. For an operator that is diagonal in coordinate representation, we simply calculate the expectation value of $O(\vec{R}_1)$, see Eq. (7.49) [21]. An example of an operator that is not diagonal is the Hamiltonian operator. For the energy per particle $E(N, \beta, \Omega)/N = \langle \hat{H} \rangle / N$, we can instead use the thermodynamic definition of the partition function and write the energy as the derivative of the partition function Z with respect to β,

$$\frac{E(N, \beta, \Omega)}{N} = -\frac{1}{NZ} \frac{\partial Z(N, \beta, \Omega)}{\partial \beta}. \tag{8.21}$$

Applying this derivative to the partition sum Eq. (8.16), we obtain for the energy per particle

$$\frac{E_{\mathrm{th}}}{N} = \left\langle \frac{d}{2\Delta\tau} - \frac{1}{MN} \sum_{j=1}^{M} \sum_{i=1}^{N} \frac{m}{2(\hbar\Delta\tau)^2} \left(\vec{r}_j^{\,i} - \vec{r}_{j+1}^{\,i} \right)^2 + \frac{1}{MN} \sum_{j=1}^{M} V(\vec{R}_j) \right\rangle. \tag{8.22}$$

8.1.3 Inverse temperature and imaginary time

We have seen that the partition sum of a quantum system can be written as a path integral in $d+1$ dimensions, which we have characterized as *imaginary time* evolution. We now make this correspondence even more explicit. For the Schrödinger equation

$$i\hbar \partial_t \varphi(\vec{R}, t) = \hat{H} \varphi(\vec{R}, t), \tag{8.23}$$

we can define the Green's function

$$G(\vec{R}, \vec{R}', t) = \langle \vec{R} | e^{-it\hat{H}/\hbar} | \vec{R}' \rangle \,, \tag{8.24}$$

which is the solution of the Schrödinger equation with initial conditions $\varphi(\vec{R}, 0) = \delta(\vec{R} - \vec{R}')$. We immediately see that the Green's function corresponds exactly to the density matrix in Eq. (8.6), if we replace β with $it/\hbar$.

8.1.4 Bose symmetry

The above expression for the partition sum is not symmetric under particle exchange. Hence, it is only valid for distinguishable particles. For identical particles—the

[21] Note that for better statistics, we can calculate $1/M \sum_j [1/Z \int \mathrm{d}X O(\vec{R}_j) W(X)]$.

typical case in quantum mechanics—we need to symmetrize the path integral. For Bose statistics, we symmetrize the density matrix as

$$\rho_{\text{Bose}}(\vec{R}_1, \vec{R}_2, \beta) = \frac{1}{N!} \sum_P \rho(\vec{R}_1, P\vec{R}_2, \beta), \tag{8.25}$$

where the $N \times N$ matrix P is one of the $N!$ permutation matrices that acts on the particle labels i in $\vec{R}_2$. Specifically,

$$P\vec{R}_2 = \left\{ \vec{r}_2^{\,p(1)}, \vec{r}_2^{\,p(2)}, \vec{r}_2^{\,p(3)}, \dots \right\}. \tag{8.26}$$

(Note that we have to introduce this permutation only once in the imaginary time evolution, not at every time step $\Delta\tau$.) If we trace the symmetrized density matrix, we obtain the partition sum for identical bosons

$$Z_{\text{Bose}}(N, \beta, \Omega) = \frac{1}{N!} \sum_P \int \cdots \int \left(\prod_{j=1}^{M} \mathrm{d}\vec{R}_j \right) \prod_{j=1}^{M} \left(\rho_{\text{free}}(\vec{R}_j, \vec{R}_{j+1}, \Delta\tau) e^{-\Delta\tau V(\vec{R}_j)} \right), \tag{8.27}$$

where it is understood that P enters through a change in boundary conditions to $\vec{R}_{M+1} = P\vec{R}_1$. A consequence is that the individual worldlines don't form closed loops anymore. Rather, the last bead $\vec{r}_M^{\,i}$ is connected to the first bead of the $p(i)$-th world line, $\vec{r}_1^{\,p(i)}$.

For small temperatures, where the thermal wavelength $\lambda_T = \sqrt{2\pi\hbar^2\beta/m}$ is larger than the interparticle distance, large permutation cycles acquire substantial weight in the partition sum. This is then responsible for macroscopic quantum phenomena, such as superfluidity or Bose-Einstein condensation. For large N, the $N!$ terms get quickly out of hand for sampling. Fortunately, all terms are still positive definite, and we can use standard Monte Carlo approaches for the high-dimensional integral.

Things are not so favorable for fermions, since we have to anti-symmetrize the density matrix, in other words all terms in the path integral acquire a sign ± 1 depending on the signature of the permutation. This means we do not deal with positive definite integrals anymore. This is the source of the infamous *fermionic sign problem*.

8.2 Diffusion Monte Carlo

So far, we have considered (path integral) Monte Carlo as a technique to obtain the partition function and associated observables. However, we are often interested in the ground state and its energy. For bosonic systems, we introduce with diffusion Monte Carlo an algorithm to extract exactly these quantities. We have already seen that imaginary time evolution of an arbitrary state will single out the ground state,

$$\begin{aligned} \varphi(\vec{R}, \tau) &= e^{-\tau H} \varphi(\vec{R}, 0) = \sum_n e^{-\tau E_n} a_n \varphi_n(\vec{R}) \\ &= e^{-\tau E_0} \left[a_0 \varphi_0(\vec{R}) + \sum_{n>0} e^{-\tau(E_n - E_0)} a_n \varphi_n(\vec{R}) \right] \\ &\overset{\tau\to\infty}{\to} e^{-\tau E_0} a_0 \varphi_0(\vec{R}). \end{aligned} \tag{8.28}$$

Further, we discussed in Sec. 8.1.3 that we can use the Green's function, or equivalently the density matrix to perform this imaginary time evolution,

$$\varphi(\vec{R},\tau) = \int \rho(\vec{R},\vec{R}',\beta=\tau)\varphi(\vec{R}',0)\,\mathrm{d}R' \tag{8.29}$$

with

$$\rho(\vec{R},\vec{R}',\beta) = \langle\vec{R}|e^{-\beta\hat{H}}|\vec{R}'\rangle\,. \tag{8.30}$$

Using a Trotter decomposition, in other words introducing (infinitesimal) time steps $\Delta\tau$, we can then factorize the density matrix into

$$\rho(\vec{R}_1,\vec{R}_2,\Delta\tau) = \left(\frac{m}{2\pi\Delta\tau}\right)^{dN/2} e^{-\frac{|\vec{R}_1-\vec{R}_2|^2}{2\Delta\tau/m}} e^{-\Delta\tau V(\vec{R}_2)}, \tag{8.31}$$

where we have set $\hbar = 1$. Finally, we can use the Monte Carlo scheme discussed in the previous section to perform the evolution.

While such a Monte Carlo method allows for the evaluation of the wave function to any imaginary time τ, it does not provide a systematic way to take the limit $\tau \to \infty$. The idea of *diffusion Monte Carlo* (DMC) is to evolve in imaginary time a large number M of "walkers" (also called "replicas" or "particles", even though that latter term is confusing), each described by its position $\vec{R}_k^{(\alpha)}$, $\alpha = 1,\cdots,M$ at time $\tau = k\Delta\tau$. Using the probability interpretation of the quantum mechanical wave function, the ground-state wave function $\varphi_0(\vec{R})$ is represented by the average density of walkers at large time,

$$\varphi_0(\vec{R}) = \lim_{k\to\infty}\langle\delta(\vec{R}_k^{(\alpha)} - \vec{R})\rangle_M. \tag{8.32}$$

For this approach to be possible, the ground-state wave function must be real and positive everywhere [22]. The main difference, then, between diffusion Monte Carlo and path-integral Monte Carlo is that the former considers many walkers $\vec{R}_k^{(\alpha)}$ at one imaginary time $\tau = k\Delta\tau$, while the latter considers one walker at many times, in other words a many-particle path $X = \{\vec{R}_1,\ldots,\vec{R}_M\}$.

We will see that the imaginary time evolution corresponds to a *diffusion process* for the walkers with the above interpretation of $\varphi(\vec{R},\beta)$ as the probability distribution for walker configurations $\vec{R}$. The simplest form of diffusion Monte Carlo algorithm that we discuss here assigns to each walker at time step k a position $\vec{R}_k^{(\alpha)}$ and a weight $w_k^{(\alpha)}$. A convenient starting configuration is

$$\varphi(\vec{R},\tau=0) = \delta(\vec{R}-\vec{R}_0), \tag{8.33}$$

meaning all the walkers are in the same state, $\vec{R}_0^{(\alpha)} = \vec{R}_0$, with the same weights $w_0^{(\alpha)} = 1\ \forall\alpha$. The time evolution of the wave function $\varphi(\vec{R},\tau)$, described by Eq. (8.29), is then exactly

$$\varphi(\vec{R},\tau) = \rho(\vec{R},\vec{R}_0,\tau) \tag{8.34}$$

and $\rho(\vec{R},\vec{R}_0,\tau)$ is a product of elementary factors in the discrete-time path integral, see Eq. (8.31). Each elementary factor is factorized into its kinetic part and its

[22] Recall that we always have the freedom to choose a global (independent of $\vec{R}$) phase. It turns out that, for a bosonic system, the ground-state wave-function can be chosen real and positive. Excited-state wave-functions (with finite angular momentum, to name an example) have nodes, as do fermionic ground-state wave-functions.

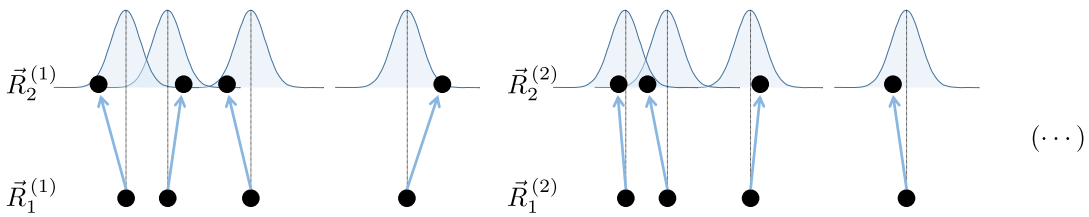


Figure 15: Diffusion Monte Carlo. Particles in each walker are going through a diffusion process, e.g. the position of a particle in the next time slice is randomly chosen from a Gaussian distribution around its previous position.

potential part, which are applied in succession to each walker. Namely, given a walker α at position $\vec{R}_{k-1}^{(\alpha)}$ with weight $w_{k-1}^{(\alpha)}$, the new position and weight are obtained as follows:

1. Change $\vec{R}_{k-1}^{(\alpha)} \to \vec{R}_k^{(\alpha)}$ according to a diffusion process described by the kinetic part. In other words, choose new coordinates around the old ones by drawing $(\vec{R}_k^{(\alpha)} - \vec{R}_{k-1}^{(\alpha)})$ from a Gaussian distribution with variance $\Delta\tau/m$. This step corresponds to a diffusion around $\vec{R}_{k-1}^{(\alpha)}$ and gives the algorithm its name (see Fig. 15). The formal reason is that the time-dependent Schroedinger equation for a free particle in imaginary time is identical to the heat equation,

$$\frac{\mathrm{d}\varphi}{\mathrm{d}\tau} = \frac{1}{2m}\nabla^2\varphi. \tag{8.35}$$

2. To account for the change in probability of a walker because of $e^{-\Delta\tau V(\vec{R})}$, we need to do something else aside from moving beads: We *change the weights and the number of walkers* in a given configuration. Let us first discuss the modification of the weights, which is done according to

$$w_k^{(\alpha)} = w_{k-1}^{(\alpha)} e^{-\frac{\Delta\tau}{2}\left[V(\vec{R}_k^{(\alpha)}) + V(\vec{R}_{k-1}^{(\alpha)})\right]}. \tag{8.36}$$

   Both factors together allow for a stochastic representation of the wave function

$$\varphi(\vec{R}, k\Delta\tau) = \left\langle w_k \delta(\vec{R}_k - \vec{R}) \right\rangle_M. \tag{8.37}$$

   One problem with this algorithm is that the weights $w_k^{(\alpha)}$ will vary considerably from one walker to another, so that the contribution of many walkers to the average Eq. (8.37) will be negligible, and the computer effort to simulate them will be wasted. All walkers should maintain identical weights for best efficiency, as in Eq. (8.32) if possible. This can be achieved by "cloning" the important walkers and "killing" the negligible ones, again stochastically, which is done in step 3.

3. Compute the normalization, or nominal weight, $w^\star = \frac{1}{M}\sum_\alpha w_k^{(\alpha)}$. Replace each walker α by a number of clones (all with weight $w^\star$) equal to

$$\mathrm{int}\left(\frac{w_k^{(\alpha)}}{w^\star} + r\right), \tag{8.38}$$

where $\mathrm{int}(x)$ is the integer part of x and r is a random number distributed uniformly in $[0, 1)$. Since this number can be zero, walkers with small weight can be killed. The average over r of this expression is $w_k^{(\alpha)}/w^\star$, so that each walker is replaced, on average, by its appropriate number of equal-weight clones. Note that the total number M of walkers will fluctuate by $O(M)$.

With these three very simple steps, one can already obtain interesting results. Two technical modifications are customary:

- One limits the maximum number of clones at step three. As can be seen from Eq. (8.36), the number of clones increases when the walker reaches a region of small potential. Since diffusion Monte Carlo is often used for systems with Coulomb interactions, where the potential is unbounded from below, this seems like a wise precaution. In any case, if this maximum number of clones is reached, it indicates that the variation in the weight over a single step is large, and thus that the step size $\Delta\tau$ is too large.

- As formulated, the nominal weight $w^\star$ varies as $\exp(-E_0\tau)$ for large τ. For $V(\vec{R}) \mapsto V(\vec{R}) - E_\mathrm{t}$ with E_t a trial energy, we can avoid having a vanishing or exploding weight. Specifically, we multiply all the weights $w^{(\alpha)}$ by $\exp(E_\mathrm{t}\Delta\tau)$ after each step. Stability of the weights is achieved when $E_\mathrm{t} = E_0$, which gives a simple means to compute the groundstate energy.

To finish our discussion of Monte Carlo methods, we explore in the following two powerful modifications of this simple algorithm.

8.2.1 Importance sampling

Fluctuations in the estimated wave function come mostly from Step 2. The variation of the weight of a single walker signals a waste of computer effort: a weight which tends to 0 indicates that the walker approaches a forbidden region (e.g., for a particle in a box, at the edge of the box); a weight which diverges occurs at a potential singularity (e.g., at the location of the charge in a Coulomb potential). In both cases, it would be advantageous to incorporate prior knowledge of the wave function in the diffusion step, so that walkers are discouraged or encouraged to enter the respective regions, and the weights vary less.

We can accomplish incorporating prior knowledge by choosing a "guiding" or "trial" wave function $\varphi_\mathrm{t}(\vec{R})$, and evolving in imaginary time the product $\Phi(\vec{R}, \tau) \equiv \varphi_\mathrm{t}(\vec{R})\varphi(\vec{R}, \tau)$. From the Schroedinger equation obeyed by $\varphi(\vec{R}, \tau)$,

$$-\hbar\frac{\mathrm{d}}{\mathrm{d}\tau}\varphi(\vec{R}, \tau) = \left[-\frac{\hbar^2}{2m}\nabla^2 + V(\vec{R})\right]\varphi(\vec{R}, \tau) \tag{8.39}$$

we obtain the equation obeyed by $\Phi(\vec{R},\tau)$. In particular,

$$\begin{aligned}
-\hbar\frac{\mathrm{d}}{\mathrm{d}\tau}\Phi(\vec{R},\tau) &= \varphi_{\mathrm{t}}(\vec{R})\left[-\frac{\hbar^2}{2m}\nabla^2 + V(\vec{R})\right]\varphi(\vec{R},\tau) \\
&= \left[-\frac{\hbar^2}{2m}\nabla^2 + V(\vec{R})\right]\Phi(\vec{R},\tau) + \frac{\hbar^2}{2m}\left(2\boldsymbol{\nabla}\varphi_{\mathrm{t}}\cdot\boldsymbol{\nabla}\varphi + \varphi\nabla^2\varphi_{\mathrm{t}}\right) \\
&= -\frac{\hbar^2}{2m}\nabla^2\Phi(\vec{R},\tau) + \frac{\hbar^2}{m}\boldsymbol{\nabla}\cdot\left[\left(\frac{\boldsymbol{\nabla}\varphi_{\mathrm{t}}}{\varphi_{\mathrm{t}}}\right)\Phi(\vec{R},\tau)\right] \\
&\quad + \left(V(\vec{R}) - \frac{\hbar^2}{2m}\frac{\nabla^2\varphi_{\mathrm{t}}}{\varphi_{\mathrm{t}}}\right)\Phi(\vec{R},\tau).
\end{aligned} \tag{8.40}$$

The first two terms define a new diffusion equation, with a *drift term* which can be simply added to the diffusion update in Step 1,

$$\vec{R}_k = \vec{R}_{k-1} + \frac{\hbar^2\Delta\tau}{2m}\frac{2\boldsymbol{\nabla}\varphi_{\mathrm{t}}(\vec{R})}{\varphi_{\mathrm{t}}(\vec{R})}\bigg|_{\vec{R}=\vec{R}_{k-1}}. \tag{8.41}$$

This forces the walkers to drift to regions with high probability. The last two terms in Eq. (8.40) define a new potential, to be used in the update of the weights in Step 2.

We can check the effect of a good trial wave function on the walkers' weight by choosing for φ_{t} the ground state φ_0 (assuming φ_0 is known). In this case, the last term in Eq. (8.40) is equal to $[-V(\vec{R}) + E_0]$ and the new potential is simply E_0, which is a constant independent of $\vec{R}$. Consequently, the weights of all walkers are multiplied by the same factor in Step 2 and, as a result, in Step 3, no walker is either cloned or killed.

Of course, the art lies in choosing good candidate wave functions, which depend on the problem at hand. Often, one chooses a trial wave functions of the form

$$\varphi_{\mathrm{t}}(\vec{R}) = \prod_{i<j} f_z(|\vec{r}_i - \vec{r}_j|), \tag{8.42}$$

where f_z incorporates two-particle correlations and are known as Jastrow factors.

8.2.2 Fermionic systems

A major application of diffusion Monte Carlo is to compute the energy of electrons in a molecule or a crystal with the positions for the atomic nuclei given. However, the electrons are indistinguishable fermions and the wave function should be anti-symmetric under interchange of any two of them. Thus, if the wave function is positive when electrons 1 and 2 are at positions $(\vec{r}_1,\vec{r}_2)$, it must be negative when they are at $(\vec{r}_2,\vec{r}_1)$. The configuration space (which has dimension dN for N electrons) has nodal surfaces separating positive and negative regions. If the location of these nodal surfaces was known, then one could perform distinct simulations (as many as there are disconnected regions), in each region of definite sign. In particular, if φ_0 is positive, then we apply the above algorithm with a potential barrier preventing the walkers to cross the nodal surface. If φ_0 is negative, we can apply

the same algorithm with the substitution $\varphi \to -\varphi$. Unfortunately, the location of the nodal surfaces is usually not known, and some ansatz of the form

$$\varphi_{\mathrm{t}'}(\vec{R}) = \varphi_{\mathrm{t}}(\vec{R}) \det_{l,n} \left[e^{i\vec{k}_l \cdot \vec{r}_n} \right], \tag{8.43}$$

is made. Here, the index $n = 1, \ldots, N$ labels the particles and $\vec{k}_l$ are the wave vectors compatible with periodic boundary conditions. This strategy is called the *fixed node approximation*.

Fixing the nodal surface away from its ground-state location can only increase the energy, such that the fixed node approximation gives a variational upper bound to the ground-state energy. One can try to relax the constraint, and move the nodal surface in the direction which most lowers the energy. The gradient of the wave function, which must be continuous across the nodal surface, can help in this relaxation strategy.

9 The electronic-structure problem

Due to the exponential scaling in the Hilbert space dimension for a many-body problem, we have so far only focused on (tractable) idealized model systems. These model Hamiltonians are well suited to study fundamental quantum phases and transitions between them. However, we are often interested in the properties of real-life materials or molecules. For example, we might want to study the binding energy and orbital structure of molecules, or we would like to know whether a material is an insulator or a metal. For this purpose, we need to calculate the electronic structure of the full (realistic) Hamiltonian.

In this chapter, we will learn about several approaches, in particular density functional theory (DFT), the most widely used method for electronic structure calculations in both chemistry and physics. While this method works exceedingly well for weakly correlated systems, such as normal metals, band insulators and semiconductors, care has to be taken when employing it for strongly correlated systems, such as nearly all transition metals.

9.1 The full Hamiltonian of matter

We consider in the following systems, such as molecules or solids, consisting of nuclei and electrons. The full many-body Hamiltonian of such a system is given by

$$\begin{aligned}\hat{H} = &-\sum_{j}^{N_\mathrm{e}} \frac{\hbar^2}{2m}\nabla^2_{\vec{r}_j} - \sum_{l}^{N_\mathrm{n}} \frac{\hbar^2}{2M_l}\nabla^2_{\vec{R}_l} + \frac{1}{2}\sum_{i\neq j}^{N_\mathrm{e}} \frac{e^2}{|\vec{r}_i - \vec{r}_j|} \\ &+ \frac{1}{2}\sum_{l\neq m}^{N_\mathrm{n}} \frac{Z_l Z_m e^2}{|\vec{R}_l - \vec{R}_m|} - \sum_{j=1}^{N_\mathrm{e}}\sum_{l=1}^{N_\mathrm{n}} \frac{Z_l e^2}{|\vec{r}_j - \vec{R}_l|} + V_\mathrm{SO} \\ = &\ \hat{T}_\mathrm{e} + \hat{T}_\mathrm{n} + V_\mathrm{ee} + V_\mathrm{nn} + V_\mathrm{en} + V_\mathrm{SO}.\end{aligned} \tag{9.1}$$

Here, m is the electron mass, $-e$ is its charge and $\vec{r}_j$ its coordinate; M_l is the mass of nucleus l, $Z_l e$ is its charge and $\vec{R}_l$ its coordinate; N_e and N_n are the total numbers of electrons and nuclei, respectively. Note that, in general, we assume charge neutrality, in other words $N_\mathrm{e} = \sum_l Z_l$. $\hat{T}_\mathrm{e}$ and $\hat{T}_\mathrm{n}$ are the kinetic terms for electrons and nuclei. The next three terms, V_ee, V_nn, V_en are the Coulomb-potential terms corresponding to electron-electron, nucleus-nucleus and electron-nucleus interactions. Although the Coulomb-potential terms are two-particle terms, V_nn and V_en can be approximated. We will discuss these approximations below. However, V_ee remains cumbersome, a truly many-body term. For generality, we have added the term V_SO to include spin-orbit coupling.

Adiabatic (Born-Oppenheimer) approximation

The Hamiltonian in Eq. (9.1) has unfortunately only one small parameter, which is m/M_l. As a result, the nuclei's motion is considered to be slow compared to that of the electrons ($\vec{v}_\mathrm{n} \ll \vec{v}_\mathrm{e}$) due to the large mass difference and the Coulomb forces being nearly equally strong (electrons are stronger accelerated). From the perspective

of the electrons, the nuclei stand still, such that we can treat the electronic wavefunction as a function of the instantaneous nuclear positions. That is, the electrons are assumed to follow the nuclei instantaneously, remaining in the same state of the electronic Hamiltonian with the nuclear coordinates treated as parameters. This allows for decoupling of the electronic and nuclear degrees of freedom and solve for

$$\Psi\left(\{\vec{r}_j\},\{\vec{R}_l\},t\right)=\chi\left(\{\vec{R}_l\},t\right)\Phi\big(\{\vec{r}_j\}\big), \tag{9.2}$$

where Φ is the ground state electron wavefunction for the current nuclei configuration.

The movement of the nuclei, in general oscillations around their equilibrium positions, is given by classical equations of motion and is nearly unaffected by the electrons. Still, every electronic state creates a potential, in which the nuclei move, giving rise to phonons and other lattice-mediated phenomena.

In this approximation, $\hat{T}_\mathrm{n}\approx 0$, $V_\mathrm{nn}\to V_\mathrm{M}$, the Madelung energy, and V_en becomes a one-body operator, describing the potential the electron feels moving in the field of immobile positive charges,

$$\hat{H}=\sum_{j=1}^{N_\mathrm{e}}\left[\frac{p_j^2}{2m}+V(\vec{r}_j)\right]+\frac{1}{2}\sum_{i\neq j}^{N_\mathrm{e}}\frac{e^2}{|\vec{r}_i-\vec{r}_j|}, \tag{9.3}$$

where

$$V(\vec{r}_j)=-\sum_{l=1}^{N_\mathrm{n}}\frac{Z_l e^2}{|\vec{r}_j-\vec{R}_l|}. \tag{9.4}$$

The first term in Eq. (9.3) is a one-body term, while the second term is two-body.

For numerical calculations a finite basis has to be introduced, which we will discuss later in this chapter. Then, using a basis set of L orbital wavefunctions $\{\psi_i\}$ and again neglecting spin-orbit coupling, the matrix elements of the Hamilton operator (9.3) are

$$t_{ij}=\int \mathrm{d}^3\vec{r}\,\psi_i^*(\vec{r})\left(-\frac{\hbar^2}{2m}\nabla^2+V(\vec{r})\right)\psi_j(\vec{r}), \tag{9.5}$$

$$V_{ijkl}=e^2\int \mathrm{d}^3\vec{r}\int \mathrm{d}^3\vec{r}\,'\,\psi_i^*(\vec{r})\psi_j(\vec{r})\frac{1}{|\vec{r}-\vec{r}\,'|}\psi_k^*(\vec{r}\,')\psi_l(\vec{r}\,'). \tag{9.6}$$

Note that quantum chemists distinguish between the self-consistent-field (SCF) approximation in a finite basis set and the Hartree-Fock (HF) limit, working in a complete basis. In physics both are known as Hartree-Fock approximation.

9.2 Hartree and Hartree-Fock method

Before trying to solve the electronic structure problem within the density functional theory, we first discuss two crude approximations, namely the "non-interacting" mean-field, or Hartree approximation and the Hartree-Fock approximation. What both methods have in common is that while they do not consider the effects of correlations on the ground state, they both take the Pauli principle into account

by the construction of the variational function. Specifically, the wavefunction is approximated as a single Slater determinant as introduced in Section 4.1.2,

$$\Phi(\vec{r}_1,\sigma_1;\ldots;\vec{r}_N,\sigma_N) = \frac{1}{\sqrt{N!}}\begin{vmatrix} \phi_1(\vec{r}_1,\sigma_1) & \ldots & \phi_N(\vec{r}_1,\sigma_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\vec{r}_N,\sigma_N) & \ldots & \phi_N(\vec{r}_N,\sigma_N) \end{vmatrix}. \tag{9.7}$$

Note that the *orthogonal* single-particle states ϕ_i combine a spin and an orbital part. For simplicity, we will abbreviate the notation in the following and write $\phi_i^\sigma(\vec{r}) = \phi_i(\vec{r},\sigma)$ and often drop the spin dependence.

9.2.1 Non-interacting (mean-field) approximation

The crudest approximation to the electronic structure problem is to assume non-interacting electrons. In order to mimic the many-body effects of the electron-electron interaction operator, an effective (mean) field is introduced and all interactions are replaced by V_{eff}, which is a single-particle operator,

$$V_{\text{en}} + V_{\text{M}} + V_{\text{ee}} \to V_{\text{eff}}. \tag{9.8}$$

The resulting problem is described by the single-particle Hamiltonian

$$\hat{H}_{\text{sp}} = -\frac{\hbar^2}{2m}\nabla^2_{\vec{r}} + V_{\text{eff}}(\vec{r}). \tag{9.9}$$

Note that the original electron is replaced by a "quasi-particle" that moves in the effective field of electrons and nuclei. The initial many-body problem is replaced by a set of single-particle equations of the form

$$\hat{H}_{\text{sp}}\,|\phi_i\rangle = \varepsilon_i\,|\phi_i\rangle \quad \text{for each quasi-particle } i. \tag{9.10}$$

Having found the N_{e} lowest single-particle states, the ground-state wavefunction is constructed using the Slater determinant, Eq. (9.7).

9.2.2 Hartree-Fock approximation

The Hartree-Fock approximation is again based on the assumption of independent electrons. However, unlike the above approximation of non-interacting electrons, the Hamiltonian in Hatree-Fock theory includes the Coulomb interaction V_{ee}. The orthogonal single-particle wavefunctions ϕ_i^σ in Eq. (9.7) are chosen such that the energy $\langle\Phi|\hat{H}|\Phi\rangle$ is minimized.

The full energy given Φ in the form of Eq. (9.7) then reads

$$\langle\Phi|\hat{H}|\Phi\rangle = \sum_{i,\sigma}\int d^3\vec{r}\,\phi_i^{\sigma*}(\vec{r})[-\frac{\hbar^2}{2m}\nabla^2 + V_{\text{en}}(\vec{r})]\phi_i^\sigma(\vec{r}) + V_{\text{M}} \tag{9.11}$$

$$+ \sum_{i,j,\sigma,\sigma'} e^2 \int \mathrm{d}^3\vec{r}\,\mathrm{d}^3\vec{r}'\,\phi_i^{\sigma*}(\vec{r})\phi_j^{\sigma'*}(\vec{r}')\frac{1}{|\vec{r}-\vec{r}'|}\phi_i^\sigma(\vec{r})\phi_j^{\sigma'}(\vec{r}') \tag{9.12}$$

$$- \sum_{i,j,\sigma} e^2 \int \mathrm{d}^3\vec{r}\,\mathrm{d}^3\vec{r}'\,\phi_i^{\sigma*}(\vec{r})\phi_j^{\sigma*}(\vec{r}')\frac{1}{|\vec{r}-\vec{r}'|}\phi_j^\sigma(\vec{r})\phi_i^\sigma(\vec{r}'). \tag{9.13}$$

The first line contains single-particle terms only. The second and third lines contain the interaction terms with the first of these corresponding to the energy of classical charge densities, called the direct or Hartree interaction. The third line, however, is a direct consequence of the Pauli exclusion principle and is called the exchange interaction. Note that for the derivation of this expression, we have assumed orthonormal basis functions. This orthonormality needs to be maintained during minimization using Lagrange multipliers.

9.2.3 Configuration-Interaction

The approximations used in Hartree-Fock and related methods are based on the non-interacting electron picture. 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

$$|\Phi_{\mathrm{HF}}\rangle = \prod_{\mu=1}^{N} \hat{c}_\mu^\dagger |0\rangle \,, \tag{9.14}$$

one or two of the $\hat{c}_\mu^\dagger$ are replaced by other orbitals $\hat{c}_i^\dagger$:

$$|\Phi_0\rangle = \left(1 + \sum_{i,\mu} \alpha_\mu^i \hat{c}_i^\dagger \hat{c}_\mu + \sum_{i<j,\mu<\nu} \alpha_{\mu\nu}^{ij} \hat{c}_i^\dagger \hat{c}_j^\dagger \hat{c}_\mu \hat{c}_\nu\right) |\Phi_{\mathrm{HF}}\rangle \,. \tag{9.15}$$

The energies are then minimized using this variational ansatz with α_μ^i and $\alpha_{\mu\nu}^{ij}$ the variational parameters. 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 we can calculate the low lying eigenstates in $O\left((N+M)^2\right)$ steps.

Further improvements are possible by allowing more than double substitutions. The optimal method treats the full quantum problem of dimension $(N+M)!/N!M!$. Quantum chemists call this method "full-CI".

9.3 Density functional theory

9.3.1 Hohenberg-Kohn theorems

In 1964, Hohenberg and Kohn provided a proof that any property of an N-electron system can be obtained from its ground-state electron density

$$n_0(\vec{r}) = N \int \Psi_0^*(\vec{r}, \vec{r}_2, \ldots, \vec{r}_N) \Psi_0(\vec{r}, \vec{r}_2, \ldots, \vec{r}_N) \, \mathrm{d}\vec{r}_2 \ldots \mathrm{d}\vec{r}_N \,. \tag{9.16}$$

In other words, every property of the system is a unique functional of its ground-state electron density. The resulting density functional theory (DFT) is an exact theory that allows us to write the total energy as a functional of the electron density, the minimum of which should occur at the ground state density. The exact form of the functional is still unknown though, which limits the applications of DFT in condensed matter physics to materials with weakly correlated electrons.

Remember that the Hamiltonian of the electron system is given by

$$\hat{H}_{\text{el}} = -\frac{\hbar^2}{2m}\sum_j \nabla_j^2 + \frac{1}{2}\sum_{i\neq j}\frac{e^2}{|\vec{r}_i - \vec{r}_j|} + V_{\text{ext}} \tag{9.17}$$

with

$$V_{\text{ext}} = \int v_{\text{ext}}(\vec{r})n(\vec{r})\,\mathrm{d}\vec{r} \tag{9.18}$$

and $v_{\text{ext}}(\vec{r})$ is the external potential density. V_{ext} is the only non-universal part of the Hamiltonian and as such determines all physical and chemical properties of the system. The whole density functional theory rests on the two theorems by Hohenberg and Kohn:

1. The external potential and its density $v_{\text{ext}}(\vec{r})$ stay in one-to-one correspondence with the ground state electron density $n_0(\vec{r})$. Thus, $\Psi_0 = \Psi_0[n_0]$, since it is determined by $v_{\text{ext}}(\vec{r})$.

2. There exists a universal functional for the energy $E[n]$, valid for any external potential V_{ext}. The exact ground-state energy is at the global minimum of this functional of the electron density (which is only a function of 3 variables instead of $3N$) and the minimum of the functional occurs when $n(\vec{r}) = n_0(\vec{r})$:

$$\begin{aligned} E^{[v_{\text{ext}}]}[n] &= \min_{\Psi\to n}\langle\Psi|\hat{T} + \hat{V}_{\text{ee}} + \hat{V}_{\text{ext}}|\Psi\rangle \\ &= \min_{\Psi\to n}\langle\Psi|\hat{T} + \hat{V}_{\text{ee}}|\Psi\rangle + \int v_{\text{ext}}(\vec{r})n(\vec{r})\,\mathrm{d}\vec{r} \\ &= F[n] + \int v_{\text{ext}}(\vec{r})n(\vec{r})\,\mathrm{d}\vec{r}, \end{aligned} \tag{9.19}$$

   where the minimum is taken over wavefunctions that reproduce the electron density n. $F[n]$ is a universal density functional, since $\hat{T}$ and $\hat{V}_{\text{ee}}$ are not system specific but the same for any N-electron system.

Note that this is not yet any simpler, rather it presents an exact rewriting of the problem. Furthermore, it is not clear how to extract any physical information from the ground-state energy density without solving again for the exact ground-state wavefunction. Here, the Kohn-Sham scheme provides help.

9.3.2 Kohn-Sham solution scheme

The basic idea behind the approach of Kohn and Sham is to find a *non-interacting* system that has the same particle density as the interacting one. A many-body state for non-interacting fermions takes the form of a Slater determinant, Eq. (9.7) of single-particle orbitals $\phi_j(\vec{r})$, where for simplicity, j stands for a collective index representing all quantum numbers. The density is then $n(\vec{r}) = \sum_j |\phi_j(\vec{r})|^2$ and the energy functional reads

$$\begin{aligned} E[n] &= E_{\text{K}}[n] + E_{\text{C}}[n] + E^{\text{XC}}[n] + \int v_{\text{ext}}(\vec{r})n(\vec{r})\,\mathrm{d}\vec{r} \\ &= -\frac{\hbar^2}{2m}\sum_j\langle\phi_j|\nabla^2|\phi_j\rangle + \frac{1}{2}\int e^2\frac{n(\vec{r})n(\vec{r}')}{|\vec{r}-\vec{r}'|}\,\mathrm{d}\vec{r}\,\mathrm{d}\vec{r}' + E^{\text{XC}}[n] + \\ &\qquad + \int v_{\text{ext}}(\vec{r})n(\vec{r})\,\mathrm{d}\vec{r}. \end{aligned} \tag{9.20}$$

In the above equation, we introduced the kinetic functional $E_{\mathrm{K}}[n]$, the Hartree energy functional $E_{\mathrm{C}}[n]$, and $E^{\mathrm{XC}}[n]$ is the exchange-correlation functional:

$$E^{\mathrm{XC}}[n] = \langle\Psi[n]|\hat{T}|\Psi[n]\rangle - E_{\mathrm{K}}[n] + \langle\Psi[n]|\hat{V}_{\mathrm{ee}}|\Psi[n]\rangle - E_{\mathrm{C}}[n]. \tag{9.21}$$

With this decomposition, we have put all unknown contributions in the exchange-correlation functional $E^{\mathrm{XC}}[n]$, which captures the effects of interactions.

Note that the Hartree energy $E_{\mathrm{C}}[n]$ is the self-interaction energy of the density $n(\vec{r})$ treated as a classical charge density. $\langle\Psi|\hat{V}_{\mathrm{ee}}|\Psi\rangle$ describes the full Coulomb energies of interacting, correlated electrons and is thus long ranged. $\langle\psi|\hat{V}_{\mathrm{ee}}|\psi\rangle - E_{\mathrm{C}}$ filters out all these long-range interactions and we thus expect that $E^{\mathrm{XC}}[n]$ is short ranged.

To calculate the ground state energy, we have to minimize $E[n]$ with respect to n under the constraint that the total number of electrons is conserved $\int \mathrm{d}^3r\, n(\vec{r}) = N$. To incorporate the particle conservation, we use the method of Lagrange multipliers

$$\begin{aligned} 0 &\stackrel{!}{=} \delta\left[E[n] + \lambda\left(\int \mathrm{d}^3r\, n(\vec{r}) - N\right)\right] \\ &= \delta E_{\mathrm{K}}[n] + \delta E_{\mathrm{C}}[n] + \delta E^{\mathrm{XC}}[n] + \delta V_{\mathrm{ext}}[n] + \lambda\delta\left(\int \mathrm{d}^3r\, n(\vec{r}) - N\right). \end{aligned} \tag{9.22}$$

Instead of minimizing with respect to the density, we perform the minimization by varying the functional with respect to ϕ_j^* with

$$\frac{\delta n(\vec{r})}{\delta\phi_j^*(\vec{r})} = \frac{\delta\sum_i \phi_i^*\phi_i}{\delta\phi_j^*} = \sum_i \delta_{ij}\phi_i = \phi_j(\vec{r}). \tag{9.23}$$

With this, we find the condition

$$\begin{aligned} 0 &= \frac{\delta E_{\mathrm{K}}[n]}{\delta\phi_j^*} + \frac{\delta E_{\mathrm{C}}[n]}{\delta n}\frac{\delta n}{\delta\phi_j^*} + \frac{\delta E^{\mathrm{XC}}[n]}{\delta n}\frac{\delta n}{\delta\phi_j^*} + \frac{\delta\int \mathrm{d}^3r\, v_{\mathrm{ext}}(\vec{r})n(\vec{r})}{\delta n}\frac{\delta n}{\delta\phi_j^*} + \lambda\frac{\delta n}{\delta\phi_j^*} \\ &= -\frac{\hbar^2}{2m}\nabla^2\phi_j(\vec{r}) + \int \mathrm{d}^3r'\,\frac{n(\vec{r}')}{|\vec{r}-\vec{r}'|}\phi_j(\vec{r}) + \mu^{\mathrm{XC}}(\vec{r})\phi_j(\vec{r}) + v_{\mathrm{ext}}(\vec{r})\phi_j(\vec{r}) + \lambda\phi_j(\vec{r}), \end{aligned} \tag{9.24}$$

where $\mu^{\mathrm{XC}}(\vec{r})$ is the functional derivative of the exchange-correlation functional. We have thus reduced the problem to single-particle equations, one for each of the N non-interacting electrons in the external potential $V_{\mathrm{eff}}(\vec{r})$ with corresponding single-particle energies $\lambda = \varepsilon_j$. These equations are known as the Kohn-Sham equations:

$$\text{KS 1}: \qquad \left[-\frac{\hbar^2}{2m_e}\nabla^2 + V_{\mathrm{eff}}(\vec{r})\right]\phi_j(\vec{r}) = \varepsilon_j\phi_j(\vec{r}), \tag{9.25}$$

$$\text{KS 2}: \qquad V_{\mathrm{eff}}(\vec{r}) = \int \mathrm{d}^3r'\,\frac{n(\vec{r}')}{|\vec{r}-\vec{r}'|} + \mu^{\mathrm{XC}}(\vec{r}) + v_{\mathrm{ext}}(\vec{r}), \tag{9.26}$$

where $\phi_j(\vec{r})$ are called Kohn-Sham orbitals. The effective potential in the KS equations is a functional of the density, which in turn is constructed out of the single-particle states. These types of self-consistency problems are solved iteratively. This leads to the

Kohn-Sham solution iteration scheme

1. Make some initial guess for the effective potential V_{eff}^0.
2. Solve the single-particle equation (KS 1) to find $\phi_j(\vec{r})$.
3. Calculate the charge density for these states $n(\vec{r}) = \sum_i |\phi_j(\vec{r})|^2$.
4. Construct the new V_{eff}^1 from this density using (KS 2) and compare to the previous one.
5. Keep following this procedure until convergence is reached, e.g. $\left|V_{\text{eff}}^{\text{new}} - V_{\text{eff}}^{\text{old}}\right| < \alpha_{\text{threshold}}$ twice in a row.

The effective potential (KS 2) is the exact one, but only once the exact form of E^{XC} is provided. This is the problem!

9.3.3 Exchange correlation potential

The functional $E^{\text{XC}}[n]$ and therefore also μ^{XC} are unknown. After all, we put all unknown terms in it. The simplest approximation assumes that the two are only functionals of the local density. This approximation is called the *local density approximation* (LDA). Since, as we already noted above, E^{XC} is short-ranged, this is a reasonable approximation. Specifically, we write

$$E^{\text{XC}}[n] \approx \int n(\vec{r}) \varepsilon^{\text{XC}}(n(\vec{r}))\, \mathrm{d}\vec{r} = \int n(\vec{r}) \left[\varepsilon^{\text{X}}(n(\vec{r})) + \varepsilon^{\text{C}}(n(\vec{r}))\right] \mathrm{d}\vec{r}, \tag{9.27}$$

where ε^{XC} is the exchange correlation energy density, ε^{X} is the exchange energy density and ε^{C} is the correlation energy density. The functional is an integral with an integrand that depends only on the density at the coordinate where it is evaluated.

Since the energy functional should be universal, it follows that it is the same as for the homogeneous electron gas, in other words the jellium model of interacting electrons for a uniform distribution of ions. For the uniform electron gas, the exchange energy density is known exactly

$$\varepsilon^{\text{X}}\left(n(\vec{r})\right) = -\frac{3}{4}\left(\frac{3}{\pi}\right)^{1/3} \left(n(\vec{r})\right)^{1/3} = -\frac{3}{4}\left(\frac{9}{4\pi^2}\right)^{1/3} \frac{1}{r_s}, \tag{9.28}$$

where $r_s = \left(\frac{3}{4\pi n}\right)^{1/3}$ is the radius of the sphere that an electron would on average occupy in the uniform electron gas of density n.

Analytic expressions for ε^{C} are known only in the high- and low-density limits. Monte Carlo simulations are used for intermediate values. Various interpolation schemes are based on these known points, e.g. (due to Goedecker-Teter-Hutter)

$$\varepsilon^{\text{XC}}(r_s) = -\frac{A_0 + A_1 r_s + A_2 r_s^2 + A_3 r_s^3}{B_1 r_s + B_2 r_s^2 + B_3 r_s^3 + B_4 r_s^4}. \tag{9.29}$$

While this is in principle a bad approximation—correlation effects are non-local and the density in a real crystal is far from being uniform—it works great in many cases.

It is worth mentioning, however, that there are better approximations than the LDA, e.g., the local spin density approximation, where one distinguishes between $n_\uparrow$ and $n_\downarrow$, or the generalized gradient approximation (GGA), where one includes also $\boldsymbol{\nabla} n, \nabla^2 n$, etc.

9.4 Basis functions

Depending on the problem at hand, different basis functions for the single-particle wavefunctions are better suited than others. We will introduce in the following the most common ones.

9.4.1 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):

$$\psi^i_{nlm}(r,\theta,\phi) \propto r^{n-1}e^{-\zeta_i r}Y_{lm}(\theta,\phi). \tag{9.30}$$

These wavefunctions 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 Eq. (9.6) for basis functions centered around two different nuclei at position $\vec{R}_A$ and $\vec{R}_B$. There, we have to evaluate integrals containing terms like

$$\frac{1}{|\vec{r}-\vec{r}'|}e^{-\zeta_i|\vec{r}-\vec{R}_A|}e^{-\zeta_j|\vec{r}-\vec{R}_B|}, \tag{9.31}$$

which cannot be solved in closed form.

The *Gauss-Type-Orbitals* (GTO)

$$\psi^i_{nlm}(\vec{r}) \propto x^l y^m z^n e^{-\zeta_i r^2} \tag{9.32}$$

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

$$e^{-\zeta_i|\vec{r}-\vec{R}_A|^2}e^{-\zeta_j|\vec{r}-\vec{R}_B|^2} = Ke^{-\zeta|\vec{r}-\vec{R}|^2} \tag{9.33}$$

with

$$K = e^{-\frac{\zeta_i\zeta_j}{\zeta_i+\zeta_j}|\vec{R}_A-\vec{R}_B|^2}, \tag{9.34}$$

$$\zeta = \zeta_i + \zeta_j, \tag{9.35}$$

$$\vec{R} = \frac{\zeta_i\vec{R}_A + \zeta_j\vec{R}_B}{\zeta_i+\zeta_j}. \tag{9.36}$$

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

$$\frac{1}{|\vec{r}-\vec{r}'|} = \frac{2}{\sqrt{\pi}}\int_0^\infty \mathrm{d}t\, e^{-t^2(\vec{r}-\vec{r}')^2}. \tag{9.37}$$

Thus, all the integrals in Eq. (9.6)) 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.

9.4.2 The free electron gas

For the free electron gas with Hamilton operator

$$\hat{H} = -\sum_{i=1}^{N_e} \frac{\hbar^2}{2m} \nabla_{\vec{r}_i}^2 + e^2 \sum_{i<j} \frac{1}{|\vec{r} - \vec{r}'|}, \tag{9.38}$$

the ideal choice for basis functions are plane waves

$$\psi_{\vec{k}}(\vec{r}) = \exp\Big(-i\vec{k} \cdot \vec{r}\Big). \tag{9.39}$$

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.

9.4.3 Pseudo-potentials

The electrons in inner, fully occupied shells do not contribute to the chemical bindings. To simplify the calculations and reduce the size of the required basis, 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.

10 Quantum computing

While this lecture is called "computational quantum physics", we did not spend any time studying the computing substrate, in other words the hardware we use to perform computations. Implicitly, we had in mind that the algorithms are executed on classical digital computing hardware.

At the same time, quantum computing is an emerging technological reality and the first generation of noisy intermediate scale quantum (NISQ) computers are in the labs of academic and industrial research institutions. In this context, we want to use this chapter for a brief discussion of the possibilities that quantum computers hold, specifically for simulations of quantum mechanical systems. Quantum computing algorithms (such as quantum Fourier transform, Shore's algorithms and others) have potential applications in many areas of science and technology including big data processing, encryption, and more. However, quantum computers are powerful also for computations that model intrinsically quantum mechanical systems themselves. As such, they can be used in material science, quantum chemistry, and in particular drug design. Importantly, algorithms considered for this task should at no point involve steps that scale exponentially with the system size. If that were the case, one might just as well solve the problem on a classical computer and there is no need for a quantum computer.

10.1 A quantum computer

To keep the exposition relatively general, we will not discuss in detail the hardware foundations of quantum computers here. We simply assume that quantum computers consist of a set of qubits, that is, quantum two-level systems with basis states $|0\rangle$ and $|1\rangle$ each, which can be initialized in an uncorrelated product state. In the diagrammatic representation, a set of initialized qubits (for example, three qubits initialized in states $|q_1\rangle$, $|q_2\rangle$, $|q_3\rangle$) looks like

$$\begin{aligned} |q_1\rangle &\text{ —} \\ |q_2\rangle &\text{ —} \\ |q_3\rangle &\text{ —} \end{aligned}$$

where each $|q_i\rangle$ can be either $|0\rangle$ or $|1\rangle$. Then, the computation consists of performing unitary operators, called gate operations (represented by unitary matrices) on the qubits.

10.1.1 Quantum gates

Any gate operation can be built up from just two types of gates:

1. Single qubit rotations. We will assume that the Pauli matrices have the conventional representations

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad Y = \begin{pmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{pmatrix}, \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{10.1}$$

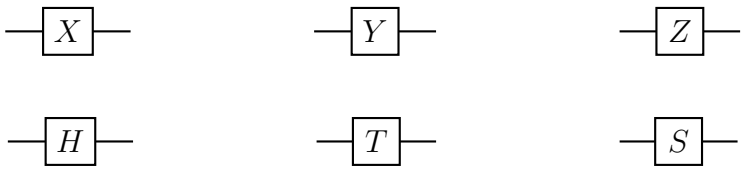
in the $\{|0\rangle, |1\rangle\}$ basis of a given qubit. Of all the potential SU(2) rotations, a particularly commonly used one is the Hadamard gate,

$$H = \frac{X+Z}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{10.2}$$

which switches an x-polarized state into a z-polarized state via a $\pi/2$ rotation around the y axis. Or, in other words, it brings the state $|0\rangle$ into $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and the state $|1\rangle$ into $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Other common one-qubit gates are the S and T gates

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{\mathrm{i}\pi/4} \end{pmatrix}, \qquad S = T^2 = \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{i} \end{pmatrix}. \tag{10.3}$$

In the diagrammatic representation single-qubit gates are represented simply as



and so on.

2. Two-qubit gates. Particularly common two-qubit gates are *controlled-U* gates, which in the two-qubit basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ have the form

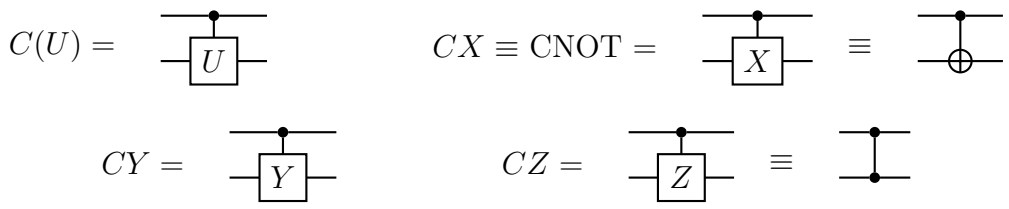
$$C(U) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & & \\ 0 & 0 & & U \end{pmatrix}, \tag{10.4}$$

where U is a one-qubit gate. In the controlled-U gate, the first qubit acts as a control for application of the U gate to the second qubit, meaning that if $|q_1\rangle = |1\rangle$, the gate U will be applied to $|q_2\rangle$; if $|q_1\rangle = |0\rangle$, the second qubit will be left unchanged. Commonly used examples of such gates are controlled-X (CX), controlled-Y (CY) and controlled-Z (CZ) gates.

The CX gate is also called *controlled-NOT* gate (CNOT), as it performs the NOT operation (controlled by the first qubit) on the second qubit,

$$\mathrm{CNOT} \equiv CX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{10.5}$$

The diagrammatic representations for controlled two-qubit gates are



There exist many other gates in addition to those we described, of course. An important question is, which gates are necessary to form a universal set, meaning they can be effectively used to compose any desired operation on the qubits. The so-called Clifford gates (S, H, CNOT) together with T form such a universal set. Without T, however, this is not the case. All quantum circuits that can be built from the Clifford gates alone can be efficiently simulated classically, where "efficiently" means with only polynomially increasing computation as the number of qubits is scaled.

10.1.2 Measurement

The final operation that a quantum computer needs to be able to perform are measurements. Here lies both its strength and weakness compared to a conventional physical system: For one, we can in principle measure any qubit or set of qubits simultaneously. Let us assume the measurement is done in the Z eigenbasis and we denote by $|z_i\rangle\,,\ z_i = \{0, 1\}$ the eigenstates of qubit i. The measurement projects the state $|\psi\rangle$ on one of the basis states $|z_1, z_2...\rangle$ with the probability $|\langle z_1, z_2...|\psi\rangle|^2$ and is thus statistical in nature. Therefore, we have to repeat the measurement many times to obtain the desired information about the state or correlation function that we seek. Typically, measurements are repeated $O(1000)$ times. However, with this procedure we have access to all correlation functions of our quantum computer, which is very powerful.

10.1.3 Errors

Quantum computations are subject to various errors. Which ones prevail depends on the hardware foundations. Fundamentally, each qubit is subject to decoherence due to its coupling to the environment. It thus loses its quantum state and is described by a mixed density matrix as this decoherence progresses. In addition, all gate operations are prone to errors, as they may not land the qubit exactly on the desired final state. Finally, there are read-out or measurement errors, meaning that the measurement does not return the value that corresponds to the state of the qubit before.

One fundamentally distinguishes analog and digital quantum computation. In analog quantum computation, these errors are accepted as intrinsic to the procedure, while attempts are made to mitigate them quantitatively. Results of analogue quantum computations are a priori continuous. In contrast, digital quantum computation seeks to eliminate the effects of these errors completely (with certainty exponentially close to 1) and therefore allows for computations with quantized, digital outcomes (just as in digital classical computers). Digital quantum computers rely on error-stabilized logical qubits. These represent a single two-dimensional Hilbert space, in other words a single qubit, but are made out of hundreds or thousands of *physical* qubits. Error correction algorithms then continuously correct for potential errors that occur in the physical qubits such that the state of the computational qubit is unaffected. To date, no error-corrected qubit and hence no digital quantum computer exists.

10.2 Representing the Hilbert space

A prerequisite for solving a quantum mechanical model on a quantum computer is to find a mapping of the Hilbert space basis states of the model to the basis states of the quantum computer's qubits.

This is most straight-forward for a spin-1/2 system, where one can simply map each spin-1/2 to a qubit. We give three explicit examples of spin states constructed in this way in Fig. 16. The circuit a) constructs the ground state of a trivial paramagnet given by

$$\hat{H}_{\text{triv}} = -\sum_i \sigma_i^x, \tag{10.6}$$

which, for the example of a chain of length N, is a trivial product state,

$$|\text{PM}\rangle = \frac{1}{\sqrt{2^N}} \sum_{\vec{r}} |\vec{r}\rangle = |+\rangle^{\otimes N}, \tag{10.7}$$

where $\vec{r}$ represents all possible binary strings of length N and $|+\rangle$ is the eigenstate of σ^x with eigenvalue $+1$.

The circuit b) constructs the so-called *cat state*, which is named after Schrödinger's cat and was already introduced in its matrix product state representation. In our case it is

$$|\text{cat}\rangle = \frac{1}{\sqrt{2}}(|000000\rangle + |111111\rangle). \tag{10.8}$$

This state is in the ground-state subspace of an Ising ferromagnet, but would not be assumed in a physical system with spontaneous symmetry breaking. Due to small symmetry breaking terms, either the all-up or the all-down configuration would be favored.

Finally, the circuit c) constructs a topologically nontrivial paramagnetic state, which is a ground state of the Hamiltonian

$$\hat{H}_{\text{topo}} = -\sum_i \sigma_{i-1}^z \sigma_i^x \sigma_{i+1}^z, \tag{10.9}$$

with periodic boundary conditions imposed. Since all terms in H_{topo} commute, its unique ground state $|\text{GS}\rangle$ is defined by

$$\sigma_{i-1}^z \sigma_i^x \sigma_{i+1}^z |\text{GS}\rangle = |\text{GS}\rangle, \quad \forall i. \tag{10.10}$$

This is the so-called *graph state* [23].

The case is more difficult for fermionic systems, where various transformations exist that keep track of the fermionic statistics. One is the *Jordan-Wigner transformation*, which maps states as

$$|\Psi\rangle = |n_{N-1}, \cdots, n_0\rangle \quad \rightarrow \quad |z_{N-1}, \cdots, z_0\rangle, \qquad \text{with } n_i = z_i, \tag{10.11}$$

where N is the number of fermionic orbitals/site and $n_i \in \{0, 1\}$, $i = 0, \cdots, N-1$ is the occupation number of the respective orbital/site. The operators are mapped as

$$\hat{c}_i \quad \rightarrow \quad A_i Z_{i-1} \cdots Z_0, \qquad \hat{c}_i^\dagger \quad \rightarrow \quad A_i^\dagger Z_{i-1} \cdots Z_0, \tag{10.12}$$

[23] This state is topologically equivalent to the AKLT state

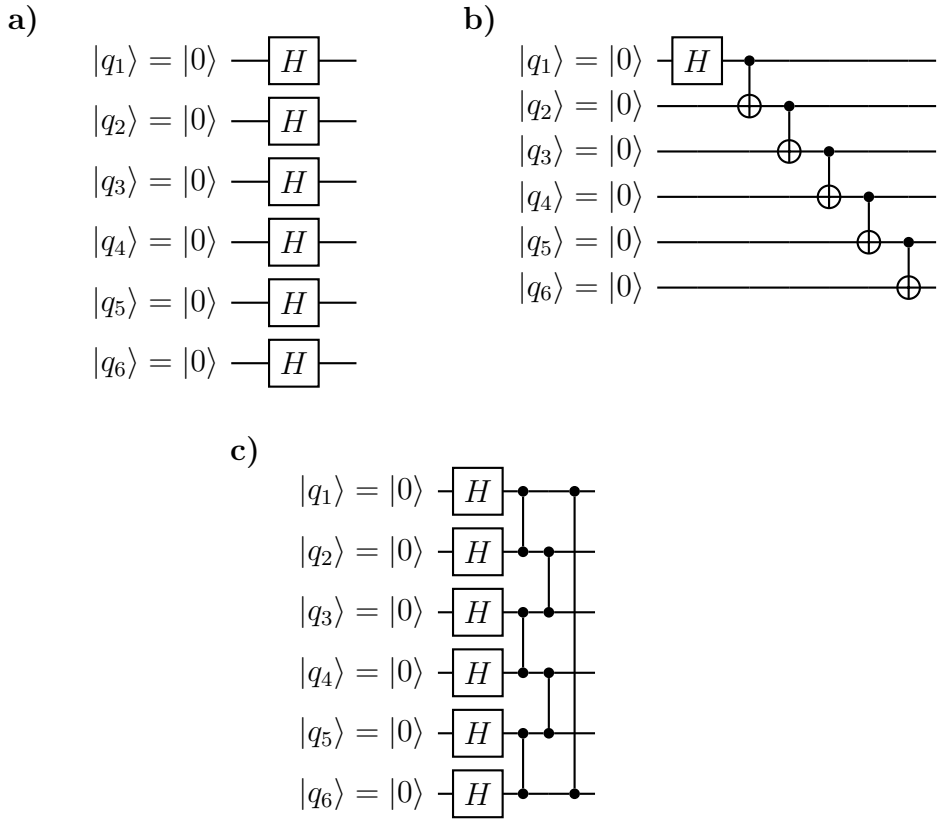


Figure 16: Examples of quantum circuits: a) constructs the ground state of the trivial paramagnet, b) constructs the six qubit cat state, c) constructs the ground state of the topological paramagnet.

where $A_i = (X_i + \mathrm{i}Y_i)/2$. Note that the ordering of the orbitals is arbitrary, but a specific choice may produce more or less computational overhead for a given problem. Using this encoding, measuring the parity requires $O(N)$ operators, whereas updating an occupation number can be done with $O(1)$ operators.

An alternative is *parity encoding*, where one stores the parity of the occupation of the first $(i+1)$ orbitals/sites $0, 1, \cdots, i$ in the qubit i. That means

$$|\Psi\rangle = |n_{N-1}, \cdots, n_0\rangle \quad \rightarrow \quad |z_{N-1}, \cdots, z_0\rangle\,, \qquad \text{with } z_i = \left[\sum_{j=0}^{i} n_i\right] \mod 2. \tag{10.13}$$

For the creation and annihilation operators, the parity of all following qubits is switched, which is the computationally expensive part for the parity encoding

$$\hat{c}_i \quad \rightarrow \quad X_{N-1} \cdots X_{i+1}(X_i Z_{i-1} + \mathrm{i}Y_i), \qquad \hat{c}_i^\dagger \quad \rightarrow \quad X_{N-1} \cdots X_{i+1}(X_i Z_{i-1} - \mathrm{i}Y_i). \tag{10.14}$$

Using this encoding, measuring the parity requires only $O(1)$ operators, whereas updating an occupation number can be done with $O(N)$ operators.

This is by far not a complete list of possible encodings. Another popular choice goes by the name Bravyi-Kitaev encoding and is somewhat of a hybrid between Jordan-Wigner and parity encodings.

10.3 Variational quantum eigensolver

There are several approaches to obtaining the spectrum of an operator (such as the Hamiltonian) on a quantum computer. An exact algorithm is based on the quantum Fourier transform. However, this requires a large number of gate operations and is thus not the method of choice for current quantum hardware. An alternative are variational approaches, one of which, the variational quantum eigensolver, we want to discuss here in short. The algorithms typically consist of parts that can be executed on classical computers and parts that are executed on the quantum computer.

We recall that the variational principle states that an upper bound to the ground state energy E_0 can be obtained as

$$E_0 \leq \frac{\langle \Psi(\boldsymbol{\theta})|\hat{H}|\Psi(\boldsymbol{\theta})\rangle}{\langle \Psi(\boldsymbol{\theta})|\Psi(\boldsymbol{\theta})\rangle}, \tag{10.15}$$

where $|\Psi(\boldsymbol{\theta})\rangle$ is an arbitrary wave function family that depends on the variational parameters collected in the vector $\boldsymbol{\theta}$. The energy estimate of the VQE algorithm is then simply the minimum of the righthand side of Eq. (10.15) over $\boldsymbol{\theta}$. While the expectation value is evaluated on the quantum computers, the minimization is performed on a classical computer. The most straight-forward algorithm for this latter task would be of gradient-descend type. However, because all the measurements need to be performed on the quantum computer and are noisy, we need to use an algorithm that does not require gradients of the righthand side of Eq. (10.15). Such algorithms exist, with an example going by the name COBYLA [24].

What is a good choice of variational state? There is a trade-off between introducing too many variational parameters/using too many CNOT gates and having a too small variational subspace. We are just introducing one family of states here, which are motivated by quantum chemistry and are called *unitary coupled cluster* (UCC) states. They are in essence the same states that we considered in the configuration interaction method and are written as

$$|\Psi(\boldsymbol{\theta})\rangle = e^{\hat{T}(\boldsymbol{\theta})-\hat{T}^\dagger(\boldsymbol{\theta})}\,|\Psi_0\rangle\,, \tag{10.16}$$

with the excitation operator $\hat{T}(\boldsymbol{\theta})$ and a Hartree-Fock/single Slater determinant state $|\Psi_0\rangle$. Using the combination $\hat{T}(\boldsymbol{\theta})-\hat{T}^\dagger(\boldsymbol{\theta})$ has an additional advantage besides returning a unitary operator: in Jordan-Wigner encoding some expressions cancel, halving the circuit depth compared to general operators. The excitation operator $\hat{T}(\boldsymbol{\theta})$ can include excitations of arbitrary order. Up to second order, it reads

$$\hat{T}(\boldsymbol{\theta}) = \hat{T}_1(\boldsymbol{\theta}_1) + \hat{T}_2(\boldsymbol{\theta}_2), \tag{10.17}$$

with

$$\begin{aligned} \hat{T}_1(\boldsymbol{\theta}_1) &= \sum_{i,j} \theta_{1,i,j}\hat{c}_i^\dagger \hat{c}_j, \\ \hat{T}_2(\boldsymbol{\theta}_2) &= \sum_{i,j} \theta_{2,i,j,k,l}\hat{c}_i^\dagger \hat{c}_k^\dagger \hat{c}_j \hat{c}_l. \end{aligned} \tag{10.18}$$

Clearly, this expansion can be systematically continued with terms that contain 6 operators and so on. With these single and double (SD) excitations, which are the

[24] Constrained Optimization BY Linear Approximation

most popular choice, the ansatz is called UCCSD. Of course, much of the success of the method depends on the appropriate choice of the state $|\Psi_0\rangle$. The UCCSD is commonly applied to problems from quantum chemistry.

The UCC yields approximative good results for the ground state of weakly interacting systems. However, the drawback of this variational form is the huge circuit depth, which scales as

$$\binom{N_{\text{occ}}}{2}\binom{N_{\text{empty}}}{2} N_{\text{qubits}} = \frac{N_{\text{occ}}(N_{\text{occ}}-1)}{2}\frac{N_{\text{empty}}(N_{\text{empty}}-1)}{2} N_{\text{qubits}}, \tag{10.19}$$

where the first two factors are the number of terms in $\hat{T}_2(\boldsymbol{\theta}_2)$ and the number of qubits N_{qubits} is roughly equal to the number of gates needed for one such term.