Lecture notes for PHY 52006 EP

Computational and Machine Learning methods for Quantum Many-Body Physics

Filippo Vicentini

Academic year 2025/2026

Practical session material can be found at
github.com/PhilipVinc/ComputationalQuantumPhysics

Those notes are a work in progress being built throught this academic year, to help you and the students of next years. They still contain several imperfections, errors, wrong statements, missing references and a strongly biased point of view. They are incomplete.

They are not meant for being publicly shared.

If you find errors or sections that are particularly unclear please let me know in class or by email.

Contents

Foreword

Congratulations! You chose to follow this course on Computational and Machine Learning Quantum Physics, which, in my biased yet humble opinion, is one of the best choices you could have taken according to the courses offered to you this year.

TODO...

Chapter 1

Motivation

In introductory courses on quantum mechanics, you were introduced to the physics of *single-particle* quantum systems. Canonical examples include a single spin evolving under a static or time-dependent magnetic field, the propagation of a wavepacket in free space or in the presence of a potential barrier, or the dynamics of a particle confined in a potential well. While these problems often require several pages of analytical calculations, it is nevertheless possible to compute their observables *exactly*, thanks to the underlying simplicity of the Hilbert space and, in many cases, to the presence of strong symmetries or integrability.

One of the greatest successes of single-particle quantum mechanics is the prediction of the energy levels of the hydrogen atom. This system, consisting of a single electron orbiting a nucleus, is one of the very few non-trivial quantum systems that remains exactly solvable, owing to its central potential and the resulting hidden symmetries. It can indeed be treated to very high precision by solving a well-defined set of differential equations. The remarkable agreement between theory and experiment in this case stands as one of the foundational triumphs of quantum mechanics.

However, all other elements of the periodic table contain not one but several electrons, which interact with one another and generically become entangled. The treatment of atoms with $N \geq 2$ electrons becomes dramatically more complicated. Already at this level, exact solutions using pen and paper are no longer possible: the Schrödinger equation ceases to be separable, the dimension of the relevant Hilbert space grows combinatorially with the number of particles, and the resulting equations are, in general, analytically unsolvable.

This does not imply that analytical approaches are useless. On the contrary, early pioneers of quantum mechanics developed extraordinarily clever approximation schemes—such as mean-field theories, variational principles, and perturbative expansions—that allowed them to predict physical properties with remarkable accuracy while keeping the mathematical complexity under control. However, these approximations are often uncontrolled, must be validated *a posteriori*, and require significant physical insight to construct. Moreover, they are rarely universal: each atom, molecule, or physical situation typically demands its own carefully adapted approach.

Atoms themselves are merely the elementary building blocks of nature, and they are almost never found in isolation. Modern technology instead relies on aggregates of atoms, whether in the form of molecules or extended solid-state systems. In such systems, qualitatively new collective phenomena that cannot be inferred from the properties of isolated atoms alone emerge. Common examples are chemical bonding, magnetism or superconductivity. As a consequence, quantum physics lies at the heart of many technologically relevant problems, and our ability to address several major societal challenges ultimately depends on our understanding of complex interacting quantum many-body systems.

A well-known example is the Haber-Bosch process, which converts nitrogen into ammonia for use as fertilizer and is of central importance to modern industrial agriculture.[1] Despite its importance,

[1] Admittedly, I often wonder whether a much more profound impact on society could be achieved by reducing our reliance on industrial farming practices that progressively impoverish the soil, rather than by pursuing technological advances that allow us to exhaust it even faster. There exist many *real-world* applications of quantum physics with genuine societal relevance, yet this particular process is often presented—by startups, researchers, and policymakers alike—as the flagship example of "Quantum/AI/Physics for Society". One may hope that future scientists will be more honest in advertising which problems truly matter.

the Haber-Bosch process is extremely energy intensive, and for decades researchers have attempted to understand its microscopic mechanisms, in particular the role of electronic correlations at catalytic surfaces, in order to identify efficient catalysts capable of lowering the activation temperature.

Similarly, our understanding of the quantum mechanisms that allow chlorophyll to capture light and convert it into chemical energy remains incomplete. This process involves highly non-equilibrium quantum dynamics, strong coupling to vibrational and environmental degrees of freedom, and subtle coherence effects. Despite decades of study, we still struggle to faithfully reproduce these mechanisms in artificial systems, highlighting once again the difficulty of accurately describing complex quantum dynamics in interacting many-body systems.

But chemistry is not the only domain in which quantum mechanics has demonstrated its importance. Condensed-matter physics, which aims at predicting the properties of gases, liquids, and solids, must naturally deal with systems composed of an enormous (effectively infinite) number of interacting degrees of freedom. At first sight, this complexity would appear entirely intractable.

By a fortunate accident[2], the apparent complexity of ordinary metals such as copper or iron, in the solid state and around room temperature, largely disappears. In these systems, the collective behaviour of electrons can often be described *as if* they were weakly interacting, quasi-independent particles moving in an effective background potential. This remarkable simplification underlies the early successes of classical and semiclassical theories of electronic transport, such as the Drude model and its quantum extension due to Sommerfeld, which together explain with surprising accuracy basic properties like electrical conductivity, heat capacity, and the existence of a Fermi surface.

However, this benign situation is far from generic. As one considers more complex compounds, reduced dimensionality, or lower temperatures, the independent-particle picture rapidly breaks down. Electronic correlations, quantum statistics, and collective effects become dominant and can no longer be treated as small perturbations. In this regime, entirely new phases of matter emerge, driven by strong interactions and quantum coherence, and the theoretical description of the system becomes qualitatively more challenging.

This strongly correlated quantum regime is of central importance for a wide range of technological applications. High-temperature superconductors, for instance, remain poorly understood despite decades of intense theoretical and experimental effort, and the discovery or rational design of new materials with higher critical temperatures or larger critical currents would represent a major technological breakthrough. Similarly, correlated quantum materials play a crucial role in modern electronics and spintronics, in quantum magnets with potential applications to information storage, and in low-dimensional systems where quantum effects dominate transport and optical properties. More recently, these same strongly interacting systems have also emerged as promising platforms for quantum technologies, ranging from quantum simulators to components of future quantum computers.

The quest to understand and predict the properties of those materials, molecules, atoms and compounds that drives our search beyond what can be analytically accomplished.

[2] A miracle sometimes attributed to screening, sometimes to quasiparticles, and sometimes to the collective decision of theorists to stop asking further questions once the answers start agreeing with experiments.

Chapter 2

Introduction to Computational Physics

2.1 Formalism

In this section we introduce the formalism used throught the rest of the lectures

2.1.1 Single-particle

Hilbert space (Single particle): We will consider particles (such as fermions, bosons or spins) whose state is defined in a single-particle Hilbert space $\mathcal{H}$ of dimension $\dim[\mathcal{H}] = d$. We call the **computational basis of** $\mathcal{H}$ the set of orthonormal[1] basis vectors $|x\rangle$ upon which we can expand the wave-function such that

$$\mathcal{H} \ni |\psi\rangle = \sum_{x}^{d} \psi_x |x\rangle , \tag{2.1}$$

where $\psi_x \in \mathbb{C}$ is the **representation of the wave-function** $|\psi\rangle$ **in the computational basis** $\{|x\rangle\}_x$. As an example, for a qubit, $x = \{0, 1\}$, for a spin-1/2, $x = \{\downarrow, \uparrow\}$, for bosons in a Fock space $x = \{0, \dots, d\}$ is an occupation number. We will often refer to this scalar variable x as the single **degree of freedom** of the system.

We will often refer to such Hilbert spaces as **Discrete Hilbert Spaces** because they are labelled by a countable ($\approx$ discrete) basis. The same reasoning also applies to infinite-dimensional Hilbert spaces such as $L^2(\mathbb{R})$ whose basis is $B(L^2(\mathbb{R})) = \mathbb{R}$. In such case, the discrete sum over x of eq. (2.1) becomes an integral. We will refer to those spaces as **Continuous Hilbert Spaces** because their basis is continuous[2].

Single-particle operators An operator $\hat{A}$ is a map associating elements in an Hilbert space to elements of the same Hilbert space. Its action is formally defined as

$$\begin{aligned} \hat{A} &: \mathcal{H} \to \mathcal{H} \\ &|\psi\rangle \to \hat{A} |\psi\rangle = |A\psi\rangle . \end{aligned}$$

Unless otherwise noted, we will assume that the Hilbert space is closed under the action of the operators we consider, meaning that $|A\psi\rangle \in \mathcal{H}\ \forall \psi \in \mathcal{H}$. It is possible to relax this condition, but some care must be taken and therefore we will discuss it later as a special case[3].

[1] In those lecture notes we consider only the case of orthonormal bases. This requirement could in some cases be relaxed to also account for overcomplete bases (such as coherent states), but we won't be dealing with it for now. Therefore we assume that $\langle x|y\rangle = \delta_{x,y}$. basis vectors $B(\mathcal{H}) = \{|x\rangle\}$ upon which we canonically expand

[2] For the first part of the course we will focus on discrete spaces, as they are simpler to reason and work with.

[3] A simple example of a space that is not closed under the action of an operator is a (many-body) fock space with N total particles, and the ladder operator $\hat{a}$ or $\hat{a}^\dagger$ which change the total particle number and therefore go out of the considered space.

Computational representation of a wave-function

The *ket* or *wave-function* $|\psi\rangle$ is a vector of an Hilbert space. This is an *abstract* object which cannot be stored directly on a computer.
To represent the wave-function on a calculator, we must chose a specific computational basis. We can then store the **representation of the wave-function in the computatinal basis** as a complex-valued vector or more efficient data structure.
Most importantly, the same *wave-function* can have multiple representations, some more efficient than others. It is important to keep in mind that when doing calculations numerically, we can only operate on a specific representation.

Example. Consider a spin-$\frac{1}{2}$ system and the state

$$|\psi\rangle = \frac{1}{\sqrt{2}}\big(|\uparrow\rangle + |\downarrow\rangle\big).$$

In the canonical computational basis $B = \{|\uparrow\rangle, |\downarrow\rangle\}$, its representation is the complex vector

$$\vec{\psi_B} = \begin{pmatrix} \langle\uparrow|\psi\rangle \\ \langle\downarrow|\psi\rangle \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}.$$

If instead we choose the basis $B_x = \{|+\rangle, |-\rangle\}$ with $|\pm\rangle = (|\uparrow\rangle \pm |\downarrow\rangle)/\sqrt{2}$, the same abstract state reads

$$\psi_{B_x} = \begin{pmatrix} \langle+|\psi\rangle \\ \langle-|\psi\rangle \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Numerical representation Looking at eq. (2.1) it is evident that we can **query** the wave-function for every possible value x of our *computational basis* by applying the bra $\langle x|$ to the wave-function $|\psi\rangle$. Somewhat redundantly, we can write the *query* as $\langle x|\psi\rangle = \psi_x$.

This expression might seem redundant, and mathematically trivial, but at a conceptual level it is not. The ket $|\psi\rangle$ represent the quantum *state* itself, while $\psi(x) = \psi_x$ are the individual complex amplitudes that determine the state in a specific basis. From a mathematical description, knowledge of the state ψ is equivalent to knowing ALL associated complex amplitudes (in an arbitrary basis). But from an algorithmic perspective there is a remarkable difference: you might *know* (have access to) the state, but you do not necessarily know the individual complex amplitudes unless you query them individually. And you are free to query only a limited subset of them[4].

If the space is $d-$dimensional, we can query the $d-$ different values in order to construct the column vector $\boldsymbol{\psi}$ representing the wave-function $|\psi\rangle$ in the computational basis:

$$\boldsymbol{\psi} = \begin{pmatrix} \langle 0|\psi\rangle \\ \langle 1|\psi\rangle \\ \vdots \\ \langle d|\psi\rangle \end{pmatrix} = \begin{pmatrix} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_d \end{pmatrix}. \tag{2.2}$$

This allows us to go form the abstract notion of *quantum state* to an algorithmically useful $d-$dimensional vector $\boldsymbol{\psi} \in \mathbb{C}^d$, which can be stored on a calculator as an array. *The major consequence is that, when working with a computer, we can treat wavefunctions as numerical vectors of complex numbers.*

Likewise, a finite-dimensional $\hat{A}$ can be stored in the computational basis by **querying** the matrix elements. Formally, this is achieved by respectively left- and right- multiplying $\hat{A}$ by every possible $\langle x|$ and $|y\rangle$. As there are d possible values of x, and d possible values of y, we will get d^2 elements which

[4] A terrible analogy might be that of a book: if you have a book, it does not mean you know what the book talks about. You need to read (cf query) every page, or a subset of them, to actually know what is inside. Even if you own the book, you know nothing about it until you query it.

can be arranged in the matrix $\boldsymbol{A}$:

$$\boldsymbol{A} = \begin{pmatrix} \langle 0|\hat{A}|0\rangle & \langle 0|\hat{A}|1\rangle & \cdots & \langle 0|\hat{A}|d\rangle \\ \langle 1|\hat{A}|0\rangle & \langle 1|\hat{A}|1\rangle & \cdots & \langle 1|\hat{A}|d\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle d|\hat{A}|0\rangle & \langle d|\hat{A}|1\rangle & \cdots & \langle d|\hat{A}|d\rangle \end{pmatrix} = \begin{pmatrix} A_{00} & A_{01} & \cdots & A_{0d} \\ A_{10} & A_{11} & \cdots & A_{1d} \\ \vdots & \vdots & \ddots & \vdots \\ A_{d0} & A_{d1} & \cdots & A_{dd} \end{pmatrix}, \tag{2.3}$$

which is the representation of the operator $\hat{A}$ in the computational basis.

It is possible to show that applying an operator to a wave-function can be numerically performed by multiplying the matrix representation of the operator by the vector representation of the wave-function, as long as both are in the same computational basis.

$$\begin{aligned} (A\psi)_x = \langle x|A\psi\rangle = \langle x|\hat{A}|\psi\rangle = \langle x|\hat{A}\mathbb{I}|\psi\rangle & \qquad (2.4)\\ = \sum_y \langle x|\hat{A}|y\rangle\langle y|\psi\rangle & \qquad (2.5)\\ = \sum_y A_{xy}\psi_y & \qquad (2.6)\\ & \qquad (2.7) \end{aligned}$$

As the equation above must be verified $\forall x$, it corresponds to the matrix-vector product $\boldsymbol{A\psi}$.

2.1.2 Many-body formalism:

In Quantum Physics we usually deal with systems composed of several interacting degrees of freedom, such as spins, fermions or bosons.

Hilbert Space (Many-Body): The state of a *many-body* system composed of N identical degrees of freedom is

$$\mathcal{H}^{\otimes N} = \otimes_{i=1}^{N}\mathcal{H} = \mathcal{H}\otimes\mathcal{H}\otimes\mathcal{H}\otimes\cdots\otimes\mathcal{H}, \tag{2.8}$$

where $\mathcal{H}$ is the single-particle Hilbert space of the independent particles. The **computational basis of** $\mathcal{H}^{\otimes N}$ is the set of basis vectors $B(\mathcal{H}^{\otimes N}) = \{|x_1, x_2, \ldots, x_N\rangle\}$ upon which we can canonically expand the wavefunction. If the local degrees of freedom x_i can take over d different values, the total number of combinations of the many-body basis vectors is $\dim[\mathcal{H}^{\otimes N}] = d^N$.

The many-body wave-function can be expanded in this basis in the same spirit of eq. (2.1),

$$\mathcal{H}^{\otimes N} \ni |\psi\rangle = \sum_{x_1}^{d}\sum_{x_2}^{d}\cdots\sum_{x_N}^{d}\psi_{x_1,x_2,\ldots,x_N}|x_1, x_2, \ldots, x_N\rangle, \tag{2.9}$$

where all the d^N basis elements will appear. We will mostly use a more compact notation of $\sum_{x_1,\ldots,x_N}$ to indicate all the sums.

We remark that $\psi_{x_1,x_2,\ldots,x_N}$ above is technically an $N-$tensor where every *leg* has dimension d, which is the point of view usually employed by the community of tensor network algorithms. However, for simplicity of notation, in this manuscript we will mostly treat it as a vector with a single index $\boldsymbol{x} \leftrightarrow x_1, x_2, \ldots, x_N$, which corresponds to the original indices[5].

It is also possible to consider spaces made up of different degrees of fredom, such as spins coupled to bosons, but we will restrict ourselves to the case of identical degrees of freedom for a simpler presentation.

[5]Both pictures have clearly the same number of total entries, and are threfore equivalent. Implementation-wise, the connection might be as simple as a reshape operation from tensor shape $(d, d, \ldots, d)$ to $(2^d,)$.

Operators (many-body): Operators acting on the many-body Hilbert space $\mathcal{H}^{\otimes N}$ can be classified according to how many degrees of freedom they couple.

A **single-body operator** acts non-trivially on only one degree of freedom at a time, with the identity acting on all others. If $\hat{a}^{(i)}$ denotes a single-particle operator acting on site i, the corresponding many-body operator is

$$\hat{A} = \sum_{i=1}^{N} \hat{a}^{(i)} = \sum_{i=1}^{N} \mathbb{I} \otimes \cdots \otimes \underbrace{\hat{a}}_{i\text{-th position}} \otimes \cdots \otimes \mathbb{I}. \tag{2.10}$$

Examples include the total magnetization $\hat{M}_z = \sum_i \sigma_i^z$ or the kinetic energy $\hat{T} = \sum_i \hat{p}_i^2/2m$.

A **two-body operator** couples pairs of degrees of freedom and takes the general form

$$\hat{B} = \sum_{i<j} \hat{b}^{(i,j)}, \tag{2.11}$$

where $\hat{b}^{(i,j)}$ acts non-trivially only on sites i and j. Most physical interactions fall into this category, such as the Ising interaction $\sum_{i<j} \sigma_i^z \sigma_j^z$ or the Coulomb repulsion $\sum_{i<j} e^2/|\hat{r}_i - \hat{r}_j|$.

More generally, a **k-body operator** couples k degrees of freedom simultaneously. Physical Hamiltonians are typically sums of one- and two-body terms, though effective Hamiltonians arising from perturbative treatments may contain higher-body interactions.

An operator is said to be **K-local** if it can be written as a sum of terms, each acting non-trivially on at most K degrees of freedom. Most physical Hamiltonians are K-local with K independent of the number of degrees of freedom N, which has profound computational implications: the matrix representation of such operators is *sparse*, with at most $\mathcal{O}(N \cdot d^K)$ non-zero entries per row rather than d^N.

Numerical representation (many-body): The numerical representation of many-body states and operators follows the same logic as in the single-particle case, but the Hilbert-space dimension grows exponentially with the number of degrees of freedom.

Let $\mathcal{H}$ be a local Hilbert space of dimension $\dim[\mathcal{H}] = d$ and consider N sites, so that $\mathcal{H}^{\otimes N} = \mathcal{H}^{\otimes N}$ has dimension d^N. Fix a computational basis $B(\mathcal{H}) = \{|x\rangle\}_{x=0}^{d-1}$ and the corresponding product basis of $\mathcal{H}^{\otimes N}$,

$$|\boldsymbol{x}\rangle \equiv |x_1, x_2, \ldots, x_N\rangle := |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_N\rangle\,, \qquad \boldsymbol{x} \in \{0, \ldots, d-1\}^N. \tag{2.12}$$

A generic many-body wave-function $|\psi\rangle \in \mathcal{H}^{\otimes N}$ is then expanded as

$$|\psi\rangle = \sum_{\boldsymbol{x}} \psi_{\boldsymbol{x}} |\boldsymbol{x}\rangle\,, \qquad \psi_{\boldsymbol{x}} \equiv \psi_{x_1, x_2, \ldots, x_N} \in \mathbb{C}. \tag{2.13}$$

Numerically, one often stores the coefficients $\psi_{\boldsymbol{x}}$ as a complex array of shape $(d, d, \ldots, d)$ (a rank-N tensor), or as a flattened vector $\boldsymbol{\psi} \in \mathbb{C}^{d^N}$. Flattening requires a convention (e.g. lexicographic ordering of $\boldsymbol{x} = (x_1, \ldots, x_N)$), which is exactly what libraries like NumPy implement when reshaping arrays.

Example (three-body separable/product state). Consider $N = 3$ with local dimension d, and assume the state is separable:

$$|\psi\rangle = |\phi\rangle \otimes |\phi\rangle \otimes |\phi\rangle\,, \qquad |\phi\rangle = \sum_{i=0}^{d-1} \phi_i |i\rangle\,. \tag{2.14}$$

Then the many-body amplitudes factorize as

$$\psi_{ijk} = \phi_i\, \phi_j\, \phi_k. \tag{2.15}$$

In other words, the rank-3 tensor ψ_{ijk} is the outer product of the vector $\boldsymbol{\phi}$ with itself three times, and the flattened vector $\boldsymbol{\psi} \in \mathbb{C}^{d^3}$ is the Kronecker (tensor) product of $\boldsymbol{\phi}$ with itself three times. In NumPy, the latter is implemented by `np.kron`:

```python
import numpy as np

# local state |phi> in C^d
phi = np.array([1.0, 2.0, 3.0], dtype=np.complex128) # example with d=3
phi = phi / np.linalg.norm(phi)

# many-body product state |psi> = |phi> |phi> |phi>
psi_vec = np.kron(np.kron(phi, phi), phi) # shape: (d**3,)

# view it as a rank-3 tensor psi[i,j,k]
d = phi.shape[0]
psi_tensor = psi_vec.reshape(d, d, d)

# check factorization on a random index triple
i, j, k = 0, 1, 2
print(psi_tensor[i, j, k], phi[i] * phi[j] * phi[k])
```

With the standard reshape convention, the coefficient `psi_tensor[i,j,k]` corresponds to the basis vector $|i\rangle \otimes |j\rangle \otimes |k\rangle$, and `psi_vec` stores the same information in a single 1D array.

Operators. A many-body operator $\hat{A}$ is represented (in the same basis) by a matrix of size $d^N \times d^N$ with entries

$$A_{\boldsymbol{x},\boldsymbol{y}} = \langle \boldsymbol{x}|\hat{A}|\boldsymbol{y}\rangle \,. \tag{2.16}$$

Applying $\hat{A}$ to a state $|\phi\rangle = \hat{A}\,|\psi\rangle$ becomes the matrix-vector product

$$\phi_{\boldsymbol{x}} = \sum_{\boldsymbol{y}} A_{\boldsymbol{x},\boldsymbol{y}}\, \psi_{\boldsymbol{y}}. \tag{2.17}$$

For a dense matrix this costs $\mathcal{O}(d^{2N})$ operations.

Sparsity for K-local operators. If $\hat{A}$ is a sum of K-local terms (each term acts non-trivially on at most K sites), its matrix in the computational basis is sparse. For instance, for a two-body term $\hat{h}^{(i,j)}$ acting only on sites i and j, the element $\langle \boldsymbol{x}|\hat{h}^{(i,j)}|\boldsymbol{y}\rangle$ can be non-zero only if

$$x_k = y_k \quad \text{for all } k \neq i, j. \tag{2.18}$$

Thus, for fixed $\boldsymbol{x}$ there are at most d^K configurations $\boldsymbol{y}$ connected by that term (for two-body terms, $K = 2$ so at most d^2 connections), rather than d^N. If $\hat{A}$ contains $\mathcal{O}(N)$ such local terms (e.g. nearest-neighbor interactions on a lattice), a typical application cost scales like

$$\mathcal{O}\big(N \cdot d^K \cdot d^N\big)\,, \tag{2.19}$$

which is still exponential in N due to the state dimension d^N, but dramatically cheaper than the dense $\mathcal{O}(d^{2N})$ scaling.

Example (explicit construction of local operators). To make the abstract discussion concrete, consider again a system of $N = 3$ sites with local dimension $d = 2$ (spin-$\frac{1}{2}$ degrees of freedom). Let $\{|0\rangle\,, |1\rangle\}$ be the computational basis of each site, and define the Pauli matrices

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \qquad \mathbb{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{2.20}$$

One-body operator. A one-body operator acting with σ_x on site i (and as the identity elsewhere) is written as

$$\hat{\sigma}_x^{(i)} = \mathbb{I}^{\otimes(i-1)} \otimes \sigma_x \otimes \mathbb{I}^{\otimes(N-i)}. \tag{2.21}$$

For $N = 3$, the three possible one-body operators are

$$\hat{\sigma}_x^{(1)} = \sigma_x \otimes \mathbb{I} \otimes \mathbb{I}, \tag{2.22}$$

$$\hat{\sigma}_x^{(2)} = \mathbb{I} \otimes \sigma_x \otimes \mathbb{I}, \tag{2.23}$$

$$\hat{\sigma}_x^{(3)} = \mathbb{I} \otimes \mathbb{I} \otimes \sigma_x. \tag{2.24}$$

In NumPy, these operators are constructed explicitly using Kronecker products[6]:

```
import numpy as np

# local operators
sx = np.array([[0, 1],
               [1, 0]], dtype=np.complex128)
sz = np.array([[1, 0],
               [0, -1]], dtype=np.complex128)
I = np.eye(2, dtype=np.complex128)

# one-body operators on N=3 sites
sx_1 = np.kron(np.kron(sx, I), I) # sigma_x on site 1
sx_2 = np.kron(np.kron(I, sx), I) # sigma_x on site 2
sx_3 = np.kron(np.kron(I, I), sx) # sigma_x on site 3
```

Each of these matrices has size $2^3 \times 2^3 = 8 \times 8$ and is extremely sparse.

Two-body operators. Similarly, a two-body interaction such as $\sigma_z\sigma_z$ acting on sites i and j is given by

$$\hat{\sigma}_z^{(i)}\hat{\sigma}_z^{(j)} = \mathbb{I}^{\otimes(i-1)} \otimes \sigma_z \otimes \mathbb{I}^{\otimes(j-i-1)} \otimes \sigma_z \otimes \mathbb{I}^{\otimes(N-j)}. \tag{2.25}$$

For $N = 3$, the interactions between sites $(1, 2)$ and $(1, 3)$ read

$$\hat{\sigma}_z^{(1)}\hat{\sigma}_z^{(2)} = \sigma_z \otimes \sigma_z \otimes \mathbb{I}, \tag{2.26}$$
$$\hat{\sigma}_z^{(1)}\hat{\sigma}_z^{(3)} = \sigma_z \otimes \mathbb{I} \otimes \sigma_z. \tag{2.27}$$

In NumPy:

```
# two-body zz interactions
zz_12 = np.kron(np.kron(sz, sz), I) # sites 1 and 2
zz_13 = np.kron(np.kron(sz, I), sz) # sites 1 and 3
```

Each of these matrices connects only basis states that differ on the specified sites, and leaves all other local indices unchanged, exactly as discussed in the general sparsity argument above.

Remark. While these explicit constructions using `np.kron` are perfectly adequate for small systems and pedagogical purposes, their cost scales exponentially with N. Practical many-body algorithms therefore avoid forming these full $d^N \times d^N$ matrices explicitly, and instead exploit tensor-product structure, sparsity, or implicit operator application.

[6]The code listing to build those operatorsare explicit, but quite ugly as you need to write them by hand. A more general way to embed a 1-body operator in a many body basis is to use a `functools.reduce` operation: `sx_i = reduce(np.kron, [sx if k==i else I for k in range(N)])` , where `N` is the total number of degrees of sites, `i` is the site on which this operator acts and `I` is the $d \times d$ identity matrix. Reduce will recursively calls `np.kron` on the result of the previous call and the next element in the iterator.

Sparse versus dense matrix representations

In the state-vector formalism, applying a Hamiltonian $\hat{H}$ to a state $|\psi\rangle$ amounts to a matrix–vector multiplication

$$\phi_{\boldsymbol{x}} = \sum_{\boldsymbol{y}} H_{\boldsymbol{x},\boldsymbol{y}}\, \psi_{\boldsymbol{y}}.$$

For a dense matrix of size $d^N \times d^N$, this requires storing d^{2N} entries and performing $\mathcal{O}(d^{2N})$ operations.

For physically relevant Hamiltonians, this is highly wasteful. As discussed above, K-local operators connect only basis states that differ on at most K sites. As a result, each row of H contains only $\mathcal{O}(d^K)$ non-zero entries, while the remaining entries are exactly zero. For one-body operators, the fraction of non-zero elements per row is exponentially small in N.

This motivates using **sparse matrix representations**, where zero entries are neither stored nor processed. Sparse storage significantly reduces memory usage and can dramatically accelerate matrix–vector multiplication.

In practice, Python libraries provide direct support for sparse matrices. In `NumPy/SciPy`, a dense matrix can be converted to a sparse representation using, for instance,

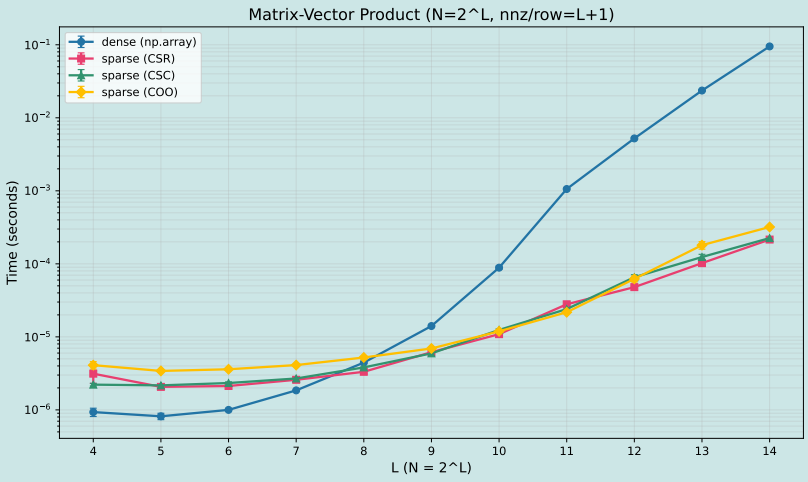
$$H_{\text{sparse}} = \texttt{scipy.sparse.csr\_matrix}(H_{\text{dense}}),$$

which stores the matrix in **Compressed Sparse Row (CSR)** format. CSR is particularly efficient for repeated matrix–vector products, as required in time evolution or Krylov-based algorithms.

Sparse matrices can also be constructed directly. For example, Kronecker products of local operators can be built in sparse form using `scipy.sparse.kron`, which takes sparse inputs and returns a sparse matrix, avoiding the explicit construction of dense intermediates.

Once in sparse form, dedicated routines can be used to perform linear-algebra operations. In particular, SciPy provides sparse eigensolvers (e.g. `scipy.sparse.linalg.eigh` and related Krylov methods) that compute a small number of extremal eigenvalues without ever forming the full dense spectrum.

Sparse representations are not universally optimal. For very small matrices, the overhead associated with indirect memory access and index bookkeeping can outweigh the benefits of sparsity. In this regime, modern hardware often performs better with densely packed arrays due to cache locality and vectorization. Sparse formats become advantageous only once the matrix dimension is sufficiently large and the sparsity is pronounced.



The figure illustrates the crossover between dense and sparse representations for matrix–vector multiplication, highlighting the performance gains achieved by sparse storage as the Hilbert-space dimension grows. Notice that for $N = 14$ the sparse implementation is almost a factor of $\times 1000$ faster than a dense matrix multiplication.

2.2 Time-independent Schroedinger's Equation

One of the most direct approaches to studying quantum many-body systems is *exact diagonalisation* (ED). The starting point is the time-independent Schrödinger equation

$$\hat{H}\,|\psi_n\rangle = E_n\,|\psi_n\rangle\,, \tag{2.28}$$

where $\hat{H}$ is the Hamiltonian of the system, $\{|\psi_n\rangle\}$ are its eigenstates, and $\{E_n\}$ the corresponding energy eigenvalues. Knowledge of the spectrum and eigenstates allows, in principle, the computation of all equilibrium and dynamical properties of the system.

2.2.1 The Gibbs ensemble and the ground-state

In thermal equilibrium at temperature T, the state of a quantum system is described by the Gibbs (thermal) density matrix[7]

$$\rho(T) = \frac{e^{-\beta\hat{H}}}{Z}, \qquad \beta = \frac{1}{k_B T}, \tag{2.29}$$

where $Z = \mathrm{Tr}\left(e^{-\beta\hat{H}}\right)$ is the partition function. In the zero-temperature limit $\beta \to \infty$ (or $T \to 0$), the Gibbs state reduces to[8] a projector onto the ground state $|\psi_0\rangle$ of $\hat{H}$.

More generally, if the thermal energy $k_B T$ is much smaller than the excitation gap above the ground state, thermal occupation of excited states is strongly suppressed. In this regime, many physical observables are well approximated by their ground-state expectation values.

This situation is common in several contexts, such as quantum chemistry, where one is typically interested in electronic ground states, condensed-matter systems cooled at sufficiently low temperature or quantum computers operating in cryogenic environments. For these reasons, a central task of many numerical methods is the accurate determination of the ground state and, in some cases, a few low-lying excited states.

2.2.2 Exact diagonalisation (ED)

In the state-vector representation, the Hamiltonian $\hat{H}$ is represented by a matrix of size $d^N \times d^N$. Exact diagonalisation consists in computing its eigenvalues and eigenvectors by standard linear-algebra techniques.

This approach is conceptually straightforward and numerically exact, but its applicability is severely limited by the exponential growth of the Hilbert-space dimension. In practice, ED is restricted to relatively small systems (typically $N \lesssim 20$–25 for spin-$\frac{1}{2}$ systems, depending on symmetries and available memory). Nevertheless, ED plays a crucial role as a benchmark for more advanced methods and as a tool to gain insight into strongly correlated systems.

2.2.3 The Lanczos algorithm

The Lanczos algorithm is an iterative method to approximate a few extremal eigenvalues and eigenvectors of a large Hermitian operator $\hat{H}$. It belongs to the class of *Krylov subspace methods* and is particularly well suited to sparse many-body Hamiltonians.

---

[7] We will not be proving this, but it's a well-known result from quantum statistical mechanics

[8] Let $\hat{H}\,|\psi_n\rangle = E_n\,|\psi_n\rangle$ with $E_0 \le E_1 \le \cdots$ and expand

$$e^{-\beta\hat{H}} = \sum_n e^{-\beta E_n}\,|\psi_n\rangle\langle\psi_n|\,, \qquad Z = \mathrm{Tr}\left(e^{-\beta\hat{H}}\right) = \sum_n e^{-\beta E_n}.$$

Factor out the lowest energy,

$$\rho(\beta) = \frac{e^{-\beta\hat{H}}}{Z} = \frac{\sum_n e^{-\beta(E_n-E_0)}\,|\psi_n\rangle\langle\psi_n|}{\sum_n e^{-\beta(E_n-E_0)}}.$$

As $\beta \to \infty$, for all $n > 0$ one has $E_n - E_0 > 0$ and thus $e^{-\beta(E_n-E_0)} \to 0$, so $\rho(\beta) \to |\psi_0\rangle\langle\psi_0|$. If the ground state is g-fold degenerate ($E_0 = \cdots = E_{g-1}$), the same argument gives $\rho(\beta) \to \frac{1}{g}\sum_{n=0}^{g-1} |\psi_n\rangle\langle\psi_n|$, i.e. the maximally mixed state on the ground-state manifold.

Krylov subspaces. Given an initial (normalized) vector $|v_0\rangle$ with non-zero overlap with the ground state, the Lanczos algorithm constructs an orthonormal basis of the Krylov subspace by repeatedly applying the Hamiltonian $\hat{H}$ to the initial vector[9],

$$\mathcal{K}_m(\hat{H}, |v_0\rangle) = \text{span}\{|v_0\rangle, \hat{H}|v_0\rangle, \hat{H}^2|v_0\rangle, \ldots, \hat{H}^{m-1}|v_0\rangle\}. \tag{2.30}$$

By construction, this subspace contains increasingly accurate approximations to the extremal eigenstates of $\hat{H}$ as m increases[10].

Lanczos recursion. The algorithm proceeds by building an orthonormal basis $\{|v_0\rangle, |v_1\rangle, \ldots\}$ through a three-term recurrence relation. Starting from $|v_0\rangle$, one defines

$$|w_{n+1}\rangle = \hat{H}|v_n\rangle - \alpha_n|v_n\rangle - \beta_n|v_{n-1}\rangle, \tag{2.31}$$

$$\alpha_n = \langle v_n|\hat{H}|v_n\rangle, \tag{2.32}$$

$$\beta_{n+1} = \| |w_{n+1}\rangle \|, \qquad |v_{n+1}\rangle = \frac{|w_{n+1}\rangle}{\beta_{n+1}}, \tag{2.33}$$

with $\beta_0 = 0$. At each step, the new vector is orthogonal to the two previous ones by construction.

In the Lanczos basis, the Hamiltonian is represented by a real symmetric tridiagonal matrix

$$T_m = \begin{pmatrix} \alpha_0 & \beta_1 & 0 & \cdots & 0 \\ \beta_1 & \alpha_1 & \beta_2 & \ddots & \vdots \\ 0 & \beta_2 & \alpha_2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \beta_{m-1} \\ 0 & \cdots & 0 & \beta_{m-1} & \alpha_{m-1} \end{pmatrix}. \tag{2.34}$$

Approximate eigenvalues and eigenvectors. The eigenvalues of T_m are called *Ritz values*. As m increases, the extremal Ritz values converge rapidly to the extremal eigenvalues of $\hat{H}$. The corresponding eigenvectors of $\hat{H}$ are approximated by linear combinations of the Lanczos basis vectors,

$$\left|\psi^{(m)}\right\rangle = \sum_{n=0}^{m-1} c_n |v_n\rangle, \tag{2.35}$$

where $\boldsymbol{c}$ is an eigenvector of T_m.

In practice, convergence to the ground state is often exponential in m, provided the initial vector has non-zero overlap with it.

Computational cost and advantages. Each Lanczos iteration requires one application of $\hat{H}$ to a vector and a small number of inner products and vector updates. Most importantly, we never need to really store the full, dense matrix H. This means that we can also diagonalise matrices that we never really store in memory, but which we only know how to apply to a vector! The memory cost scales as $\mathcal{O}(m\, d^N)$ to store the Lanczos vectors, while the dominant computational cost is repeated sparse matrix–vector multiplication.

This makes Lanczos a standard tool to diagonalise the large sparse Hamiltonians arising in many-body physics.

Loss of orthogonality and practical variants. For completeness, I will mention that the Lanczos vectors are mutually orthogonal only if we were to perform arithmetic operations exactly with no error. However, in computers we operate with finite-precision arithmetic. For example, imagine the two vectors $\vec{a} = (1.0, 2.0)$ and $\vec{b} = (0.99996, 2.00004)$. The two vectors are clearly not collinear... if we have sufficient digits of decimal precision. However, if we only had 4 digits of decimal precision,

[9] Notice that this is done by repeatedly performing matrix-vector multiplications, which we have shown before can benefit greatly from sparse matrix encodings.

[10] See Gene H. Golub and Charles F. Van Loan, Matrix Computation (2013), Chapter 9.1 in the 3rd edition or chapter 10.1 in the 4th edition.

the truncated version of the latter vector is $\vec{\tilde{b}} = (1.0000, 2.0000)$ and the two vectors become perfectly identical.

While this is a fabricated example, this is ultimately the problem of the finite numerical precision we have available on computers[11] Indeed, it may happen that after a certain number of Lanczos iterations loss of orthogonality occurs, leading to spurious repeated eigenvalues ("ghost states").

When one encounters loss of orthogonality, there exist some strategies to mitigate those issues but in general those are not well motivated theories but rather *recipes*. In practice, modern numerical libraries implement robust restarted versions of Lanczos that balance numerical stability and efficiency.

Scope and limitations. Lanczos-based exact diagonalisation provides essentially exact results for ground states and low-lying excitations of small to intermediate-sized systems. However, the exponential growth of the Hilbert space ultimately limits its applicability. This motivates the development of alternative approaches—such as tensor networks and variational methods—which trade exactness for scalability.

Nevertheless, the Lanczos algorithm remains a cornerstone of numerical many-body physics, both as a computational tool and as a conceptual reference point for more advanced methods. They also serve as the conceptual foundation for more advanced techniques, such as time evolution, Green's-function methods, and finite-temperature extensions.

2.3 Measures of quantum complexity: entanglement entropy

A key difficulty of quantum many-body physics is to quantify *how complex* a given quantum state is. At an intuitive level, we know that *separable* (or product) states are simple: they do not display genuinely quantum correlations and can be described by specifying independent states for each degree of freedom. In contrast, entangled states encode correlations that cannot be reduced to classical probabilistic mixtures and are intrinsically many-body in nature.

Separable states as simple states. A pure state $|\psi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$ is called **separable** if it can be written as

$$|\psi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle \,. \tag{2.36}$$

For such states, expectation values of *product observables* factorize:

$$\langle\psi|\hat{O}_1 \otimes \hat{O}_2|\psi\rangle = \langle\phi_1|\hat{O}_1|\phi_1\rangle \;\; \langle\phi_2|\hat{O}_2|\phi_2\rangle \,. \tag{2.37}$$

As a simple example, consider an Ising Hamiltonian on two sites,

$$\hat{H} = J\,\sigma_1^z\sigma_2^z + h\,(\sigma_1^z + \sigma_2^z). \tag{2.38}$$

On a product state $|\psi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle$ the interaction term reduces to a product of single-site expectation values,

$$\langle\psi|\sigma_1^z\sigma_2^z|\psi\rangle = \langle\phi_1|\sigma^z|\phi_1\rangle \;\; \langle\phi_2|\sigma^z|\phi_2\rangle \,, \tag{2.39}$$

so that the energy can be written solely in terms of local one-point functions:

$$\langle\psi|\hat{H}|\psi\rangle = J\;\langle\phi_1|\sigma^z|\phi_1\rangle\;\langle\phi_2|\sigma^z|\phi_2\rangle + h\Big(\,\langle\phi_1|\sigma^z|\phi_1\rangle + \;\langle\phi_2|\sigma^z|\phi_2\rangle\,\Big). \tag{2.40}$$

This simplification is lost as soon as the state is not separable: correlations no longer factorize, and the reduced state of each subsystem becomes mixed.

[11] By default computers operate on 32-bit (single precision) or 64-bit (double-precision) floating point numbers, which have resspectively ≈ 7 and ≈ 15 digits of decimal precision. More recently, the most modern GPUs used for Machine Learning operate efficiently only with half precision (3 digits). There exist methods to perform calculations with an arbitrary number of digits, called arbitrary-precision numerics, implemented by packages such as `mpmath` in Python, but they are order of magnitude slower than standard single/double precision arithmetic, which benefits from specialised circuitry in the processors.

Beyond a binary notion of entanglement. While separability provides a clear-cut distinction between "classical-like" and "quantum" states, it is a binary notion: a state is either separable or entangled. For many-body systems this is too crude. We would like a quantitative notion that answers the question: *how entangled is a given state?*

This motivates the introduction of **entanglement measures**, which quantify the degree of non-separability between different parts of a system. The most widely used measure for pure states is the *bipartite entanglement entropy*. Do notice that several other measures of *quantum complexity* exists, and it is an open question on how they are related to each other and how to interpret them precisely.

Reduced density matrix and entanglement entropy. Given a pure state $|\psi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$, we define the density matrix

$$\rho = |\psi\rangle \langle\psi| . \tag{2.41}$$

The reduced density matrix of subsystem A is obtained by tracing out subsystem B by means of the **partial-trace** operation Tr_B,

$$\rho_A = \mathrm{Tr}_B \, \rho. \tag{2.42}$$

This formal definition can be better understood as follows: the partitioning of the hilbert space also defines a partitioning of the basis elements $B(\mathcal{H}) = B(\mathcal{H}_A) \otimes B(\mathcal{H}_B)$. We will label elements of each basis as $\left|x_{A/B}\right\rangle \in B(\mathcal{H}_{A/B})$ respectively. Then, the matrix elements of the reduced density matrix of the subsystem A, ρ_A are

$$\langle x_A| \rho_A |y_A\rangle = \sum_{x_B \in B(\mathcal{H}_B)} \langle x_A, x_B| \rho |y_A, x_B\rangle = \sum_{x_B \in B(\mathcal{H}_B)} \langle x_A, x_B|\psi\rangle \langle\psi|y_A, x_B\rangle \tag{2.43}$$

If the state is separable ($|\psi\rangle = |\phi_A\rangle \otimes |\phi_B\rangle$), then the reduced density matrix ρ_A is a rank-one projector and therefore represents a pure state. Substituting a separable state in eq. (2.43) we obtain

$$\begin{aligned}
\langle x_A| \rho_A |y_A\rangle &= \sum_{x_B \in B(\mathcal{H}_B)} \langle x_A, x_B|\phi_A, \phi_B\rangle \langle\phi_A, \phi_B|y_A, x_B\rangle && (2.44)\\
&= \langle x_A|\phi_A\rangle \langle\phi_A|y_A\rangle \sum_{x_B \in B(\mathcal{H}_B)} \langle x_B|\phi_B\rangle \langle\phi_B|x_B\rangle && (2.45)\\
&= \langle x_A|\phi_A\rangle \langle\phi_A|y_A\rangle \sum_{x_B \in B(\mathcal{H}_B)} |\langle x_B|\phi_B\rangle|^2 && (2.46)\\
&= \langle x_A|\phi_A\rangle \langle\phi_A|y_A\rangle && (2.47)
\end{aligned}$$

If the state is entangled, ρ_A is mixed, reflecting the fact that subsystem A is correlated with B.

The **bipartite entanglement entropy** is defined as the von Neumann entropy of the reduced density matrix,

$$S_A = -\,\mathrm{Tr}(\rho_A \log \rho_A) . \tag{2.48}$$

For pure states, one has $S_A = S_B$, so the entanglement entropy is a property of the bipartition rather than of an individual subsystem.

Example: two-site system. Consider a two-qubit system and the family of states

$$|\psi(\theta)\rangle = \cos\theta \; |00\rangle + \sin\theta \; |11\rangle , \qquad \theta \in [0, \tfrac{\pi}{2}]. \tag{2.49}$$

For $\theta = 0$, the state is fully separable, $|\psi\rangle = |00\rangle$. For $\theta = \frac{\pi}{4}$, the state is maximally entangled. Intermediate values of θ interpolate continuously between these two extremes.

Intuitively, we expect the entanglement to grow smoothly as θ increases. Entanglement entropy provides a quantitative way to capture this intuition.

For the two-qubit state $|\psi(\theta)\rangle$, the reduced density matrix of one qubit is:

$$\rho_A = \mathrm{Tr}_B \, |\psi(\theta)\rangle \langle\psi(\theta)| = \begin{pmatrix} \cos^2\theta & 0 \\ 0 & \sin^2\theta \end{pmatrix}, \tag{2.50}$$

which has eigenvalues

$$\lambda_\pm = \cos^2\theta, \; \sin^2\theta, \tag{2.51}$$

leading to an entanglement entropy

$$S(\theta) = -\cos^2\theta \log\cos^2\theta - \sin^2\theta \log\sin^2\theta. \tag{2.52}$$

This entropy vanishes at $\theta = 0$, where the state is separable, and reaches its maximum value $\log 2$ at $\theta = \frac{\pi}{4}$, corresponding to a maximally entangled state. This is very useful: if we know that the entanglement entropy is zero, or almost 0, we have a good argument for performing a mean-field treatment of the system and assume that the wave-function is separable. On the contrary, if the entanglement entropy is large, we know that this approximation would not be motivated.

Entanglement entropy as a measure of complexity. In extended many-body systems, one typically partitions the system into two spatial regions A and B. The entanglement entropy across this bipartition quantifies how strongly the two regions are quantum mechanically correlated. States with low entanglement entropy are, in a precise sense, simple: they can often be efficiently represented using tensor-network methods. Conversely, states with large entanglement entropy are complex and generically require exponentially large resources to describe.

For this reason, entanglement entropy plays a central role as a measure of quantum complexity in many-body physics and provides a crucial link between physical properties of quantum states and their computational representability.

2.4 Example: The Ising model, a paradigmatic many-body problem

A central goal of many-body physics is to understand how collective, macroscopic behaviour emerges from simple microscopic interaction. A paradigmatic model showcasing this behaviour is the *Ising model*, which was originally introduced by Ernst Ising and Wilhelm Lenz in 1924 as a model of ferromagnetism. Despite its apparent simplicity, the Ising model captures essential features of interacting systems—cooperativity, competition, symmetry breaking, and criticality—and has played a foundational role in the development of statistical mechanics, quantum many-body theory, and computational physics.

Definition and physical motivation. In its simplest (classical) form, the Ising model describes a collection of binary variables ("spins") $s_i = \pm 1$ placed on the sites of a lattice and interacting through nearest-neighbour couplings. The Hamiltonian reads

$$H = J \sum_{\langle i,j \rangle} s_i s_j, \tag{2.53}$$

where J is the interaction strength and $\langle i, j\rangle$ denotes pairs of interacting sites. For $J < 0$ the interaction favours alignment, leading to ferromagnetic order at low temperature, while thermal fluctuations disorder the system at high temperature.

Originally introduced as a model of ferromagnetism, the Ising model rapidly transcended this context. Its degrees of freedom need not be interpreted literally as magnetic moments: they can represent occupation variables, lattice distortions, logical bits, or abstract order parameters. As such, the Ising model has found applications ranging from magnetism and alloys to neural networks, optimization, and error-correcting codes.

We remark that at zero-temperature it is trivial to identify the minimum-energy configuration $(s_1, s_2, \ldots, s_N)$ of this classical Hamiltonian. Indeed, it is easy to see that if interacting spins are aligned ($s_i = s_j$) then the energy is $E_{\text{ferro}} = (zN)J$ where zN is the number of interacting pairs of variables[12]. Instead, if we consider ferromagnetic anti-alignment ($s_i = -s_j$) the total energy will be $E_{\text{anti}} = -(zN)J$. As an exercise, it can be easily proven that those are the maximum and minimum energy that the system can have, and therefore if $J > 0$ we have an anti-ferromagnetic ground-state and for $J < 0$ we have a ferro-magnetic ground-state.

[12] The notation stems from z beeing the coordination number of the lattice. For example, $z = 2$ for a 1 Dimensional chain, $z = 4$ for a square lattice or $z = 3$ for a triangular lattice.

Quantum Ising model and dynamics. Promoting the Ising variables to quantum spins leads to the *(transverse-field) quantum Ising model* (**TFIM**), typically written as

$$\hat{H} = J \sum_{\langle i,j \rangle} \sigma_i^z \sigma_j^z - h \sum_i \sigma_i^x, \tag{2.54}$$

where $\sigma_i^{x,z}$ are Pauli matrices acting on site i, and h is a transverse field. The transverse field introduces quantum fluctuations that compete with the interaction term, driving a *quantum phase transition* at zero temperature.

The quantum Ising model is a cornerstone of quantum many-body physics. It provides one of the simplest examples of a system where ground-state properties, excitation spectra, entanglement, and real-time dynamics can be studied in a controlled yet non-trivial setting. In one dimension, the model is exactly solvable via a mapping to free fermions that can be analytically treated, while in higher dimensions it becomes genuinely interacting and computationally challenging. Nevertheless, even the 1-Dimensional problem is a very useful toy problem on which to test numerous computational techniques.

Why study the Ising model in this course. Throughout the first half of this course, the Ising model will serve as a unifying example to illustrate how different many-body representations and numerical techniques operate in practice. Its small local Hilbert space ($d = 2$) makes it ideal for explicit constructions in the state-vector formalism, while its locality and structure allow for efficient Monte Carlo and tensor-network approaches. At the same time, the exponential growth of the Hilbert space with system size ensures that naive methods rapidly become intractable, making the model an excellent motivation for more powerful techniques.

By revisiting the Ising model from multiple perspectives—classical and quantum, static and dynamical, analytical and numerical—we will use it as a concrete guide to develop intuition for many-body complexity and to benchmark the tools that will later be applied to more realistic and less understood systems.

References. Classic and modern treatments of the Ising model and its many-body aspects can be found in:

- R. J. Baxter, *Exactly Solved Models in Statistical Mechanics*, Academic Press.
- S. Sachdev, *Quantum Phase Transitions*, Cambridge University Press.
- J. Cardy, *Scaling and Renormalization in Statistical Physics*, Cambridge University Press.

These references cover, respectively, exact results, quantum extensions, and the broader conceptual framework underlying critical phenomena.

Chapter 3

Symmetries and Hilbert-space reduction

A symmetry is a transformation of a system that *may* change its microscopic description but leaves all physical predictions unchanged. Two configurations related by a symmetry are physically indistinguishable, even though they may look different when written in terms of local degrees of freedom.

A simple example is provided by an antiferromagnetic spin chain. In a perfectly homogeneous system, the staggered patterns

$$\cdots\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\cdots \qquad \text{and} \qquad \cdots\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\cdots$$

represent the same physical state. They differ only by a global spin flip, which does not modify any observable property of the infinite system[1]. The transformation that exchanges them is therefore a symmetry.

The same principle applies much more generally. A crystal is invariant under translations by lattice vectors, a molecule may be invariant under rotations or reflections[2]. In each case there exists a set of transformations that reshuffle the microscopic description while leaving the physics unchanged.

In quantum mechanics this invariance is encoded in the Hamiltonian. If a transformation is a symmetry of the system, then it must leave the Hamiltonian invariant, and therefore commute with it. From a computational point of view, this implies a redundancy in the Hilbert space. Many basis states that appear distinct are in fact related by symmetry and describe equivalent physics. By reorganising the Hilbert space according to these symmetries, we can decompose it into smaller independent sectors, each labelled by a set of symmetry quantum numbers.

This is useful for many reasons: not only it can be used to reduce the complexity of a computation, or to check its correctness, but it can also help us to improve our understanding of quantum mechanical systems.

3.1 Symmetries as operators and as groups

As the proper mathematical definition of an Hamiltonian simmetry is quite involved, and is not relevant for simpler use-cases, I first introduce an hand-waving definition that does not distinguish a simmetry group from its representation, as this is not needed to get an high-level picture understanding. A more precise discussion will follow later.

3.1.1 Hamiltonian simmetry (informal)

When we say that a Hamiltonian has a symmetry G, we usually mean that there exists a *set* of unitary operators

$$G \approx \{U_g\}$$

[1] The inifite system does not have a beginning or an end, so you can only define relative quantities to an arbitrary point and therefore you would not be able to detect any difference

[2] For example, the water molecule has the structure $\mathrm{H-O-H}$. A rotation by 180° around the oxigen atom and perpendicular to the molecular plane exchanges the two hydrogen atoms, but leaves the molecule invariant.

acting on the Hilbert space $\mathcal{H}$, such that

1. $[H, U_g] = 0$, meaning that every transformation U_g leaves the Hamiltonian invariant;

2. the set $\{U_g\}$ is closed under multiplication and contains the identity and inverses, such that $U_g U_{g'} \in \{U_g\}$ and $U_{g^{-1}} \in \{U_g\}$

In other words, the symmetry is *represented* by a whole family of operators that form a group under composition. An example of such a group are the rotational simmetry of, for example, a square table. The table is invariant under rotations around it's center for angles $\{0, \pi/2, \pi, 3/2\pi\}$. Each of those rotations correspond to a different $\{U_0, U_{\pi/2}, U_\pi, U_{3/2\pi}\}$, and it's easy to see that the set is closed under multiplication and includes the identity (U_0).

3.1.2 Hamiltonian simmetry (formal)

The set $\{U_g\}$ however is not the simmetry group itself. Instead, we must distinguish the elements $g \in G$ which form the simmetry group, and the representations of the group's elements U_g on a given space. A group element g is the abstract object that we can't apply to objects. Its representation as an unitary operator, U_g, is instead a concrete object that can transform wavefunctions. This distinction might seem a mathematical detail, but it is rather important because the same simmetry group might have different representations depending on the specific (Hilbert) space considered.

More precisely, a symmetry is a *group*[3] G whose elements $g \in G$ label abstract symmetry operations, such as a translation by one site, a reflection, or a rotation. The elements g of the symmetry group G do not act directly on quantum states. Instead, the group is represented on the Hilbert space by a map

$$R : G \longrightarrow B(\mathcal{H}) \tag{3.1}$$
$$g \longmapsto R(g) \equiv U_g \tag{3.2}$$

where $B(\mathcal{H})$ denotes the set of linear operators on $\mathcal{H}$. For each group element $g \in G$, this map assigns a unitary operator U_g. Evidently, there may exist multiple inequivalent representations of the same group[4]. This map must preserve the group structure, namely

$$U_g U_h = U_{gh}, \qquad U_e = \mathbb{I}, \tag{3.3}$$

where e is the identity element of the group. Such a map is called a *representation* of the group G on the Hilbert space.

The Hamiltonian is symmetric under the group G if

$$[H, U_g] = 0 \qquad \forall g \in G. \tag{3.4}$$

In this case the action of the symmetry group is compatible with the quantum dynamics.

When dealing with a single symmetry, for example only parity or only translation, the distinction between the abstract group element g and its operator representative U_g is often suppressed in practice, and one loosely refers to the operator itself as "the symmetry". However, this distinction becomes essential when several symmetry groups are present and combined, such as lattice translations together with point-group symmetries or spin rotations.

[3] A group is a set closed under multiplication, containing the inverse and identity.

[4] From a computational point of view this is easy to illustrate using structured data types. Consider two (Python) functions `mygroup_dense_representation(g: GroupElement)-> DenseMatrix` and `mygroup_sparse_representation(g: GroupElement)-> SparseMatrix`. Both map the same abstract group element g to an operator acting on a vector space, but one stores it as a dense matrix and the other as a sparse matrix. Although these two operators implement exactly the same symmetry transformation on states, they are distinct objects in memory and obey different numerical and algebraic rules. More in general, other representations acting on even different spaces could be thought of.

3.1.3 Symmetry eigenbases (Abelian symmetries)

We start by considering an *Abelian* symmetry group. The general results also hold for non-abelian groups, but the discussion is considerably more complicated (see section 3.3). In this case, the symmetry is generated by a set of unitary operators $\{\hat{U}_g\}_{g\in G}$ that commute with each other,

$$[\hat{U}_g, \hat{U}_{g'}] = 0 \qquad \forall g, g' \in G. \tag{3.5}$$

Because all symmetry operators commute, the spectral theorem guarantees the existence of an orthonormal basis of $\mathcal{H}$ made of *simultaneous eigenvectors of all symmetry operators*. In practice, it is often sufficient to work with a single generator $\hat{U}$ of the symmetry group. We define the corresponding symmetry eigenbasis by

$$\hat{U}_g \left|\boldsymbol{x}_u\right\rangle = u_g \left|\boldsymbol{x}_u\right\rangle, \qquad \mathcal{B}_U(\mathcal{H}) = \{\boldsymbol{x}_u \quad \forall u \in \mathrm{eig}[U]\}. \tag{3.6}$$

The eigenvalue u_g labels how a basis state transforms under the symmetry and is called a *symmetry quantum number*. Notice that the eigenbasis of U_g does not depend on g, because they all commute and are simultanously diagonalised. States with the same symmetry eigenvalue span a vector subspace of the Hilbert space,

$$\mathcal{H}_u = \mathrm{span}\{|\boldsymbol{x}_u\rangle\}, \tag{3.7}$$

which we call a *symmetry subsector*. The Hilbert space therefore decomposes as a direct sum

$$\mathcal{H} = \bigoplus_u \mathcal{H}_u, \qquad \dim[\mathcal{H}] = \sum_u \dim[\mathcal{H}_u]. \tag{3.8}$$

The canonical basis of each subsector is

$$\mathcal{B}(\mathcal{H}_u) = \{\boldsymbol{x}_u\}, \tag{3.9}$$

and the full symmetry-adapted basis is simply the union of all subsector bases,

$$\mathcal{B}_U(\mathcal{H}) = \bigoplus_u \mathcal{B}(\mathcal{H}_u). \tag{3.10}$$

This decomposition is a decomposition of vector spaces, not of tensor products. Each $\mathcal{H}_u$ still lives inside the same many-body Hilbert space and preserves the original tensor-product structure of sites, spins, or orbitals. We are only reorganising the basis according to symmetry quantum numbers.

3.1.4 Compatibility with the Hamiltonian and block diagonalisation

As we said before, the simmetry of an hamiltonian corresponds to a set of unitary operators that commute with the Hamiltonian

$$[\hat{H}, \hat{U}_g] = 0 \qquad \forall g \in G. \tag{3.11}$$

Once again, thanks to the spectral theorem the Hamiltonian and the unitaries can all be diagonalised together. In particular, this means that the matrix elements between states belonging to different symmetry sectors vanish,

$$\langle \boldsymbol{x}_u | \hat{H} | \boldsymbol{y}_{u'} \rangle = 0 \qquad \text{if } u \neq u'. \tag{3.12}$$

Therefore, in the symmetry eigenbasis $\mathcal{B}_U(\mathcal{H}) = \{|\boldsymbol{x}_u\rangle\}$, the Hamiltonian becomes block diagonal,

$$H = \begin{pmatrix} H^{(u_1)} & 0 & 0 & \cdots \\ 0 & H^{(u_2)} & 0 & \cdots \\ 0 & 0 & H^{(u_3)} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \tag{3.13}$$

where each block is the restriction of the Hamiltonian to a single symmetry subsector,

$$H^{(u)}_{x,y} = \langle \boldsymbol{x}_u | \hat{H} | \boldsymbol{y}_u \rangle, \qquad H^{(u)} \in \mathbb{C}^{\dim[\mathcal{H}_u]\times\dim[\mathcal{H}_u]}. \tag{3.14}$$

We call the restricted Hamiltonian the *Hamiltonian projection in the $u-$simmetry subspace*. I remark that the Hamiltonian projection is strictly smaller than the original. From a computational point of view, this allows us to diagonalise each block $H^{(u)}$ independently, yielding a reduction of the numerical cost. For example, if we have a $2^N \times 2^N$ hamiltonian, diagonalising it has complexity $\mathcal{O}(2^{3N})$. However, if we can partition the Hilbert space into M blocks of equal size $2^N/M \times 2^N/M$, the complxity of computing the full spectrum of the Hamiltonian, by diagonalising the M simmetry subsectors is $\mathcal{O}(2^{3N}/M^2)$.

In addition to the computational gain, the eigenstates of the Hamiltonian can be labelled by a pair (E, u),

$$\hat{H}\,|E,u\rangle = E\,|E,u\rangle\,, \qquad \hat{U}\,|E,u\rangle = u\,|E,u\rangle\,. \tag{3.15}$$

This is useful because (i) it means that the simmetry quantum number u is conserved by quantum dynamics, and (ii) that we know that eigenstates are either degenerate in the simmtry, or they must lie in a specific simmetry subsector, which can be exploited to understand the nature of the ground-state wave-function.

3.2 Examples

3.2.1 Parity symmetry for two qubits

Consider two spins with Hamiltonian

$$H = \sigma_1^x + \sigma_2^x. \tag{3.16}$$

Define the swap operator

$$P\,|s_1, s_2\rangle = |s_2, s_1\rangle\,. \tag{3.17}$$

The operator P by itself is not a group, since $P^2 \neq P$. Instead,

$$P^2 = \mathbb{I}, \tag{3.18}$$

so the set

$$G = \{\mathbb{I}, P\} \tag{3.19}$$

forms a group under composition, with

$$\mathbb{I}P = P\mathbb{I} = P, \qquad PP = \mathbb{I}.$$

This group is isomorphic to the cyclic group $\mathbb{Z}_2$. One checks that $[H, P] = 0$ and that $[H, \mathbb{I}] = 0$, so the Hamiltonian is symmetric under this representation of the $\mathbb{Z}_2$ group. We usually say that the Hamiltonian has a $\mathbb{Z}_2$ simmetry corresponding to the parity simmetry.

The eigenbasis. In the computational basis $\mathcal{B}(\mathcal{H}) = \{|\uparrow\uparrow\rangle\,, |\uparrow\downarrow\rangle\,, |\downarrow\uparrow\rangle\,, |\downarrow\downarrow\rangle\}$, P is not diagonal[5]. As the group is Abelian[6], to identify the simmetric eigenbasis we can simply identify the eigenbasis of P. We form parity eigenstates

$$|\uparrow\downarrow\rangle_\pm = \frac{1}{\sqrt{2}}\big(\,|\uparrow\downarrow\rangle \pm |\downarrow\uparrow\rangle\,\big), \tag{3.20}$$

$$|\uparrow\uparrow\rangle_+ = |\uparrow\uparrow\rangle\,, \qquad |\downarrow\downarrow\rangle_+ = |\downarrow\downarrow\rangle\,, \tag{3.21}$$

for which its easy to check that they are eigenstates of P with eigenvalues

$$P\,|\uparrow\downarrow\rangle_\pm = \frac{1}{\sqrt{2}}\big(P\,|\uparrow\downarrow\rangle \pm P\,|\downarrow\uparrow\rangle\,\big) = \frac{1}{\sqrt{2}}\big(\,|\downarrow\uparrow\rangle \pm |\uparrow\downarrow\rangle\,\big) = \pm\,|\uparrow\downarrow\rangle_\pm\,, \tag{3.22}$$

$$P\,|\uparrow\uparrow\rangle_+ = |\uparrow\uparrow\rangle_+\,, \tag{3.23}$$

$$P\,|\downarrow\downarrow\rangle_+ = |\downarrow\downarrow\rangle_+\,. \tag{3.24}$$

Therefore the spectrum of P is $\{+1, -1\}$ and the Hilbert space decomposes as

$$\mathcal{H} = \mathcal{H}_+ \oplus \mathcal{H}_-, \tag{3.25}$$

where the two sectors are often called the triplet and singlet sectors,

$$\mathcal{H}_+ = \mathrm{span}\{|\uparrow\uparrow\rangle\,, |\uparrow\downarrow\rangle_+\,, |\downarrow\downarrow\rangle\}, \qquad \mathcal{H}_- = \mathrm{span}\{|\uparrow\downarrow\rangle_-\}. \tag{3.26}$$

[5] Proved because $P\,|\uparrow\downarrow\rangle = |\downarrow\uparrow\rangle$.

[6] Trivial to prove that all its elements commute with eachother.

Triplet and singlet Hamiltonians. Using the definitions above, it is possible to write in the triplet basis

$$\mathcal{B}_+ = \{\, |\uparrow\uparrow\rangle\,,\ |\uparrow\downarrow\rangle_+\,,\ |\downarrow\downarrow\rangle \,\},$$

the restricted Hamiltonian, which is

$$H^{(+)} = \begin{pmatrix} \langle\uparrow\uparrow|H|\uparrow\uparrow\rangle & \langle\uparrow\uparrow|H|\uparrow\downarrow\rangle_+ & \langle\uparrow\uparrow|H|\downarrow\downarrow\rangle \\ \langle\uparrow\downarrow_+|H|\uparrow\uparrow\rangle & \langle\uparrow\downarrow_+|H|\uparrow\downarrow\rangle_+ & \langle\uparrow\downarrow_+|H|\downarrow\downarrow\rangle \\ \langle\downarrow\downarrow|H|\uparrow\uparrow\rangle & \langle\downarrow\downarrow|H|\uparrow\downarrow\rangle_+ & \langle\downarrow\downarrow|H|\downarrow\downarrow\rangle \end{pmatrix} = \begin{pmatrix} 0 & \sqrt{2} & 0 \\ \sqrt{2} & 0 & \sqrt{2} \\ 0 & \sqrt{2} & 0 \end{pmatrix}. \tag{3.27}$$

In the singlet basis

$$\mathcal{B}_- = \{\, |\uparrow\downarrow\rangle_- \,\},$$

since $H\,|\uparrow\downarrow\rangle_- = 0$, we have

$$H_- \equiv H\big|_{\mathcal{H}_-} = \big(\,\langle\uparrow\downarrow_-|H|\uparrow\downarrow_-\rangle\big) = \big(0\big)\,. \tag{3.28}$$

Quspin For larger systems the symmetry-adapted basis is constructed algorithmically. For example, in QuSpin one can restrict to a given parity sector as

```
from quspin.basis import spin_basis_1d
basis = spin_basis_1d(L=10, pblock=1)
```

and all operators are built directly in that reduced Hilbert space.

3.2.2 Example: Translation symmetry on a periodic chain

We now consider a symmetry that is ubiquitous in lattice models: *discrete translations*. Let us consider a one–dimensional chain of L sites with periodic boundary conditions. The canonical basis of the Hilbert space is $\mathcal{B}[\mathcal{H}] = \{\boldsymbol{s}\}$ where the basis elements are

$$|\boldsymbol{s}\rangle = |s_1, s_2, \ldots, s_L\rangle\,,$$

where s_i labels the local degree of freedom at site i.

The translation operator T is defined by shifting all sites by one lattice spacing,

$$T\,|s_1, s_2, \ldots, s_L\rangle = |s_L, s_1, s_2, \ldots, s_{L-1}\rangle\,. \tag{3.29}$$

Applying T repeatedly generates[7] the cyclic group

$$G = \{\mathbb{I}, T, T^2, \ldots, T^{L-1}\}, \qquad T^L = \mathbb{I}, \tag{3.30}$$

which is isomorphic to the Abelian group $\mathbb{Z}_L$.

For a translationally invariant Hamiltonian we have

$$[H, T] = 0, \tag{3.31}$$

so we may simultaneously diagonalise H and T. The symmetry quantum number associated to translations is the *crystal momentum*.

Translation eigenstates. The operator T is unitary and satisfies $T^L = \mathbb{I}$, so its eigenvalues must be L-th roots of unity,

$$t_k = e^{ik}, \qquad k = \frac{2\pi n}{L}, \quad n = 0, 1, \ldots, L-1. \tag{3.32}$$

Given any computational basis state $|\boldsymbol{s}\rangle$, we can construct a momentum eigenstate by summing over its orbit under translations,

$$|\boldsymbol{s}_k\rangle = \frac{1}{\sqrt{\mathcal{N}_{\boldsymbol{s}}}} \sum_{r=0}^{L-1} e^{-ikr}\, T^r\,|\boldsymbol{s}\rangle\,, \tag{3.33}$$

[7] We did not discuss it yet, but this means that T is the generator of the group. For discrete and abelian groups this is quite trivial, but for Lie groups the generator is very important to identify correctly.

where $\mathcal{N}_{\boldsymbol{s}}$ is a normalisation factor that accounts for the fact that some configurations may be invariant under a subgroup of translations. One easily checks that

$$T\,|\boldsymbol{s}\rangle_k = e^{ik}\,|\boldsymbol{s}\rangle_k\,. \tag{3.34}$$

States constructed in this way span the momentum subsector $\mathcal{H}_k$. The Hilbert space therefore decomposes as

$$\mathcal{H} = \bigoplus_k \mathcal{H}_k, \tag{3.35}$$

and the Hamiltonian becomes block diagonal in the momentum basis,

$$H = \bigoplus_k H^{(k)}.$$

In practice, one chooses a single representative from each translation orbit (often called a *root state*) and builds the symmetry–adapted basis using eq. (3.33). This leads to a drastic reduction of the Hilbert-space dimension, especially for large L.

Explicit example: three spins. Consider $L = 3$ spins with periodic boundary conditions. The translation operator acts as

$$T\,|s_1 s_2 s_3\rangle = |s_3 s_1 s_2\rangle\,, \qquad T^3 = \mathbb{I},$$

so the allowed crystal momenta are (see eq. (3.32)

$$k = 0,\ \frac{2\pi}{3},\ \frac{4\pi}{3}.$$

Let us construct the symmetry–adapted basis explicitly. Start from the computational basis state $|\uparrow\downarrow\downarrow\rangle$. Its translation orbit is

$$|\uparrow\downarrow\downarrow\rangle \xrightarrow{T} |\downarrow\uparrow\downarrow\rangle \xrightarrow{T} |\downarrow\downarrow\uparrow\rangle \xrightarrow{T} |\uparrow\downarrow\downarrow\rangle\,.$$

From this orbit we build momentum eigenstates like

$$|s_1 s_2 s_3\rangle_k = \frac{1}{\sqrt{3}}\Big(\,|s_1 s_2 s_3\rangle + e^{-ik}\,|s_2 s_3 s_1\rangle + e^{-i2k}\,|s_3 s_1 s_2\rangle\,\Big),$$

and the value of k must be determined by solving the following eigenvalue equation (notice that the same sequence $(s_1 s_2 s_3)$ might correspond to multiple values of k)

$$T\,|s_1 s_2 s_3\rangle_k = e^{ik}\,|s_1 s_2 s_3\rangle_k\,.$$

As an example, we can do this for two possible sequences, such as $\uparrow\downarrow\downarrow$ and $\uparrow\uparrow\downarrow$, which give

$$|\uparrow\downarrow\downarrow\rangle_k = \frac{1}{\sqrt{3}}\Big(\,|\uparrow\downarrow\downarrow\rangle + e^{-ik}\,|\downarrow\uparrow\downarrow\rangle + e^{-i2k}\,|\downarrow\downarrow\uparrow\rangle\,\Big) \qquad \text{valid for} \qquad k \in \{0,\ \frac{2\pi}{3},\ \frac{4\pi}{3}\}$$

$$|\uparrow\uparrow\downarrow\rangle_k = \frac{1}{\sqrt{3}}\Big(\,|\uparrow\uparrow\downarrow\rangle + e^{-ik}\,|\downarrow\uparrow\uparrow\rangle + e^{-i2k}\,|\uparrow\downarrow\uparrow\rangle\,\Big) \qquad \text{valid for} \qquad k \in \{0,\ \frac{2\pi}{3},\ \frac{4\pi}{3}\}$$

Finally, the fully polarised states are already translation invariant:

$$T\,|\uparrow\uparrow\uparrow\rangle = |\uparrow\uparrow\uparrow\rangle\,, \qquad T\,|\downarrow\downarrow\downarrow\rangle = |\downarrow\downarrow\downarrow\rangle\,,$$

so they only appear in the $k = 0$ sector and we write

$$|\uparrow\uparrow\uparrow\rangle_0 = |\uparrow\uparrow\uparrow\rangle\,, \qquad |\downarrow\downarrow\downarrow\rangle_0 = |\downarrow\downarrow\downarrow\rangle\,.$$

The Hilbert space therefore decomposes as

$$\mathcal{H} = \mathcal{H}_{k=0} \oplus \mathcal{H}_{k=2\pi/3} \oplus \mathcal{H}_{k=4\pi/3},$$

with

$$\mathcal{B}(\mathcal{H}_{k=0}) = \{|\uparrow\uparrow\uparrow\rangle_0, |\downarrow\downarrow\downarrow\rangle_0, |\uparrow\downarrow\downarrow\rangle_0, |\uparrow\uparrow\downarrow\rangle_0\}, \tag{3.36}$$

$$\mathcal{B}(\mathcal{H}_{k=2\pi/3}) = \{|\uparrow\downarrow\downarrow\rangle_{2\pi/3}, |\uparrow\uparrow\downarrow\rangle_{2\pi/3}\}, \tag{3.37}$$

$$\mathcal{B}(\mathcal{H}_{k=4\pi/3}) = \{|\uparrow\downarrow\downarrow\rangle_{4\pi/3}, |\uparrow\uparrow\downarrow\rangle_{4\pi/3}\}. \tag{3.38}$$

This explicitly shows how translation symmetry reorganises the $2^3 = 8$ states into 3 momentum subsectors of dimension 4, 2 and 2 respectively.

Quspin To restrict to a k-subsector in quspin one must do the following

```
from quspin.basis import spin_basis_1d
basis = spin_basis_1d(L=10, kblock=0...L)
```

When performing calculations within simmetry subsectors it is vital to be careful and remember that each eigenstate obtained by eigendiagonalisation is defined in the basis of the subsector. So even if a vector represented in $\mathcal{B}(\mathcal{H}_{k=2\pi/3})$ is a vector of 2 elements, we cannot take the overlap with a vector represented in $\mathcal{B}(\mathcal{H}_{k=4\pi/3})$ (which is also 2 dimensional). Indeed, the bases of the two subspaces are orthogonal...

3.3 Non–Abelian symmetries and irreducible representations

When the symmetry group is non–Abelian, the unitary operators $\{U_g\}_{g\in G}$ do not commute and therefore cannot be simultaneously diagonalised. As a result, eq. (3.6) must be generalised.

The correct mathematical objects are now the *irreducible representations* (irreps) of the group. Instead of being labelled by a single eigenvalue u, symmetry sectors are labelled by an irrep λ and an internal index m running over its dimension.

More precisely, the Hilbert space decomposes as

$$\mathcal{H} = \bigoplus_\lambda \left(\mathbb{C}^{d_\lambda} \otimes \mathbb{C}^{n_\lambda}\right), \tag{3.39}$$

where d_λ is the dimension of the irrep λ and n_λ its multiplicity. The symmetry operators act as

$$U_g = \bigoplus_\lambda \left(D^{(\lambda)}(g) \otimes \mathbb{I}_{n_\lambda}\right), \tag{3.40}$$

where $D^{(\lambda)}(g) \in \mathbb{C}^{d_\lambda \times d_\lambda}$ is the matrix representation of g in the irrep λ.

Generalised symmetry eigenproblem. Instead of an eigenvalue equation

$$U_g |\boldsymbol{x}_u\rangle = u_g |\boldsymbol{x}_u\rangle ,$$

one now has

$$U_g |\boldsymbol{x}_{\lambda,m,a}\rangle = \sum_{m'=1}^{d_\lambda} D^{(\lambda)}_{mm'}(g) \; |\boldsymbol{x}_{\lambda,m',a}\rangle , \tag{3.41}$$

where λ labels the irrep, m its internal index, and $a = 1, \ldots, n_\lambda$ labels different copies of the same irrep. Notice that the Hilbert space factors in blocks corresponding to the different irreps.

The Hamiltonian commutes with all U_g, so it preserves λ but acts non–trivially only on the multiplicity space:

$$H = \bigoplus_\lambda \left(\mathbb{I}_{d_\lambda} \otimes H^{(\lambda)}\right). \tag{3.42}$$

This is the non–Abelian analogue of block diagonalisation.

A paradigmatic example is global SU(2) spin symmetry, where λ is the total spin S and $d_\lambda = 2S + 1$. Working in fixed-S sectors dramatically reduces Hilbert-space sizes, but requires explicit use of representation theory.

3.3.1 Example: SU(2) spin symmetry for two spins

Consider two spin–$\frac{1}{2}$ particles with a rotationally invariant Hamiltonian, for example

$$H = \boldsymbol{S}_1 \cdot \boldsymbol{S}_2, \tag{3.43}$$

where $\boldsymbol{S}_i = \frac{1}{2}(\sigma_i^x, \sigma_i^y, \sigma_i^z)$. This Hamiltonian is invariant under global spin rotations,

$$U(\boldsymbol{n}, \theta) = \exp\big(-i\theta\, \boldsymbol{n} \cdot (\boldsymbol{S}_1 + \boldsymbol{S}_2)\big), \tag{3.44}$$

which form a representation of the non–Abelian group SU(2).

Computational basis. In the standard product basis

$$\mathcal{B}(\mathcal{H}) = \{|\uparrow\uparrow\rangle, |\uparrow\downarrow\rangle, |\downarrow\uparrow\rangle, |\downarrow\downarrow\rangle\},$$

the symmetry is completely hidden: the generators

$$S^a = S_1^a + S_2^a, \qquad a = x, y, z,$$

do not commute with each other, and therefore cannot be simultaneously diagonalised.

Irreducible representations. For SU(2), irreducible representations are labelled by the total spin S. For two spins $\frac{1}{2}$ one finds the decomposition

$$\tfrac{1}{2} \otimes \tfrac{1}{2} = 1 \oplus 0, \tag{3.45}$$

corresponding to a triplet ($S = 1$) and a singlet ($S = 0$).

The symmetry–adapted basis is given by

$$|1, 1\rangle = |\uparrow\uparrow\rangle, \tag{3.46}$$
$$|1, 0\rangle = \tfrac{1}{\sqrt{2}}\big(|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle\big), \tag{3.47}$$
$$|1, -1\rangle = |\downarrow\downarrow\rangle, \tag{3.48}$$
$$|0, 0\rangle = \tfrac{1}{\sqrt{2}}\big(|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle\big), \tag{3.49}$$

where $|S, M\rangle$ diagonalises S^2 and S^z,

$$S^2 |S, M\rangle = S(S+1)\, |S, M\rangle, \qquad S^z |S, M\rangle = M\, |S, M\rangle.$$

This basis realises explicitly the non–Abelian decomposition

$$\mathcal{H} = \mathcal{H}_{S=1} \oplus \mathcal{H}_{S=0},$$

where $\dim \mathcal{H}_{S=1} = 3$ and $\dim \mathcal{H}_{S=0} = 1$.

Action of the symmetry. A rotation $U(\boldsymbol{n}, \theta)$ acts inside each multiplet as a $(2S+1) \times (2S+1)$ matrix,

$$U(\boldsymbol{n}, \theta)\, |S, M\rangle = \sum_{M'=-S}^{S} D^{(S)}_{MM'}(\boldsymbol{n}, \theta)\, |S, M'\rangle,$$

which is the concrete form of eq. (3.41).

Block diagonalisation of the Hamiltonian. Because $[H, U(\boldsymbol{n}, \theta)] = 0$ for all rotations, the Hamiltonian cannot mix different S sectors. Indeed one finds

$$H = \boldsymbol{S}_1 \cdot \boldsymbol{S}_2 = \tfrac{1}{2}\big(S^2 - S_1^2 - S_2^2\big),$$

so that

$$H\,|1,M\rangle = \tfrac{1}{4}\,|1,M\rangle\,, \tag{3.50}$$

$$H\,|0,0\rangle = -\tfrac{3}{4}\,|0,0\rangle\,. \tag{3.51}$$

The Hamiltonian is therefore

$$H = \begin{pmatrix} \frac{1}{4}\,\mathbb{I}_{3\times 3} & 0 \\ 0 & -\frac{3}{4} \end{pmatrix},$$

showing explicitly the non–Abelian block structure: a three–dimensional triplet block and a one–dimensional singlet block.

This is the simplest example of how a non–Abelian symmetry reduces a many–body Hamiltonian to independent representation sectors rather than one–dimensional symmetry eigenspaces.

Chapter 4

Tensor Networks and Matrix Product States

The objective of this chapter is to introduce Matrix Product States, a tool to more efficiently represent the wave-function by truncating the Hilbert space in order to keep only states with limited Entanglement Entropy, which we already discussed in section 2.3.

4.1 Tensor, contractions and diagrammatic notation

In this section we will introduce the formalism of Tensor Networks and the diagrammatic language that we will be using throughout the lectures to discuss algorithms built on them.

4.1.1 Scalars, vectors, matrices, tensors

A **scalar** is a single number $a \in \mathbb{R}$ or $a \in \mathbb{C}$. A **vector** $x \in \mathbb{C}^d$ is a collection of scalar components x_i labelled by one index $i \in \{1, \dots, d\}$. A **matrix** $A \in \mathbb{C}^{d_1 \times d_2}$ is a collection of scalar components A_{ij} labelled by two indices $i \in \{1, \dots, d_1\}$, and $j \in \{1, \dots, d_2\}$.

More generally, an **n-th order tensor** (or rank-n tensor)

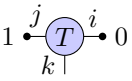
$$T \in \mathbb{C}^{d_1 \times d_2 \times \cdots \times d_n}$$

has scalar components $T_{i_1 i_2 \cdots i_n}$ with indices $i_k \in \{1, \dots, d_k\}$. A scalar is a 0-tensor, a vector is a 1-tensor and a matrix is a 2-tensor.

The *graphical representation* of a order-n tensor is a circle (or sometimes a square) with n legs spreading outwards along an arbitrary direction. Next to each leg, we show the symbol of the corresponding index. Below we give the examples of a scalar a, vector $\vec{x}$, matrix A and 3-tensor T.

$a \implies$  $\quad \vec{x}_i \implies$  $\quad A_{ij} \implies$  $\quad T_{ijk} \implies$  (4.1)

To specify the value of an index to a precise component $i_i \in [1, d_i]$ we simply write the corresponding value. For example, to draw the vector $v_k = T_{0,0,k}$

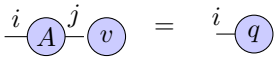
$T_{0,1,k} \implies$  (4.2)

4.1.2 Tensor contraction

The product of a matrix A with a vector v yields a vector q whose components are

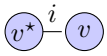
$$q_i = \sum_j A_{ij} v_j. \tag{4.3}$$

This can be written graphically by drawing the A matrix and v vector, and joining the legs with labelled by the same index



(4.4)

The **product of two vectors** (like a bra and a ket) yields a scalar, that is a 'tensor' with 0 legs

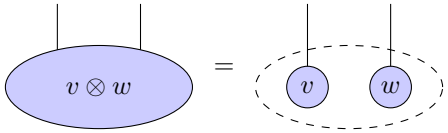


(4.5)

The **trace of a matrix** A yields a scalar, obtained by contracting its row and column indices $\mathrm{Tr}(A) = \sum_i A_{ii}$. Graphically, this corresponds to joining the two legs of the matrix carrying the same index, so that no free indices remain:

(4.6)

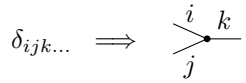
The **kronecker product of two vectors** v, w is a matrix $A = v \otimes w$ with components $A_{ij} = v_i w_j$. As the two components remain separated, graphically they can be represented as two different vectors, and one can dash a line around them to highlight the fact that we think of them as a matrix.



(4.7)

4.1.3 Commonly used tensors

Kronecker tensor An useful tensor is the Kronecker tensor $\delta_{ijk\ldots} = \delta_{ij}\delta_{jk}\delta_{k\ldots}\cdots$ that imposes that all indices are synchronised. We represent this with a point:

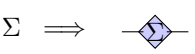


(4.8)

We remark that the special case of a kronecker tensor with only two indices can be dropped and becomes a simple leg

(4.9)

Diagonal Matrices In tensor network diagrams, diagonal matrices are conventionally represented by *squares tilted by* 45°, to visually distinguish them from tensors. Graphically,



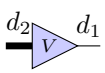
Isometric tensors A central class of tensors used throughout tensor network methods are *isometric tensors*. They represent **isometric linear maps**, i.e. linear transformations that preserve the Euclidean norm of vectors. Geometrically, isometries generalize rotations: they may act within the same vector space, or embed a vector into a higher-dimensional space without changing its length.

A matrix V is said to be *isometric* if it satisfies

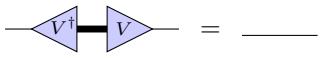
$$V^\dagger V = \mathbb{I}, \qquad \text{or, in index notation,} \qquad \sum_\alpha V^*_{\alpha i} V_{\alpha j} = \delta_{ij}.$$

where $V^\dagger$ denotes the conjugate transpose. In tensor network diagrams, this contraction property is represented graphically by the cancellation of a tensor with its conjugate when contracted along the *output* index.

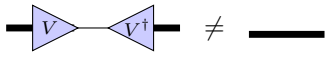
Graphically, we represent an isometric tensor by a *triangle*. The orientation of the triangle encodes the direction of the isometry: the **tip of the triangle always points towards the smaller-dimensional space**. For instance, an isometry mapping a vector space of dimension d_2 to a space of dimension d_1 with $d_2 > d_1$ is drawn as



where the single leg at the tip corresponds to the smaller Hilbert space, and the thicker line hints the fact that that leg has more potential values. The isometric property coresponds to

$$V^\dagger V = \mathbb{I}, \qquad \text{or, in graphical notation,} \qquad$$  (4.10)

which diagrammatically corresponds to the disappearance of the triangle. We remark that the isometric property does not mean that the contraction can be done in the opposite sense, meaning that

$$VV^\dagger \neq \mathbb{I}, \qquad \text{or, in graphical notation,} \qquad$$  (4.11)

Unitary matrices A particularly important special case is when the input and output spaces have the same dimension. In that case, the isometry is invertible and is called a *unitary* matrix,

$$U^\dagger U = UU^\dagger = \mathbb{I}.$$

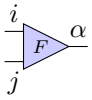
Unitary tensors are **sometimes** graphically represented by **squares**, reflecting the symmetry between their input and output indices.

Reshape (flattening) tensor An important example of an isometric tensor is the *reshape*, or *flattening*, tensor. It implements a change of basis between a multi-index (i, j) and a single composite index $\alpha \equiv (i, j)$. Concretely, it maps a matrix of shape (d_1, d_2) to a vector of size $d_1 d_2$.

We define the flattening tensor $F_{ij\alpha}$, with $i \in [1, d_1]$, $j \in [1, d_2]$ and $\alpha \in [1, d_1 d_2]$, as an isometric tensor satisfying

$$\sum_\alpha F_{ij\alpha} F_{\alpha kl} = \delta_{ik}\delta_{jl}.$$

Graphically, it is represented by a triangle,

$$F_{ij\alpha} \implies$$  (4.12)

where the tip points towards the fused index α, whose dimension is $d_1 d_2$.

Two identical flattening tensors facing each other cancel out, as expected from the isometric property:

$$\sum_\alpha F_{ij\alpha} F_{\alpha kl} = \delta_{ik}\delta_{jl}.$$

This cancellation can be read directly from the corresponding tensor network diagrams.

In practice, reshaped indices are often represented diagrammatically as a *double leg*, indicating a fused or composite index. For example, defining

$$\tilde{T}_{\alpha k\ell m} = \sum_{i,j} F_{ij\alpha}\, T_{ijk\ell m}, \qquad \alpha \equiv (i, j),$$

we may equivalently represent $\tilde{T}$ as a tensor carrying a single composite index α instead of the pair (i, j).

4.1.4 Computational complexity of contractions

A **tensor network** is a collection of tensors with some indices contracted. The number of *free indices* determines whether the network evaluates to a scalar, vector, matrix, or higher-rank tensor.

Evaluating a tensor network corresponds to performing the index contractions (i.e. the summations in the algebraic expression) until a single tensor remains. Importantly, the same tensor network can be evaluated in different orders, corresponding to different orders of summation, which may have very different computational costs. Finding the best contraction order is in principle an NP Hard (factorially hard) problem, and there is no guarantee in general that a contraction will have a polynomial complexity.

Example: Matrix-Matrix-vector product We illustrate this with a simple example involving two matrices and one vector, and the different ways to evaluate this. As you know, the matrix-matrix complexity is $\mathcal{O}(N^3)$ while the matrix-vector product is $\mathcal{O}(N^2)$, so the smartest way to do ABv is to compute it as $A(Bv)$ which performs two matrix-vector products, instead of $(AB)v$.

However, we will try to analise this in the tensor language to understand it in more general terms. Consider the tensor network

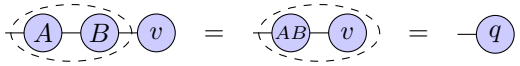
$$q_i = \sum_{j,k} A_{ij} B_{jk} v_k. \tag{4.13}$$

The same contraction can be evaluated in different orders:

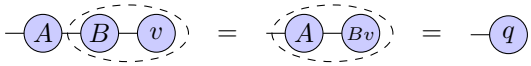
$$\text{(tonto order)} \quad C_{ik} = \sum_j A_{ij} B_{jk}, \qquad q_i = \sum_k C_{ik} v_k, \tag{4.14}$$

$$\text{(smart order)} \quad w_j = \sum_k B_{jk} v_k, \qquad q_i = \sum_j A_{ij} w_j. \tag{4.15}$$

The complexity of a contraction is proportional to the product of all legs involved. So the contraction $C_{ik} = \sum_j A_{ij} B_{jk}$ involves three legs ijk of size N therefore the contraction, as expected, has a complexity $\mathcal{O}(N^3)$.

(tonto order) 

(4.16)

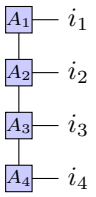
(good order) 

(4.17)

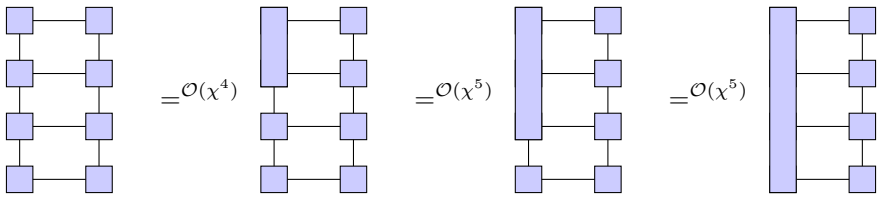
(4.18)

Example: Contracting two Matrix Product States Consider the following tensor network, called a Matrix Product State,

$$\psi_{i_1,i_2,i_3,i_4} = \sum_{k_1,k_2,k_3} A^{(1)}_{i_1 k_1} A^{(2)}_{i_2 k_1 k_2} A^{(3)}_{i_3 k_2 k_3} A^{(4)}_{i_4 k_3} \quad \Rightarrow \tag{4.19}$$

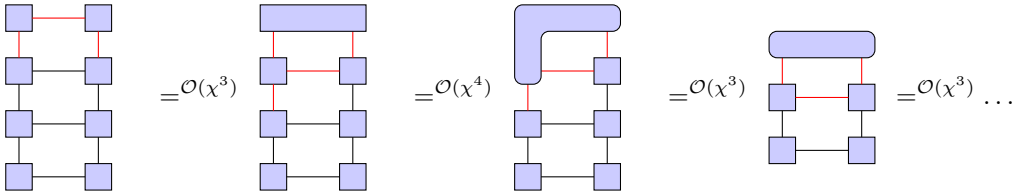


Contracting it vertically yields a complexity of order $\mathcal{O}(\chi^5)$ assuming all legs are of dimension χ.



(4.20)

A better contraction order is to *first* fuse each rung (left–right pair) into a single effective tensor (cost $\mathcal{O}(\chi^4)$ per rung), and *then* contract the resulting 1D chain vertically (cost $\mathcal{O}(\chi^3)$ per step, so overall $\mathcal{O}(\chi^4)$).



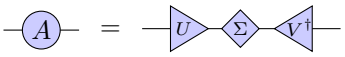
(4.21)

Notice that after contracting the first row eventually the algorithm can be recursively applied and the complexity will remain the same $\mathcal{O}(\chi^4)$ for any number of rows in

4.1.5 Matrix factorizations

Factoring a matrix means identifying some **structured** matrices whose product corresponds to the original matrix. There exist different factorizations of a matrix A, each with different algebraic properties, applicability conditions, and computational costs. The most common ones are:

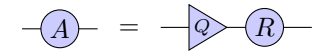
1. **Singular Value Decomposition (SVD)**, which factors

$$A = U\Sigma V^\dagger, \qquad \text{or, diagrammatically,}$$ 

(4.22)

   where U and V are unitary (or orthogonal) matrices and Σ is diagonal with non-negative real entries (the singular values). The SVD exists for any matrix A, including rectangular and rank-deficient ones.

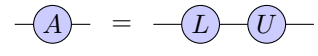
2. **QR decomposition**, which factors

$$A = QR, \qquad \text{or, diagrammatically,}$$ 

(4.23)

   where Q is unitary (or orthogonal) and R is upper triangular. This decomposition is numerically stable and commonly used in least-squares problems and iterative algorithms.

3. **LU decomposition**, which factors

$$A = LU, \qquad \text{or, diagrammatically,}$$ 

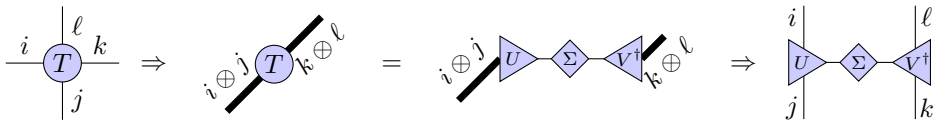
(4.24)

   where L is lower triangular and U is upper triangular (often after a permutation of rows). LU decomposition is mainly used for solving linear systems efficiently.

and we deliberately ignore the diagonalization because it is only applicable to specific matrices[1]. We further remark that while all these decompositions have a computational complexity of order $\mathcal{O}(N^3)$ for $N \times N$ matrices, their prefactor is remarkably different. Indeed, the QR decomposition is typically the cheapest, both in terms of floating-point operations and numerical stability. LU decomposition has a comparable cost but may require pivoting, which slightly increases the overhead. Finally, the SVD is the most costly decomposition but is extremely stable to numerical noise[2]

4.1.6 Tensor Factorizations

By joining together indices (aka, reshaping) it is possible to turn an $n-$tensor into a matrix, which can then be factorized using one of the decompositions discussed above. Of course, those decompositions are not unique and will depend on the particular legs/indices that are merged. For example, consider a rank–4 tensor T_{ijkl}. By reshaping (i.e. merging) the pairs of indices (i, j) and (k, ℓ), the tensor can be viewed as a matrix $T_{(ij)(k\ell)}$:



(4.25)

[1] For completeness, we can mention

- **Diagonalization**, which factors

$$A = PDP^{-1},$$

  where D is a diagonal matrix containing the eigenvalues of A, and P is the matrix of eigenvectors. If A is Hermitian (or real symmetric), P can be chosen unitary (or orthogonal), so that $P^{-1} = P^\dagger$. This factorization exists only if A is diagonalizable.

- **Cholesky decomposition**, which applies only to Hermitian positive-definite matrices and factors

$$A = LL^\dagger,$$

  with L lower triangular. In this restricted but common setting, Cholesky decomposition has a computational cost of $\mathcal{O}(N^3/3)$, making it significantly cheaper than all other factorizations and the method of choice whenever its applicability conditions are satisfied.

[2] Whether you need this stability or the additional information of the SVD decomposition or not, depends on the algorithm. Often, it is not needed to have the full SVD decomposition, and algorithms can be written in terms of QR or LU decompositions!

I remark that there are multiple possible factorizations for the same tensor, depending on how we reshape the tensor into a matrix.

4.2 Matrix Product States as a truncation

The dominant computational cost when simulating a quantum many-body system arises from the exponential growth of the Hilbert space dimension. By reducing the effective size of the Hilbert space, we can push the limits of accessible system sizes. We have already seen that it is possible to partition the Hilbert space into non-interacting subsectors by leveraging symmetries, and to restrict calculations to a single subsector. However, this approach only gives a modest reduction of the Hilbert space size and therefore cannot be used to push calculations much farther.

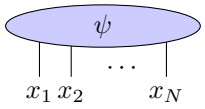
This raises a natural question:

Are there other truncation schemes?

We have already discussed that the most brute-force approximation we can consider in the realm of quantum physics is to restrict the variational space to product states.

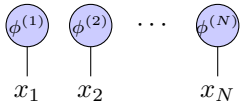
For example, a generic many-body wavefunction

$$|\psi\rangle = \sum_{x_1,\ldots,x_N} \psi_{x_1\ldots x_N} \; |x_1\rangle \otimes \cdots \otimes |x_N\rangle \,, \qquad \text{with} \qquad \psi_{x_1\ldots x_N} \equiv \tag{4.26}$$



is encoded by a complex tensor with d^N coefficients for a local Hilbert space dimension d. In contrast, a fully factorized state takes the form

$$|\psi\rangle = \bigotimes_{i=1}^{N} \left(\sum_{x_i} \phi^{(i)}_{x_i} |x_i\rangle \right), \qquad \text{with} \qquad \phi^{(1)}_{x_1} \cdots \phi^{(N)}_{x_N} \equiv \tag{4.27}$$

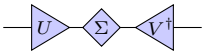


which requires only dN complex parameters (up to normalization). A natural question is: *how can we connect those limits?* If a state is not entangled, we know that we can represent it correctly with a product state. But what about a state that is *barely entangled*? It would be reasonable to expect that with few extra parameters it could still be represented compactly.

4.2.1 Schmidt decomposition

Consider a bipartite system of 2 sub-partitions A and B with local hilbert space dimension d. The wavefunction Ψ can be interpreted as a $d \times d$ matrix, which can be decomposed by an SVD decomposition into the form

$$\Psi_{x_1 x_2} = \sum_{i=1}^{\chi} U_{x_1,i} \Sigma_i V_{x_2,i} \tag{4.28}$$



where the so-called Schmidt rank (or rank of the matrix $\chi \in [1, d]$) determines the total number of parameters of this decomposition.

If $\chi = 1$ it should be easy to see that this form is equivalent to a product state, where

$$\phi^{(1)}_{x_1} = U_{x_1,1} \sqrt{\Sigma_1} \tag{4.29}$$

$$\phi^{(2)}_{x_2} = V_{x_2,1} \sqrt{\Sigma_1}, \tag{4.30}$$

and if $\chi = d$ we can represent an arbitrary state in the Hilbert space. Evidently, if $1 < \chi < d$ this form will be able to represent some states in the Hilbert space, which are neither product states (therefore they carry some entanglement) but are not completely generic either.

In the following, we will show that this χ effectively bounds the amount of bipartite entanglement entropy that the system can support, and therefore can be used to truncate efficiently the Hilbert space.

4.2.2 Bipartite entanglement

We recall from section 2.3 that the bipartite entanglement entropy of a pure state is

$$S_A = -\sum_i \lambda_i^{(A)} \log \lambda_i^{(A)} \tag{4.31}$$

where λ_i are the eigenvalues of the reduced density matrix ρ_A defined as

$$\rho_A = \mathrm{Tr}_B \left[|\psi\rangle \langle\psi| \right]. \tag{4.32}$$

Let's first compute the reduced density matrix. Mathematically, assuming the same decomposition $\Psi_{x_1 x_2} = \sum_{i=1}^{\chi} U_{x_1,i} \Sigma_i V_{x_2,i}$ used previously we get

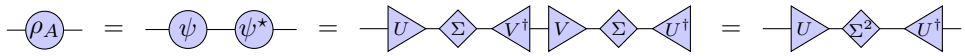
$$\langle x_1 | \hat{\rho}_A | y_1 \rangle = \sum_{x_2} \langle x_1, x_2 | \psi \rangle \langle \psi | y_1, x_2 \rangle = \sum_{x_2} \Psi_{x_1 x_2} \Psi^*_{y_1 x_2} \tag{4.33}$$

$$= \sum_{x_2} \sum_{i,j=1}^{\chi} \left(U_{x_1,i} \Sigma_i V_{x_2,i} \right) \left(U^*_{y_1,j} \Sigma_j V^*_{x_2,j} \right) \tag{4.34}$$

$$= \sum_{i,j=1}^{\chi} U_{x_1,i} \Sigma_i \, U^*_{y_1,j} \Sigma_j \underbrace{\sum_{x_2} V_{x_2,i} V^*_{x_2,j}}_{=\delta_{ij}} \tag{4.35}$$

$$= \sum_{i=1}^{\chi} U_{x_1,i} \, \Sigma_i^2 \, U^*_{y_1,i}. \tag{4.36}$$

Equivalently, in the graphical language



(4.37)

We see that the reduced density matrix takes the form $\rho_A = U \, \Sigma^2 \, U^\dagger$ so that its eigenvalues are simply $\lambda_i^{(A)} = \Sigma_i^2$. These eigenvalues enter directly the von Neumann (Shannon) entropy

$$S_A = -\sum_i \lambda_i^{(A)} \log \lambda_i^{(A)}.$$

If $\chi = 1$, only a single Schmidt coefficient is non-zero, $\Sigma_1 = 1$. In this case ρ_A has rank one, the state is a product state, and the entanglement entropy vanishes, $S_A = 0$.

At the opposite extreme, the entanglement entropy is maximized when all χ Schmidt coefficients are equal,

$$\Sigma_i^2 = \frac{1}{\chi}, \qquad i = 1, \ldots, \chi,$$

corresponding to a maximally mixed reduced density matrix on its support. The maximal bipartite entanglement entropy is then

$$S_A^{\max} = \log \chi.$$

In particular, the Schmidt rank χ provides a strict upper bound on the bipartite entanglement entropy of a pure state.

4.2.3 Area-law for the ground-state of gapped Hamiltonians

By studying[3] some problems like the Transverse Field Ising model you have computed the entanglement entropy of the ground-state wave-function and you have observed in the lab that

- For a system of size L, the maximal bipartite entanglement entropy was found for two bipartitions of size $L/2$.

[3] Hopefully you actually finished the lab assignments.

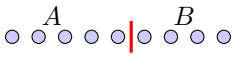
- Each bipartition has local Hilbert space dimension $2^{L/2}$, so the maximal possible entanglement entropy is
$$S_A^{\max} = \frac{L}{2} \log 2.$$
- The actual entanglement entropy of the ground state satisfies
$$S_A(\psi) \ll S_A^{\max}.$$

This suggests that one can take $\chi \ll 2^{L/2}$, and that the state can be faithfully represented using far fewer parameters.

This observation is a manifestation of a more fundamental result, known as an *area law*.

Theorem 1 (Area law in one dimension)**.** *Consider a one-dimensional quantum lattice system with a local, gapped Hamiltonian and ground state $|\psi\rangle$. For any bipartition of the system into two contiguous regions A and B, the bipartite entanglement entropy satisfies*
$$S_A(\psi) \leq C,$$
where C is a constant independent of the system size L. Equivalently, the entanglement entropy scales with the area of the interface *between A and B.*

In one dimension, the interface between two contiguous regions consists of a *finite number of cut bonds* (typically one), and therefore its "area" does not grow with system size. This is illustrated schematically below:



interface (4.38)

Since the size of the interface remains constant as L increases, the entanglement entropy of the ground state remains bounded, in stark contrast with the worst-case scaling of $\propto L$, usually called *volume law scaling*[4].

Higher dimensions. In higher dimensions, the situation is more subtle. While a rigorous proof exists only in special cases, it is widely believed that ground states of local, gapped Hamiltonians satisfy the following generalization:

> *Conjecture (Area law in higher dimensions).* For gapped local Hamiltonians in $d > 1$ dimensions, the entanglement entropy of a region A scales proportionally to the area of the boundary ∂A, i.e.
> $$S_A \propto |\partial A|,$$
> up to possible logarithmic corrections.

This area-law behavior underlies the efficiency of tensor network representations such as matrix product states in one dimension and projected entangled-pair states in higher dimensions.

4.3 Matrix Product State

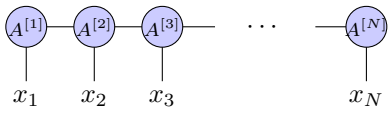
A Matrix Product State (MPS) is a representation of a many-body wave-function in terms of a sequence of local tensors with one physical index and two auxiliary (bond) indices:
$$\Psi_{x_1,x_2,\ldots,x_N} = \sum_{\alpha_1,\ldots,\alpha_{N-1}} A^{[1]}_{x_1,\alpha_1} A^{[2]}_{\alpha_1,x_2,\alpha_2} \cdots A^{[N]}_{\alpha_{N-1},x_N}. \tag{4.39}$$

Here x_i labels the local physical degree of freedom at site i (with dimension d), while the auxiliary indices α_i have dimension at most χ, called the *bond dimension*. The expressive power of the MPS is controlled by χ, which bounds the bipartite entanglement entropy across any cut.

[4] Because it grows with the Volume of the partition, instaed of with it's area.

The number of parameters in the MPS is of the order $\mathcal{O}(Nd\chi^2)$ If $\chi = 1$ this corresponds to a product state, while if $\chi = 2^L$ the MPS can represent an arbitrary wave-function.

Graphically, an MPS is represented as a one-dimensional chain of tensors:

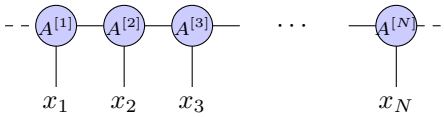


(4.40)

Several remarks are in order.

- **Boundary conditions.** In the expression above we have assumed open boundary conditions, for which the first and last tensors carry only one auxiliary index. Periodic boundary conditions could be accommodated by contracting the first and last auxiliary indices, but then it would be impossible to contract efficiently the tensor network, so MPS are not used to represent periodic systems.

- **Uniform tensor structure.** For notational and algorithmic convenience, it is customary to attach a *virtual bond* of dimension 1 to the left of the first tensor and to the right of the last tensor. With this convention, all tensors in the MPS carry the same number of indices (one physical index and two auxiliary indices), which greatly simplifies both the graphical representation and the implementation of numerical algorithms.



(4.41)

- **Bond dimension and entanglement.** The auxiliary dimension χ is called the *bond dimension*. It controls the amount of bipartite entanglement that the MPS can represent. A consequence of what we discussed in the previous section is that for any bipartition of the chain, the entanglement entropy satisfies

$$S_A \leq \log \chi.$$

  In particular, an MPS with finite χ naturally obeys an area law in one dimension.

- **Exact and approximate representations.** Any quantum state can be written exactly as an MPS provided χ is allowed to grow exponentially with system size. However, ground states of local, gapped one-dimensional Hamiltonians can be accurately approximated by MPS with a *moderate* bond dimension, independent of the system size.

- **Canonical forms.** MPS admit gauge freedoms on their auxiliary indices. By exploiting this freedom, one can bring the MPS into what are called left-canonical, right-canonical, or mixed-canonical form, in which the tensors become isometric and the Schmidt coefficients across a given bond appear explicitly. This structure plays a crucial role in both numerical stability and physical interpretation.

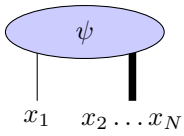
4.3.1 Algorithm: finding the MPS representation of a state

We will now discuss how to convert an arbitrary many-body wave-function into a Matrix Product State. The construction builds upon repeated applications of the singular value decomposition (SVD), and is sometimes referred to as the *successive Schmidt decomposition*.

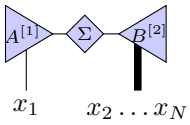
We have already investigated in section 4.2.1 how the Schmidt decomposition can be used to decompose a bipartite system. The algorithm here works by repeatedly considering the bipartitions $(1,)$ and $(2 \dots L)$, then working on the sub-systep $(2 \dots L)$ and splitting into the bipartitions $(2,)$ and $(3 \dots L)$ and so on and so forth.

Algorithm: MPS decomposition of a state vector Consider a system with L subsystems each with local hilbert space dimension d. The wavefunction is a $L-$tensor $\Psi_{x_1 x_2 \ldots x_N}$.

1. **Reshape the wave-function:** We first group the indices $(x_2, \ldots, x_N)$ into a single composite index and interpret the wave-function as a matrix of shape $[d, d^{L-1}]$,

$$\Psi_{x_1,(x_2 \ldots x_N)} \qquad \text{or, graphically,} \qquad \text{  } \tag{4.42}$$

2. **SVD decomposition:** We compute the SVD decomposition of this matrix,

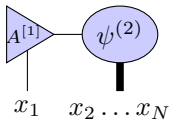
$$\Psi_{x_1,(x_2 \ldots x_N)} = \sum_{\alpha_1} A^{[1]}_{x_1,\alpha_1} \Sigma^{[1]}_{\alpha_1} B^{[2]}_{\alpha_1,(x_2 \ldots x_N)} \text{ or, graphically, } \text{  } \tag{4.43}$$

   The tensor $A^{[1]}$ will become the first MPS tensor, while the remaining degrees of freedom are absorbed into a new effective state.

   The tensors have shape$(A^{[1]}) = [1, d, \chi_1]$, where we have added the *virtual* first dimension 1, $\Sigma^{[1]}$ is a diagonal matrix with χ_1 singular values and $B^{[2]}$ has shape$(B^{[2]}) = [\chi_1, d^{L-1}]$.

   The bond dimension χ_1 is given by the number of non-zero singular values $\Sigma^{[1]}_{\alpha_1}$, and it can be truncated to some value of χ_{max} if desired.

3. **Merge singular values:** The next step is to merge the singular valus into the B tensor to get a single matrix

$$\Psi^2_{\alpha_1,(x_2 \ldots x_N)} = \sum_{\alpha_1} \Sigma^{[1]}_{\alpha_1} B^{[2]}_{\alpha_1,(x_2 \ldots x_N)} \qquad \text{  } \tag{4.44}$$

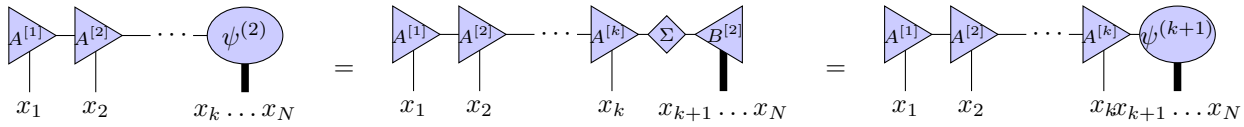
4. **Iterate:** We now repeat the same procedure on the remaining tensor $\psi^{(2)}$. At step k, we reshape the 3-tensor of shape $[\chi_{k-1}, d, 2^{L-k}]$ into the matrix of shape $[\chi_{k-1} d, 2^{L-k}]$

$$\psi^{(k)}_{\alpha_{k-1},x_k,(x_{k+1} \ldots x_N)} \to \psi^{(k)}_{(\alpha_{k-1},x_k),(x_{k+1} \ldots x_N)}$$

   which can then be factored into an SVD,

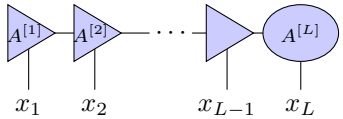
$$\psi^{(k)}_{\alpha_{k-1},x_k,(x_{k+1} \ldots x_N)} = \sum_{\alpha_k} A^{[k]}_{(\alpha_{k-1},x_k),\alpha_k} \Sigma^{[k]}_{\alpha_k} B^{[k+1]}_{\alpha_k,(x_{k+1} \ldots x_N)}.$$

   and then the $A^{[k]}$ matrix is reshaped into a 3-tensor by splitting the first two indices $\to A^{[k]}_{\alpha_{k-1},x_k,\alpha_k}$. Graphically, this corresponds to



(4.45)

After N iterations, all physical indices have been separated, yielding the MPS representation

$$\Psi_{x_1 \ldots x_N} = \sum_{\alpha_1,\ldots,\alpha_{N-1}} A^{[1]}_{x_1,\alpha_1} A^{[2]}_{\alpha_1,x_2,\alpha_2} \cdots A^{[N]}_{\alpha_{N-1},x_N}. \qquad \text{  } \tag{4.46}$$

Notice that this procedure yields all Left-isometric tensors for all tensors except the last one, which is completely unstructured. This form is known as **left-canonical** and is obtained by performing the SVD decompositions left-to-right. Similarly, if we had started from the last degree of freedom and performed the SVD decompositions right-to-left we'd have obtained the **right-canonical** form.

Truncation and efficiency. If the singular values $\Sigma^{[k]}$ decay rapidly, one may truncate them by keeping only the largest χ values at each step. This yields an approximate MPS with controlled error, and explains why ground states of gapped one-dimensional Hamiltonians admit efficient MPS representations.

Entanglement entropy from the MPS construction. An important byproduct of the successive SVD construction is that, at each step k, the singular values $\Sigma^{[k]}$ obtained from the decomposition correspond exactly to the *Schmidt coefficients* associated with the bipartition

$$(1, \ldots, k) \,\big|\, (k+1, \ldots, N).$$

If these singular values are stored (and not truncated), the reduced density matrix of either subsystem is diagonal in the Schmidt basis, with eigenvalues

$$\lambda_i^{(k)} = \big(\Sigma_i^{[k]}\big)^2.$$

As a consequence, the bipartite entanglement entropy across the cut at bond k can be computed directly as

$$S_k = -\sum_i \big(\Sigma_i^{[k]}\big)^2 \log\big(\Sigma_i^{[k]}\big)^2,$$

without the need to explicitly construct reduced density matrices. This observation is central to the efficiency of MPS-based algorithms and provides a direct connection between the bond dimension, the decay of singular values, and the entanglement structure of the quantum state.

4.3.2 Canonical forms of Matrix Product States

A key structural property of Matrix Product States is the presence of a local *gauge freedom* on the auxiliary indices. Indeed, for any invertible matrix X, the transformation

$$A^{[k]} \longrightarrow A^{[k]}X, \qquad A^{[k+1]} \longrightarrow X^{-1}A^{[k+1]}$$

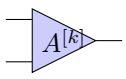
leaves the physical state invariant. Graphically, this corresponds to inserting XX^{-1} on an internal bond, which cancels out.

Left- and right-canonical forms. By exploiting this gauge freedom, an MPS can be brought into special canonical forms. An MPS is said to be in *left-canonical form* if all tensors satisfy

$$\sum_{x_k, \alpha_{k-1}} \big(A^{[k]}_{\alpha_{k-1} x_k \alpha_k}\big)^* A^{[k]}_{\alpha_{k-1} x_k \alpha'_k} = \delta_{\alpha_k \alpha'_k}.$$

That is, each tensor acts as an isometry from the combined indices (α_{k-1}, x_k) to α_k.

Graphically, left-canonical tensors are represented as triangles pointing to the right:



(4.47)

Similarly, an MPS is in *right-canonical form* if

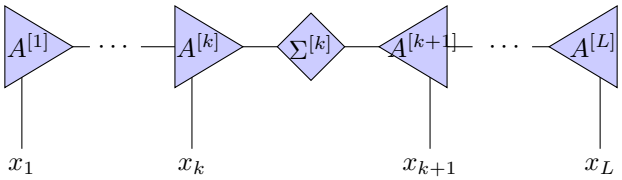
$$\sum_{x_k, \alpha_k} A^{[k]}_{\alpha_{k-1} x_k \alpha_k} \big(A^{[k]}_{\alpha'_{k-1} x_k \alpha_k}\big)^* = \delta_{\alpha_{k-1} \alpha'_{k-1}},$$

corresponding to tensors that are isometric from α_{k-1} to (x_k, α_k). These are represented as triangles pointing to the left.

Mixed-canonical form. Of particular importance is the *mixed-canonical form*. For any chosen bond k, the MPS can be written such that

- all tensors to the left of the bond are left-canonical,
- all tensors to the right are right-canonical,
- the bond between them carries a diagonal matrix of singular values.

Graphically,



(4.48)

The diagonal matrix $\Sigma^{[k]}$ contains the Schmidt coefficients associated with the bipartition

$$(1, \ldots, k) \,\big|\, (k+1, \ldots, N).$$

Physical meaning and consequences. In mixed-canonical form, the entanglement structure of the state becomes explicit: the eigenvalues of the reduced density matrix are simply $\lambda_i^{(k)} = (\Sigma_i^{[k]})^2$. As a consequence,

$$S_k = -\sum_i (\Sigma_i^{[k]})^2 \log\Big(\Sigma_i^{[k]}\Big)^2.$$

Moreover, truncating small singular values provides an optimal low-rank approximation of the state across that bond.

Canonical forms therefore play a central role in both the interpretation of MPS (entanglement, Schmidt spectra) and in numerical algorithms such as DMRG and time-evolution methods, where they ensure numerical stability and local optimality.

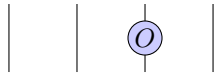
4.4 Expectation values and Matrix Product Operators

Once a state is represented as an MPS, expectation values of observables can be computed efficiently by exploiting the tensor network structure. The key idea is to represent operators themselves as tensor networks, known as *Matrix Product Operators* (MPOs).

Single-site operators. Consider a local operator $\hat{O}^{(k)}$ acting non-trivially only on site k. Therefore, it can be written as

$$\hat{O}^{(k)} = \mathbb{I} \otimes \cdots \otimes \mathbb{I} \otimes \hat{O} \otimes \mathbb{I} \otimes \cdots \otimes \mathbb{I}. \tag{4.49}$$

As all identities correspond to kronecker deltas δ_{x_i, y_i} they diagrammatically are represented as just lines. The single-site operator $\hat{O}$, instead, is just encoded as a $d \times d$ matrix. The diagrammatic representation of a single site operator is therefore

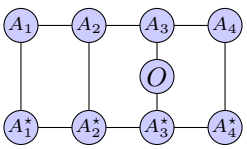


(4.50)

Its expectation value reads

$$\langle \hat{O}^{(k)} \rangle = \langle \psi | \, \mathbb{I} \otimes \cdots \otimes \mathbb{I} \otimes \hat{O}^{(k)} \otimes \mathbb{I} \otimes \cdots \otimes \mathbb{I} \, | \psi \rangle \, .$$

Which graphically corresponds to

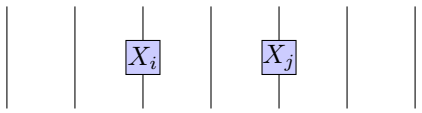


(4.51)

Two-site operators. The single-site picture straightforwardly generalizes to two-site operators obtained by taking the tensor product of two local terms

$$\hat{X}_i\hat{X}_j = \mathbb{I}\otimes\cdots\hat{X}\otimes\mathbb{I}\otimes\cdots\otimes\hat{X}\otimes\mathbb{I}\otimes\cdots\otimes\mathbb{I}, \tag{4.52}$$

which diagramttically is represented as

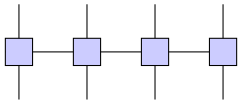


(4.53)

Matrix Product Operators. More generally, any operator acting on the full Hilbert space can be written as an MPO,

$$\hat{O} = \sum_{\{\alpha\}} W^{[1]}_{\alpha_1} W^{[2]}_{\alpha_1,\alpha_2}\cdots W^{[N]}_{\alpha_{N-1}},$$

where each local tensor $W^{[k]}$ carries two physical indices (one for the input and one for the output space) and two auxiliary indices. Graphically, an MPO is represented as a chain of tensors with two vertical legs per site.



(4.54)

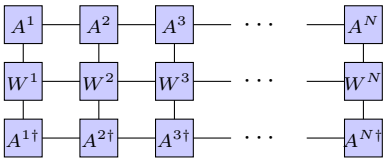
4.4.1 Algorithm: contracting MPS–MPO–MPS networks

We now describe how to efficiently compute expectation values of the form

$$\langle\psi|\hat{O}|\psi\rangle,$$

where $|\psi\rangle$ is represented as an MPS and $\hat{O}$ as an MPO. A naive contraction of all indices would scale exponentially with system size. Instead, the one-dimensional structure of the MPS allows for an efficient contraction by building *environment tensors* and sweeping along the chain.

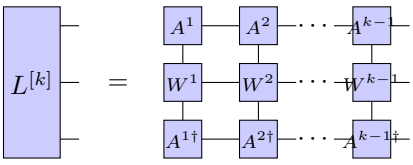
The state $|\psi\rangle$ is encoded into an MPS with 3-tensors $\{A^k\}_k$ and the operator is encoded into an MPO with 4-tensors $\{W^k\}_k$. The tensor network that must be fully contracted is



(4.55)

While this can be *brutally* contracted with `np.einsum` and summing over all indices, this would be brutally inefficient with a complexity of the order of $\mathcal{O}(d^{2L}\chi^{2L})$. However, it is possible to identify a recursive contraction order that is efficient. This will be based on contracting the network left to right, column by column. To simplify the discussion later, we introduce the **Left environment** tensor.

Left environment. We define the *left environment* $L^{[k]}$ as the 3-tensor arising from the partial contraction of all tensors to the left of site k,

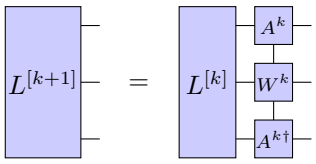


(4.56)

The left environment is a 3-tensor with shape $(\chi_k, \chi_{MPO}, \chi_k)$, and it is built iteratively by contracting a new column with the previous environment $L^{[k-1]}$. By convention, we define that $L^{[1]} = 1$ is the 3-tensor of shape $[1, 1, 1]$.

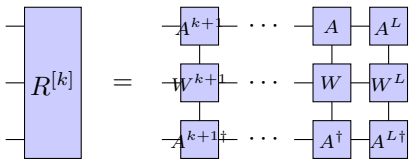
The iterative formula to incorporates one site at a time is

$$L^{[k+1]} = \text{contract}\big(L^{[k]}, A^{[k]}, W^{[k]}, (A^{[k]})^\dagger\big). \qquad (4.57)$$



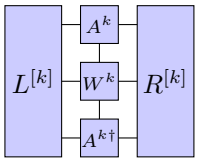
which can be implemented with complexity $\mathcal{O}(\chi^4)$

Right environment. Similarly, one defines the *right environment* $R^{[k]}$ as the contraction of all tensors to the right of site k. It can be built by sweeping from right to left, starting from $R^{[L]} = 1$.



(4.58)

Expectation value. Once both environments are available, the full expectation value is obtained by contracting the left and right environment with the intermediate matrices



(4.59)

for any site k. In practice, one usually performs a full left-to-right sweep, accumulating the contraction and finally contracting with $R^{[L]} = 1$. The algorithm is essentially the following:

1. Initialize the initial 3-tensor of shape (1,1,1) $L^{[1]} = 1$, and the same 3-tensor $R^{[L]} = 1$.
2. For $k = 1, \ldots, N$, update the left environment $L^{[k+1]}$ according to eq. (4.57)
3. To obtain $\langle\psi|\hat{O}|\psi\rangle$, contract L^{L+1} with R^L.

Computational cost. The cost of each local contraction scales as

$$\mathcal{O}(\chi^3 \chi_{\text{MPO}}),$$

where χ is the MPS bond dimension and χ_{MPO} the MPO bond dimension. As a result, expectation values can be evaluated efficiently even for large systems, provided the bond dimensions remain moderate.

This left/right environment construction is the basic building block underlying expectation values, correlation functions, and variational optimization algorithms such as DMRG.

4.5 Common Hamiltonians as MPOs

Chapter 5

Matrix Product States algorithms

In this chapter we will discuss some algorithms relying on MPS.

5.1 Overview of variational methods

In computational physics there exist a large family of algorithms, known as *variational algorithms* that rely on a *variational ansatz*[1] to encode the quantum state [NW06]. An ansatz is a guess (physically motivated or not) on the form of the wave-function with some free parameters. Mathematically, it corresponds to a parametrised wavefunction where the *variational parameters* are the parameters which, when changed, modify the wave-function. Symbolically, the variational ansatz Ψ can be seen as a map that associates to a set of P parameters θ a wavefunction $|\psi_\theta\rangle$. The parameter space will be denoted by $\mathcal{W}$: it has (real) dimension P, and in practice it is most often simply $\mathbb{R}^P$ or $\mathbb{C}^P$. We write

$$\Psi : \mathcal{W} \to \mathcal{H} \tag{5.1}$$
$$\theta \to |\psi_\theta\rangle\,. \tag{5.2}$$

There exist several variational ansatz for the wave-function, and we've seen some already: the mean field wavefunction is a variational ansatz, where we assume that the state must be a product state, or a matrix product state, where we assume that the bipartite entanglement must be less than a maximum value [Sch11, Or 14, CPGSV21].

A variational ansatz defines a *variational manifold*: it is simply the set of all states you can actually reach by tuning the parameters,

$$\mathcal{M}_\Psi \;:=\; \mathrm{Im}(\Psi) \;=\; \{\, |\psi_\theta\rangle \in \mathcal{H} \mid \theta \in \mathcal{W} \,\}.$$

In other words, $\mathcal{M}_\Psi$ is the *search space* of the variational method: regardless of the optimization strategy, the algorithm can only explore states inside this set.

A related (somewhat informal) concept is the *expressivity* of an ansatz. Intuitively, this refers to how "large" or "rich" the manifold $\mathcal{M}_\Psi$ is—loosely, one may think of it as an effective dimension of the family of representable states. Increasing the number of parameters, the bond dimension, or the model architecture typically increases expressivity.

Within this framework, ground-state search is posed as energy minimization restricted to the manifold,

$$E_0^{(\Psi)} \;:=\; \min_{|\psi\rangle \in \mathcal{M}_\Psi} \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} \;=\; \min_{\theta \in \mathcal{W}} \frac{\langle\psi_\theta|H|\psi_\theta\rangle}{\langle\psi_\theta|\psi_\theta\rangle}. \tag{5.3}$$

This is well motivated by the *(ground-state) variational principle*: for any set of parameters θ the variational energy $E(\theta)$ respects,

$$\frac{\langle\psi_\theta|H|\psi_\theta\rangle}{\langle\psi_\theta|\psi_\theta\rangle} = E_0(\theta) \geq E_0,$$

[1] **Ansatz**, *an assumption about the form of an unknown function which is made in order to facilitate solution of an equation or other problem. Derives from german.*

where E_0 is the exact ground-state energy (lowest eigenvalue) of H[2].

If the manifold is sufficiently expressive, the minimum provides a good approximation to the true ground state; otherwise, it yields the best approximation *within the chosen ansatz*.

Connection to Machine Learning: Before continuing in our discussion about Tensor Networks, I would like to stress that within this picture we have recast the problem of finding a ground-state into an optimization problem for a set of variational parameters. Indeed, when physicists realised this (c.a. 2017), it became obvious that ground-state search could be recast as a machine learning problem. But, before we get there, let's discuss an older algorithm: DMRG.

5.2 Density Matrix Renormalization Group (DMRG)

DMRG is a state-of-the-art algorithm to find ground-states of local 1-Dimensional Hamiltonians.

5.2.1 Overview

By now it should be clear that exact diagonalization rapidly becomes hopeless as the Hilbert space dimension grows exponentially with the system size. In the late 1970s and 1980s, several "real-space renormalization" ideas were explored to cope with this. The rough strategy was always the same: solve a small system exactly, keep only a limited number of states to represent it, then generalise the basis states of the small system to a larger one, and repeat.

The issue was that the naive truncation criterion considered at the beginning, such as *keeping the lowest-energy eigenstates of the small block* was usually the wrong one. In an interacting many-body system, what matters is not how a block behaves in isolation, but how it is *entangled with the rest of the chain*. This mismatch is precisely why early real-space renormalization-group schemes (and straightforward lattice adaptations of NRG) often performed poorly.

Stephen White's key insight, introduced in his original DMRG papers [Whi92, Whi93], was to choose the truncated basis using a reduced density matrix. One embeds the block in a larger "superblock" (block + environment), targets a state of interest (typically the ground state), and forms the reduced density matrix of the block. The eigenstates that are kept are not those that minimise the energy of the block or superblock, but rather those that have smaller entanglement entropy between the small system and the rest (environment). The resulting algorithm is remarkably stable and accurate in 1D, and it is the reason DMRG became the workhorse method for strongly correlated chains [Sch05, Sch11].

From the modern tensor-network viewpoint, DMRG is a *variational optimization over the manifold of MPS*. Its efficiency is tied to the fact that many 1D ground states have limited entanglement (often obeying an area law), and can therefore be represented well with moderate bond dimension.

Definition

We consider a nearest-neighbour Hamiltonian written as a Matrix Product Operator,

$$H = \sum_i h_{i,i+1},$$

and an MPS ansatz

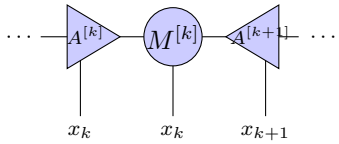
$$|\psi[A]\rangle = \sum_{\{s_i\}} A_1^{s_1} A_2^{s_2} \cdots A_L^{s_L} \, |s_1 s_2 \cdots s_L\rangle \, .$$

The central idea of DMRG is to *optimize the energy by updating one (or two) tensors at a time, while keeping the rest fixed.*

[2] Proof sketch: write $H|n\rangle = E_n|n\rangle$ with $E_0 \leq E_1 \leq \cdots$ and expand $|\phi\rangle = \sum_n c_n |n\rangle$. Then $\langle\phi|H|\phi\rangle / \langle\phi|\phi\rangle = \sum_n |c_n|^2 E_n / \sum_n |c_n|^2 \geq E_0$, since all weights $|c_n|^2$ are non-negative.

Local variational principle

Assume the MPS is brought into a mixed form, centered on site k, with all tensors left-normalized to the left, and right-normalized to the right, and an arbitrary tensor in the middle. Diagrammatically,



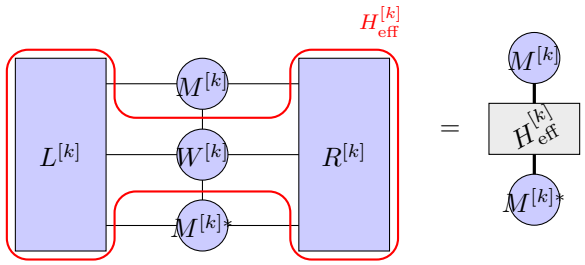
Notice that because of the mixed canonical form, all tensors except $M^{[k]}$ cancel in the norm, so that

$$\langle\psi|\psi\rangle = \langle M^{[k]}|M^{[k]}\rangle = \sum_{\alpha_k,\alpha_{k+1},s_k} \left|M^{[k]}_{\alpha_k,s_k,\alpha_k+1}\right|^2.$$

Meaning that the normalization of the state is only determined by the norm of the vectorized tensor $M^{[k]}$. Similarly, we can rewrite the energy expectation value so that all tensors other than $M^{[k]}$ are contracted out, leaving an *effective* Hamiltonian acting on the local degrees of freedom of site k,

$$\langle\psi|H|\psi\rangle = \langle M^{[k]}|H^{[k]}_{\text{eff}}|M^{[k]}\rangle.$$

Diagrammatically, the effective hamiltonian $H^{[k]}_{\text{eff}}$ can be identified from the environment-contraction diagrams introduced for expectation values (see eq. (4.57)).



(5.4)

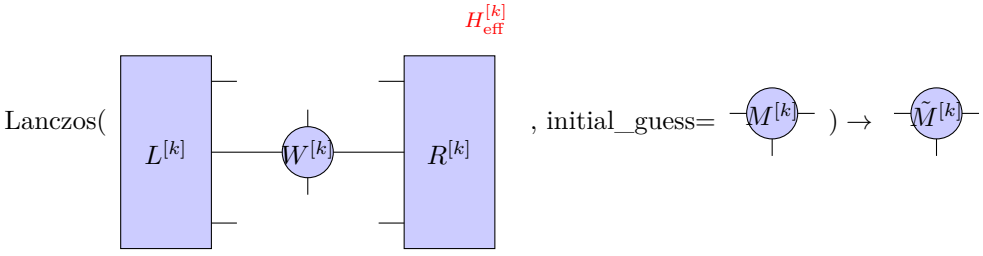
where we have contracted together everything except $M^{[k]}$. Notice that the size of the reduced hamiltonian is $d\chi_k^2 \times d\chi_k^2$ (if it's unclear why, look at the left hand side of eq. (5.4), and notice that the horizontal bonds of M^k are of size χ_k while the vertical bonds are of size d, the local Hilbert space dimension.).

The best values of $M^{[k]}$ are defined as those that minimize the Energy (or, in this setting, local Rayleigh quotient)

$$\min_{M^{[k]}} \frac{\langle M|H_{\text{eff}}|M\rangle}{\langle M|M\rangle}.$$

Notice that, differently from eq. (5.3), the quantity to minimise here does not depend on all variational parameters of the MPS $A^{[1]}, A^{[2]}, \ldots$, but only on a subset.

This leads to a sparse eigenvalue problem whose dimension scales as dD^2, where d is the local Hilbert space dimension and D the bond dimension. In practice, this eigenproblem is solved iteratively using Lanczos, never forming H_{eff} explicitly, and instead just repeatedly applying H_{eff} to the Lanczos vectors. This procedure does not necessarily require to know $M^{[k]}$ to find the optimal one, but will converge faster if a reasonable guess is provided. In practice, we write this step as

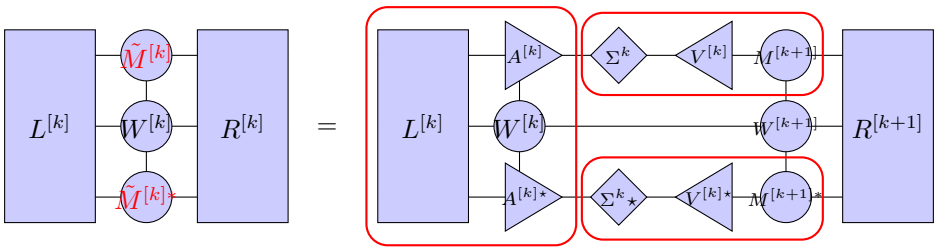


(5.5)

Sweeps and convergence

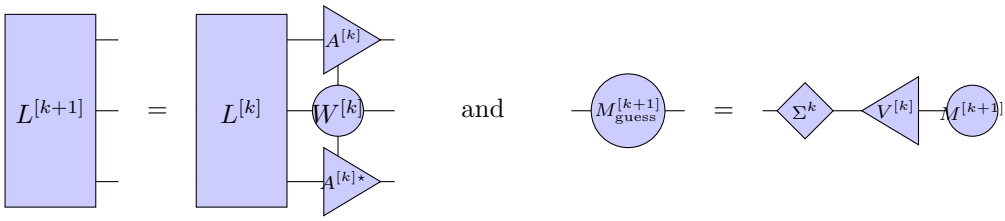
Once the optimal $M^{[k]}$ is found, it is factorized using an SVD and absorbed into neighboring tensors, shifting the orthogonality center to the left or right (depending on where we merge the singular values), and allowing us to repeat the procedure in this direction.

In diagrammatic language, assuming we found $\tilde{M}^{[k]}$ with Lanczos, we can perform the decomposition



(5.6)

and we can highlight that the left-most block corresponds exactly to $L^{[k+1]}$ constructed with the *optimized* local tensors, while on the right I will replace $R^{[k]}$ with $R^{[k+1]}$ with the old tensors. I can also collect the $M^{[k+1]}_{guess} = \Sigma^{[k]} V^{[k]} M^{[k+1]}$ to obtain the initial guess for the next Lanczos iteration



(5.7)

The procedure is then repeated site by site, until we reach the last site. We call a left-to-right **sweep** the cycle of starting from $L^{[1]} = 1$ and getting to the last $L^{[k]}$. We remark that to perform a left-to-right sweep one can precompute all $R^{[k]}$ tensors, but does not need to know the $L^{[k]}$ tnsors in advance because at each iteration of the sweep we compute the next L environment. At the end of the sweep, we will have all the Left environments, and we will be able to start a **right-to-left sweep** which is the same procedure, mirrored. At the end of a **full sweep**, which includes a left to right and a right to left sweep, we will have again a set of Right environment tensors $R^{[k]}$ updated with the new parameters, while the left environment tesors will be *trash*, in the sense that they do not include the most recent parameters.

If we start from a random set of initial tensors, a single sweep already produces a dramatic energy improvement. Repeated sweeps converge extremely reliably to the ground state, provided the bond dimension is large enough.

Most importantly, we can monitor the energy at each iteration of the sweep and see how it goes down. We continue sweeping until the energy is reasonably converged and does not change anymore.

Bond dimension As we mentioned previously, DMRG is a variational method and can only search for optimal states within its variational manifold, which is fixed by the maximal bond dimension χ of the MPS. Nevertheless, it is possible for the ground-state to have a larger entanglement entropy than what we considered, and therefore to be well represented we'd need a larger χ.

To be sure that our result is not biased by an ill choice of bond dimension χ, it is customary to repeat the same DMRG calculation for multiple values of χ until we observe convergence in the bond dimension. A particularly refined type of calculation is usually to compute multiple $E_0(\chi)$ and extrapolate the limit for $\chi \to \infty$ assuming that it must follow a power law.

A point of view from optimization From a modern perspective, DMRG corresponds to coordinate descent on a smooth variational manifold. Its success is therefore deeply tied to the geometry of MPS (which is an exponential form) and their entanglement structure.

Most importantly, we remark that because we are solving each local variational problem exactly, at each step of the sweeps we are always guaranteed to have the best **local** choice of parameters, but we are not guaranteed to have the best **global** choice of parameters. However, due to the structure of

the MPS, which is multi-linear, it would be possible to show that DMRG is to first order in an update as good as gradient descent.

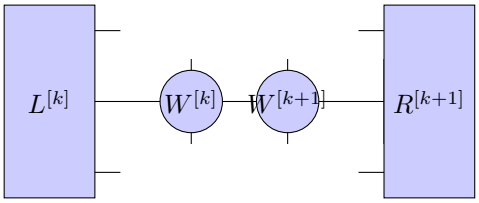
5.2.2 Two-site DMRG (DMRG-2) and adaptive bond dimension

The bond dimension χ controls how much entanglement an MPS can represent. In one-site DMRG, χ is fixed throughout the run; if it is too small, the algorithm can converge to a state that is variationally optimal only within an overly restricted manifold. A standard workflow is then to repeat the full calculation for increasing values of χ.

DMRG-2 addresses this by temporarily enlarging the local variational space. One optimizes two neighboring sites at once, so that the bond connecting them can (temporarily) accommodate more Schmidt states. After the update, an SVD decides which Schmidt states are important: small singular values are discarded, and the bond dimension is increased only where needed.

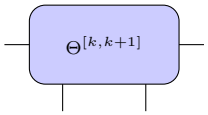
Algorithm. At a given bond $(k, k+1)$:

1. Form the two-site effective Hamiltonian $H_{\text{eff}}^{[k,k+1]}$, slightly generalizing the construction of 1-DMRG as follows,

$H_{\text{eff}}^{[k,k+1]} =$ 

(5.8)

2. Solve the local variational problem defined by the 2-site effective Hamiltonian using Lanczos, finding the optimal 2-site state $\Theta^{[k,k+1]}$, encoded as a 4-tensor,

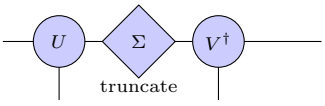


(5.9)

3. Reshape $\Theta^{[k,k+1]}$ into a matrix by grouping indices as (α_{k-1}, s_k) and (s_{k+1}, α_{k+1}) and compute an SVD

$$\Theta^{[k,k+1]} = U\,\Sigma\,V^\dagger.$$

The new bond dimension is chosen either by a hard cap $\chi_{\max}$ or by a truncation tolerance.



At the beginning of the DMRG optimization we can start with a $\chi = 1$ MPS, and gradually this procedure will increase the bond dimension, either until converging, or until hitting the $\chi_{\max}$.

It is useful to monitor during the optimization the discarded weight

$$\epsilon_{\text{trunc}}^{[k]} := \sum_{i>\chi} \Sigma_i^2.$$

which tells us how much at each iteration we are discarding because of a finite bond dimension. Most importantly, this quantity is not accessible to 1-DMRG because it solves a 1-site problem instead of the 2-site problem!

5.3 Real-time dynamics and TEBD

While DMRG excels at computing ground states, many physical questions involve *dynamics*: quenches, transport, response functions, or entanglement growth in time.

The direct time evolution

$$|\psi(t)\rangle = e^{-iHt}\,|\psi(0)\rangle$$

is, however, just as problematic as exact diagonalization. The key idea behind TEBD (Time-Evolving Block Decimation) [Vid03, Vid04] is to exploit both the locality of the Hamiltonian and the efficient structure of MPS.

The unitary evolution operator $U(t) = e^{-iHt}$ is a $2N-$tensor, which could be cast into MPO form. While the Hamiltonian is a low-bond dimension MPO, the unitary evolution operator, at least in principle, has an exponentially large bond dimension and therefore cannot be handled easily. An easy way to see this is to think about a taylor expansion of the exponential which yields $U(t) \approx \sum_n (-iHt)^n/n!$.

The TEBD algorithm tries to find an approximation/expansion of the unitary evolution operator which preserves the locality structure and therefore is more amenable to MPO representation.

Trotter decomposition

We split the Hamiltonian into two non-commuting parts,

$$H = H_{\text{even}} + H_{\text{odd}},$$

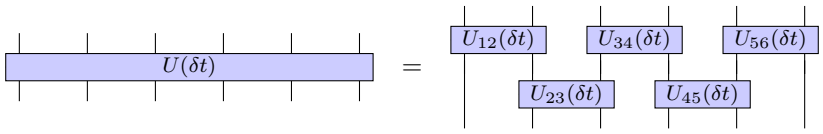
where each part is a sum of commuting two-site terms. For a small time step δt, the evolution operator can be approximated as

$$e^{-iH\delta t} \approx e^{-iH_{\text{even}}\delta t} e^{-iH_{\text{odd}}\delta t} + \mathcal{O}(\delta t^2),$$

or with higher-order Suzuki–Trotter formulas if desired. As the even and odd hamiltonians are composed of commuting terms, they are the product of commuting terms acting on only two sites

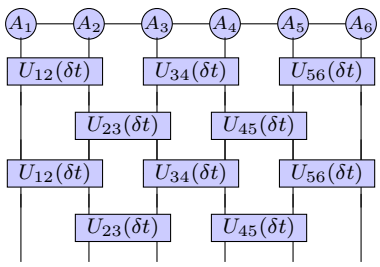
$$e^{-iH_{\text{odd}}\delta t} = \prod_{i=1}^{L-2} e^{-ih^{[i,i+1]}t}. \tag{5.10}$$

Each $e^{-ih^{[i,i+1]}t}$ is a *small* 4-tensor of dimension (d,d,d,d) which can be computed exactly. In diagrammatic notation, this decomposition of the single-timestep Unitary operator corresponds to



(5.11)

By then stacking this operator repeatedly and contracting it with an MPS, we can effectively represent the time-evolved wavefunction



(5.12)

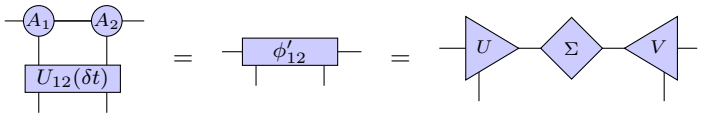
where, in this particular case, the horizontal axis represent space and the vertical axis represents time[3].

[3]Notice that this representation is essentially identical to the one used in quantum computing to represent the operations made on some qubits. Every vertical line corresponds to a qubit, initially initialised as a product states, and operations are small tensors acting on the respective qubits.

Local update and truncation

To compute the time-evolved state one must be able to contract the tensor network shown above and obtain an MPS with some new, generally larger tensors. This can be broken down into applying two-site unitaries to neighboring sites of the MPS.

The resulting contraction, as shown below, produces an effective two-site tensor, which can be decomposed back into MPS form by means of a singular value decomposition (SVD):



(5.13)

The tensor ϕ'_{12} carries two physical indices of dimension d and two virtual indices of dimension χ. To restore the MPS structure, one reshapes it into a matrix by grouping the left virtual index with the first physical index, and the second physical index with the right virtual index. This yields a matrix of size $(\chi d) \times (\chi d)$, whose singular value decomposition defines the new bond connecting the two sites.

As a consequence, the bond dimension after the local update is bounded by

$$\chi' \leq \chi d.$$

Equivalently, this reflects the fact that the Schmidt rank across the bond can increase by at most a factor d when one physical degree of freedom is absorbed into each side of the bipartition. *In practice, for generic time evolution, this bound is typically saturated, leading to a rapid growth of entanglement.* To keep the MPS representation efficient, one therefore truncates the decomposition by retaining only the largest $\chi_{\max}$ singular values. This truncation controls the bond dimension at the cost of a controlled error, and is the essential approximation underlying TEBD and related real-time MPS algorithms.

5.3.1 Entanglement growth and limitations of real-time MPS

The effectiveness of TEBD and related MPS-based time-evolution algorithms is ultimately controlled by the growth of bipartite entanglement. Given a bipartition of the chain into left and right halves, the entanglement entropy is defined as

$$S = -\sum_{\alpha} \lambda_\alpha^2 \log \lambda_\alpha^2,$$

where $\{\lambda_\alpha\}$ are the singular values appearing in the Schmidt decomposition across the corresponding bond. In an MPS representation, these singular values are precisely the diagonal entries of the tensor Σ, and the bond dimension χ sets an upper bound on the entropy,

$$S \leq \log \chi.$$

During real-time evolution under a local Hamiltonian, entanglement typically increases as quantum correlations spread through the system. In many generic, interacting systems this increase is observed to be approximately linear in time,

$$S(t) \sim v_E \, t,$$

a behavior often referred to as *ballistic entanglement spreading*. The terminology reflects the fact that entanglement propagates through the system at a finite velocity, similarly to a wavefront moving at constant speed.

Other behaviors are also possible. In some systems, entanglement may grow more slowly than linearly, for instance following a sub-ballistic or diffusive-like law, while in specially engineered settings it may grow faster. The rate at which entanglement spreads is a key physical property: slow entanglement growth is desirable when trying to keep parts of a system effectively isolated, while fast growth is essential for protocols that aim to generate entanglement efficiently, such as quantum information processing or scrambling dynamics.

From a numerical perspective, linear (ballistic) entanglement growth has an important consequence. Since an MPS with bond dimension χ can only represent entanglement entropy up to $S \sim \log \chi$, maintaining an accurate description of the state requires

$$\chi(t) \sim e^{S(t)}.$$

As a result, the computational cost of real-time MPS simulations typically grows exponentially with time, which constitutes the main limitation of TEBD-like algorithms.

There are, however, some exceptions to this generic behavior, for example when the dynamics of the system restrict how correlations can spread. One particularly important example is provided by *confining systems*, where elementary excitations cannot move freely and are instead bound together. Intuitively, if excitations are forced to remain close to each other by an effective restoring force, correlations cannot propagate over long distances. As a consequence, entanglement spreads very slowly or may even remain bounded for long times, making such systems unusually favorable for MPS simulations.

Reduced entanglement growth can also occur in integrable models, in disordered systems exhibiting many-body localization, or whenever strong constraints limit the accessible region of Hilbert space. Understanding when and why entanglement growth is suppressed is therefore crucial, both for identifying regimes where tensor-network methods remain efficient and for gaining physical insight into the dynamics of quantum many-body systems.

Chapter 6

Variational Monte Carlo

6.1 A computational point of view to the Many-Body problem

TODO: I should move this somewhere closer to the introduction chapter... In what follows, we will restrict ourselves to the case of closed-systems evolving unitarily. Similar arguments are, however, valid also for more general open systems.

For a many-particle system composed of N coupled degrees of freedom, the state of a quantum system is a vector $|\psi\rangle$ in the Hilbert space $\mathcal{H}^{\otimes N}$. The wave-function evolves over time according to the Schroedinger's equation

$$\frac{d\,|\psi(t)\rangle}{dt} = -i\hat{H}\,|\psi(t)\rangle \tag{6.1}$$

The quantum mechanics of an isolated system is therefore a deterministic theory where, as long as the Hamiltonian $\hat{H}$ is given, the fundamental laws are known. However, integrating this differential equation given some initial condition has a computational complexity that grows exponentially with the number of degrees of freedom N, and cannot therefore be achieved straightforwardly for realistic system sizes. This is commonly referred to as the *Quantum Many-Body Problem*.

From a computational point of view, there are two closely-related facets to the Many-Body problem:

- The **exponential amount of memory** necessary to store the wave-function as a vector in the memory of a computer;
- The **exponential number of operations** necessary to multiply operators and wave-functions in the hilbert space, which translate to a run-time of algorithms that grows exponentially with the system-size.

The *Quantum Many-Body Problem* makes it unreasonably expensive to study the properties of realistic-size quantum-mechanical systems. The brute-force simulation of ~ 50 interacting spin-1/2 particles requires approximately the entire resources of one of the top-10 supercomputers built in 2023, and simply adding an extra particle would double those requirements. However, realistic systems hold a number of particles of the order of $N_a \approx 10^{23}$, and even the mesoscopic scale that is capturing considerable interest accounts for several hundreds to a few thousands particles.

It is therefore clear that the study of physically and technologically relevant quantum systems will be very often out of reach of brute-force numerical methods and advanced numerical schemes, either approximated or relying on some clever computational technique, will be necessary. As the **exponential memory problem** arises from the large size of the Hilbert space, there are two broad families of methods that can be exploited to solve it:

- **Truncation schemes:** Truncating the Hilbert space, and projecting the problem to a smaller, polynomially-sized space, will address both facets of the Many-Body problem at the cost of losing some details of the description in the process. It is of paramount importance to neglect only regions of the hilbert space that do not affect the relevant physics.

  As the exponential growth of the Hilbert space is caused by quantum-mechanical entanglement effects among arbitrary set of particles, such truncations must generally neglect quantum phenomena and only consider few perturbative orders beyond what is predicted by classical theory.

Common such treatments are semiclassical techniques only valid in the limits of weak interactions or truncations considering only short-range correlations.

There also exist more refined truncation schemes relying on a controlled truncation, such as *Matrix Product States* and *tensor networks in general*. Along a calculation it is possible to maintain the truncation error below a desired threshold by continuously increasing the domain considered and therefore perform *numerically-exact* results for problems where the cost remains finite. Unfortunately, only a small subset of calculations will have a bounded computational cost for a finite error and therefore those techniques, while extremely powerful, cannot be always used. In particular, Open challenges for tensor networks nowdays are systems beyond 1 dimension, the dynamics up to long times and the simulation of dissipative quantum systems.

- **Compression schemes:** If the space is large, and the problem unstructured, truncation schemes might not be able to yield an *exponential speedup* and therefore address the many-body problem. This is the common scenario. In such cases, it is possible to leverage compression schemes that encode the state of the system in a small latent space of polynomial dimensions to address the exponential growth of memory. This mapping is in general highly nonlinear.

  A consequence of the nonlinear mapping between the latent representation space and the original hilbert space is that it is in general not possible to rewrite the calculation of physical observables directly in the latent space, and therefore a decompression step is required. As expectation values require knowledge of the wave-function over the whole hilbert-space, it might appear that computing obseravbles will require decompressing the whole state and therefore account for an exponential cost. To avoid this, compression schemes are usually combined with techniques that trade the exact calculation of observables for a manageable computational cost, usually resorting to schemes derived from Monte Carlo sampling.

The fundamental difference between the two schemes is that when we truncate the space we are able to rewrite (or project) any operator or mathematical object necessary for our algorithm in the truncated space, while compression schemes in principle might be able to compress only the state and not the other objects. If we look merely at the state, such as the wave-function, the procedures might look surprisingly similar and indeed it is possible to think about truncation schemes as compression methods. In the language of Machine Learning, we can identify the difference between truncation and compression scheme in the following: *truncation schemes have a latent space that is a subset of the full hilbert space, while compression schemes have a latent space that is completely different.*

Compression schemes therefore are wholly determined by the choice of the nonlinear mapping between the physical hilbert space and the latent space. In the early years of variational methods, physically-motivated mappings where made such that some sort of physical understanding of the latent space could be kept. Examples in the case of fermionic systems are the Jastrow or Gutzwiller ansatze [], Slater determinant or pair-product states [] and many more. Those ansatze were built to capture specific features efficiently (strong density-density correlations, one or two particle excitations...) but cannot effectively capture

The connection between **Computational Physics** and **Machine Learning** is in the choice

6.2 Introduction to Variational Monte Carlo

Variational Monte Carlo is a collection of numerical methods that combine *variational encodings of the wave-function* with *Monte-Carlo integration* methods in order to compute properties of quantum mechanical systems. While originatd in Physics, in recent years a deep connection between those algorithms and Machine Learning has been uncovered and explored, which has largely benefitted the domain.

The two main ingredients are the *Variational* component, which solves the *memory complexity* by efficiently represent wave-functions, and the *Monte-Carlo* component, which addresses the exponential *runtime complexity* of computing observables in exchange of stochastic estimations.

We remark that Variational Monte Carlo (VMC) was originally an algorithm to compute the ground-state wave-function of an Hamiltonian. However, in this manuscript we will refer to VMC as the family of algorithms, and ground-state search as a particular application among many, such as steady-state search, dynamics simulation and much more.

6.2.1 Variational wave-functions

Let us consider an Hilbert space $\mathcal{H}$ for which we will consider a basis $B_{\mathcal{H}} = \{|x\rangle\}$. Wave-functions belonging to the Hilbert space can then be expanded onto the basis as follows:

$$|\psi\rangle = \sum_x \langle x|\psi\rangle \, |x\rangle \tag{6.2}$$

where the sum runs over all possible local basis elements. In the case of Spin-$\frac{1}{2}$ a valid choice of the basis is the eigenstates of σ^z, $\{\uparrow, \downarrow\}$, and the wave-function can be expanded as

$$|\psi\rangle = \langle\uparrow|\psi\rangle \, |\uparrow\rangle + \langle\downarrow|\psi\rangle \, |\downarrow\rangle \, . \tag{6.3}$$

If the hilbert space is infinite, for example when considering the wave-functions of particles in a 1-Dimensional space$\mathcal{H} = L^2(\mathbb{R})$, the sum should be replaced by an integral and the basis can be taken to be any value $x \in \mathbb{R}$. The complex scalar coefficients $\langle\cdot|\psi\rangle \in \mathbb{C}$ fully determine the wavefunction in a given basis $B_{\mathcal{H}}$.

The space of many-body wave-functions obtained by coupling N local degrees of freedom is $\mathcal{H}^{\otimes N}$. Similarly, the wave-function can be expanded on the many-body basis $B_{\mathcal{H}^{\otimes N}} = \{|x_1, \dots x_N\rangle\}$ as

$$|\psi\rangle = \sum_{x_1, \dots x_N} \langle x_1, \dots x_N|\psi\rangle \, |x_1, \dots x_N\rangle \, , \tag{6.4}$$

where the sum runs over the $\sim d^N$ many-body basis elements. We remark that it would also be possible to chose a basis that is not *local*, or one that is overcomplete, but we won't be discussing those cases.

The wave-function $|\psi\rangle$ defined on a specific basis $B_{\mathcal{H}}$ as a map that associates complex numbers to the basis elements

$$|\psi\rangle : \qquad B_{\mathcal{H}} \quad \to \quad \mathbb{C} \tag{6.5}$$

$$|x_1, \dots x_N\rangle \to \langle x_1, \dots x_N|\psi\rangle := \psi_{x_1, \dots x_N} \, . \tag{6.6}$$

As we saw before, if the Hilbert space is discrete, the wave-function can be represented as an exponentially large complex -vector $\mathbb{C}^{d^N}$, and storing it in memory involves storing an exponential number of complex coefficients $\psi_{x_1, \dots x_N}$. Fairly quickly, as N increases, this becomes untractable: for example, the largest supercomputer on earth at the time of writing, Frontier, has $9PB = 9 \cdot 10^{15}$ Bytes of memory. Assuming double-precision floating point storage, every complex number requires 16 Bytes to be stored, which means that Frontier can store approximately $5 \cdot 10^{14} \approx 2^{49}$ complex numbers. This means that, by leveraging the entirety of the memory available on the largest supercomputer on earth we can only encode the wave-function as a full state-vector for 49 spin-1/2 particles.

This exponenential-memory requirement is indeed the *memory facet* of the quantum many-body problem. Intuitively, the solution is to compress the wave-function from the *uncompressed state-vector representation* to some exponentially compressed *compressed representation*. Whether such an efficient representation exists, and can be lead to a loss-less compression, is stil an open research problem. The variational approach relies on nonlinear maps which associate wave-function amplitudes to parameter vectors θ in a latent space[1] $\mathcal{W}$. We call those maps **variational ansatze**, and they are defined by

$$\Psi : \mathcal{W} \to \mathcal{H} \tag{6.7}$$

$$\theta \to |\psi_\theta\rangle \, . \tag{6.8}$$

Assuming that $\dim[\mathcal{W}] \sim \text{Poly}(N) \ll \dim[\mathcal{H}]$ the memory complexity aspect of the Many-Body problem is addressed. In order to effectively use a variational ansatz we must be able to query its amplitudes $\langle x_1, \dots x_N|\psi_\theta\rangle$ for any possible basis element $|x_1, \dots x_N\rangle$. This is usually achieved by evaluating the function

$$\Psi_\theta(x_1, \dots x_N) = \langle x_1, \dots x_N|\psi_\theta\rangle \, . \tag{6.9}$$

If we interpret variational ansatze as state compression algorithms, θ is the compressed representation, the map Ψ is the compression/decompression algorithm itself and the individual bitstrings $\langle x_1, \dots x_N|x_1, \dots x_N\rangle$ are *partial* decompressions of our wave-function.

[1] A latent space $\mathcal{W}$ is a vector space of dimension smaller than the original used to store a compressed representation.

Variational ansatze and algorithms

A variational ansatz is most naturally defined as a higher-order map: it takes as input a set of parameters $\theta \in \Theta$ and returns a function over the basis elements $x \in B$ of the Hilbert space, producing complex amplitudes. Formally,

$$\Psi : \Theta \to (B \to \mathbb{C}), \qquad \Psi \equiv \lambda\theta.\, \lambda x.\, \Psi_\theta(x).$$

```
type Basis = [X] -- product basis B = X^N
type Params = Θ-- parameter space
type Amplitude = Complex Double --

psi :: Params -> Basis -> Amplitude
psi  x = {- amplitude Ψ _(x) -}

-- Functional view: ansatz returns a function over basis states
mkPsi :: Params -> (Basis -> Amplitude)
mkPsi =  \ -> (\x -> psi  x)
```

Estimating expectation values

A variational ansatz as discussed above only addresses the exponential growth of the memory required to store a quantum state. However expectation values of an arbitray operator $\hat{O}$

$$\langle O \rangle_\theta = \frac{\langle \psi_\theta | \hat{O} | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} = \frac{\sum_{\mathbf{x},\mathbf{y}} O_{\mathbf{x},\mathbf{y}} \psi^\star_\theta(\mathbf{x}) \psi_\theta(\mathbf{y})}{\sum_{\mathbf{x}} |\psi_\theta(\mathbf{x})|^2}, \tag{6.10}$$

still involve three sums over an untractably-large basis of the Hilbert space[2]: two at the nominator and one at the denominator. If we were to compute the expression as shown in eq. (6.10) we would have to query the wave-function ansatz at least once for every basis element, leading to an exponentially-growing number of operations.

To avoid this untractable cost, we exploit several facts:

- The sum over $\mathbf{y}$ at the nominator can be performed exactly because the matrix representation of the operator $\hat{O}$ is sparse (or K-local).
- By using Monte-Carlo integration to avoid computing the denominator and to replace the sum over $\mathbf{x}$ can be replaced by a sum over a tractable set of configurations.

In pratice, we perform a manipulation of eq. (6.10) to explicit a probability density in order to rewrite it as a Monte-Carlo integral This manipulation is very common in the field of Variational Monte Carlo and relies on multiplying and dividing by a factor proportional to the density, in this case $\langle \mathbf{x} | \psi \rangle$,

$$\begin{aligned}
\langle O \rangle_\theta &= \frac{\langle \psi_\theta | \hat{O} | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} && (6.11)\\
&= \sum_{\mathbf{x}} \frac{\langle \psi_\theta | \mathbf{x} \rangle \langle \mathbf{x} | \hat{O} | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} && (6.12)\\
&= \sum_{\mathbf{x}} \frac{\langle \psi_\theta | \mathbf{x} \rangle \langle \mathbf{x} | \psi_\theta \rangle \langle \mathbf{x} | \hat{O} | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle \langle \mathbf{x} | \psi_\theta \rangle} && (6.13)\\
&= \sum_{\mathbf{x}} \frac{|\psi_\theta(\mathbf{x})|^2}{\langle \psi_\theta | \psi_\theta \rangle} \frac{\langle \mathbf{x} | \hat{O} | \psi_\theta \rangle}{\langle \mathbf{x} | \psi_\theta \rangle} && (6.14)\\
&= \sum_{\mathbf{x}} p_\theta(\mathbf{x}) O^{\mathrm{loc}}_\theta(\mathbf{x}) && (6.15)\\
&= \mathbb{E}_{p_\theta(\mathbf{x})} \left[O^{\mathrm{loc}}_\theta(\mathbf{x}) \right]. && (6.16)
\end{aligned}$$

[2] In the case of continuous bases, the sums should be interpreted as integrals.

In the calculations above we have defined the Born probability amplitude $p_\theta(\mathbf{x}) = \frac{|\psi_\theta(\mathbf{x})|^2}{\langle\psi_\theta|\psi_\theta\rangle}$, which can be trivially shown to be real, positive, and sum up to 1. We have also defined the *local estimator*

$$O^{\text{loc}}_\theta(\mathbf{x}) = \frac{\langle\mathbf{x}|\hat{O}|\psi_\theta\rangle}{\langle\mathbf{x}|\psi_\theta\rangle}. \tag{6.17}$$

The manipulation above shows that the quantum expectation value can be rewritten as a classical average over the Born probability amplitude $p_\theta(\mathbf{x})$ of the local estimator $O^{\text{loc}}(\mathbf{x})$.

Monte-carlo integration can be leveraged to efficiently estimate quantum expectation values, replacing the full summation over the Hilbert space of eq. (6.15) with a sum over a small set of samples when estimating the expectation value (eq. (6.16)) using the sample-mean estimator defined in eq. (6.19) of the infobox.

In practice, this leads to the expression

$$\langle O\rangle_\theta = \mathbb{E}_{p_\theta(\mathbf{x})}\left[O^{\text{loc}}_\theta(\mathbf{x})\right] \approx \frac{1}{N_s}\sum_{x\in\{x\}} O^{\text{loc}}_\theta(\mathbf{x}), \tag{6.18}$$

where the accuracy of the estimate increases as the number of samples is increased.

If the variance of the estimator grows polynomially with the system size ($\mathbb{V}\text{ar}[O^{\text{loc}}] \sim \text{Poly}(N)$), a polynomial number of samples will suffice for all system sizes and we will have a polynomially scaling computational methods. This is the case for all physical Hamiltonians (see **??**)[3].

warning It should be noted that estimating stochastically expectation values by drawing samples from $p_\theta(\mathbf{x}) = |\psi_\theta(\mathbf{x})|^2 / \langle\psi_\theta|\psi_\theta\rangle$ might not be the best possible choice. Indeed, there is no a-priori reason to sample from this distribution other than the fact that physicsts like it because it's the Born Amplitude, or because this has been the de-facto standard for the last 40 years.

[3] An often asked question is whether there is a sign problem when computing an expectation value this way. Taking the standard interpretation of the sign problem to be an exponentially-increasing variance for the estimator, then it is clear that VMC estimates of an observable are sign-problem free.

Monte-Carlo integration

Classical expectation values of the form

$$\bar{A} := \mathbb{E}_{x\sim p(x)}\left[A(x)\right] := \mathbb{E}_{p(x)}\left[A(x)\right] \tag{6.19}$$

can be estimated with Monte-Carlo sampling by generating a polynomially large set of samples x distributed according to the probability distribution, and estimating the average with the sample mean estimator

$$\mathbb{E}_{p(x)}\left[A(x)\right] \approx \frac{1}{N_s}\sum_{x\in\{x\}} A(x). \tag{6.20}$$

It is possible to show that this estimator is unbiased and has a statistical error proportional to the variance of the local estimator,

$$\epsilon_A = \sqrt{\frac{\mathbb{V}\mathrm{ar}(A)}{N_s}}, \tag{6.21}$$

suggesting that the estimate can be systematically improved by increasing the number of samples N_s. As long as the variance of the estimator is bounded, the approximate number of samples necessary to get an estimate with a given accuracy ϵ is approximately $N_s \sim \mathbb{V}\mathrm{ar}\left[A\right]\epsilon^2$.

Signal to noise ratio: In general it is useful to know how many samples are necessary to estimate *effectively* a quantity, in which case the absolute error given by eq. (6.21) is not useful. The relevant quantity is then the *signal to noise ratio* **(SNR)** of the stochastic estimator, which is proprtional to the ratio between the estimated value *(the signal)* and the variance *(the noise)*. The definition is,

$$\mathrm{SNR}_{p(x)}\left[A(x)\right] = \sqrt{\frac{\left(\mathbb{E}_{p(x)}\left[A\right]\right)^2}{\mathbb{V}\mathrm{ar}_{p(x)}\left[A\right]}}, \tag{6.22}$$

and the number of samples necessary to estimate a quantity correctly is inversely proportional to it.

The local estimator We have shown above that we can replace the intractable sum in eq. (6.10) with a stochastic average to make the cost manageable, however this assumes that the cost of every term in the sum is also polynomial. In particular, it is important to note that the local estimator also contains a sum over the full Hilbert space,

$$O_\theta^{\mathrm{loc}}(\mathbf{x}) = \sum_{\mathbf{y}} O_{\mathbf{xy}}\frac{\psi_\theta(\mathbf{y})}{\psi_\theta(\mathbf{x})}. \tag{6.23}$$

This expression corresponds to performing a matrix-vector product, where the matrix O is multiplied by the vector ψ_θ, and keeping only the row corresponding to $\mathbf{x}$. However, it is possible to verify that most physical Hamiltonians and observables are sparse and therefore most entries $O_{\mathbf{xy}}$ will be zero.

Let's consider a *Transverse Field Ising Model* (**TFIM**): the model's hamiltonian is

$$\hat{H} = \sum_i \sigma_i^{(x)} + \sum_{\langle i,j\rangle}\sigma_i^{(z)}\sigma_j^{(z)} \tag{6.24}$$

... This proves that for physical systems for which we are able to query the wave-function amplitudes, we are able to estimate expectation values efficiently at the cost of having an error bar instead of a perfect knowledge of the value.

Zero-variance principle It is easy to show that when the state is an eigenstate of the operator, the local estimator does not depend anymore on the sampled configuration and the variance of the estimator becomes zero, effectively allowing a perfect estimate to be obtained with a single sample.

To prove it, simply consider a state ψ_θ such that $\hat{O}\,|\epsilon\rangle = \epsilon\,|\epsilon\rangle$ for a set of values ϵ labelling the eigenstates of the operator. Then, it follows that

$$O_\epsilon^{\text{loc}}(\mathbf{x}) = \frac{\langle \mathbf{x}|\hat{O}|\epsilon\rangle}{\langle \mathbf{x}|\epsilon\rangle} = \epsilon \frac{\langle \mathbf{x}|\epsilon\rangle}{\langle \mathbf{x}|\epsilon\rangle} = \epsilon \tag{6.25}$$

And therefore $\mathbb{V}\text{ar}_\epsilon[O] = 0$. This property of the estimator is known as *Zero-variance property* and it is an essential property of the loss function when optimising it.

In following sections we will show that when minimising the Energy to find the ground state, as the ground-state is an eigenstate of the hamiltonian, we will be able to get extremely accurate ground-states with a reasonable number of samples thanks to this property.

In fact, it is possible to show that the signal to noise ratio of this estimator is inversely proportional to the relative error on the ground-state energy, meaning that when minimizing the energy the signal to noise ratio of the energy estimator will eventually diverge according to,

$$\text{SNR}_{p_\theta(x)}\left[H_\theta^{\text{loc}}(x)\right] \propto |E(\theta) - E_0|^{-1}, \tag{6.26}$$

which is consistent with the fact that, at convergence, the variance of the energy is zero. A proof of that can be found in section 6.2.9.

Continuous systems When working with systems with continuous degrees of freedom, such as particle in 1$^{\text{st}}$ quantisation, one can substitute the sums above with integrals $\sum_x \to \int dx$, obtaining similar expressions for the expectation value of the energy.

The only key difference is in the local estimator, which leads to slightly different expressions because operators such as the laplacian are not $K-$local.

For example, consider the hamiltonian $\hat{H} = \sum_i \frac{\hat{p_i}^2}{2m} + \hat{V}$, where $\hat{p}_i = i\hbar\partial_{x_i}$ is the momentum operator and $\hat{V}\,|x\rangle = V(x)$ is some potential which depends only on the position. Then, the local estimator will be,

$$H^{\text{loc}}(\mathbf{x}) = \frac{\langle \mathbf{x}|\hat{H}|\psi\rangle}{\langle \mathbf{x}|\psi\rangle} = \sum_i \frac{\hbar^2}{2m}\frac{\partial_{x_i}^2\psi(\boldsymbol{x})}{\psi(\boldsymbol{x})} + V(\boldsymbol{x}), \tag{6.27}$$

which can be evaluated exactly with automatic differentiation[4].

6.2.2 Optimization of variational wave-functions

In the preceding subsection we have seen that Variational ansatze with a polynomial number of parameters and with a polynomial query complexity can be used to circumvent the Many-Body Problem by combining them with Monte Carlo sampling of observables. However, we are usually interested in the properties of specific states, such as the ground-states of Hamiltonians or those obtained by evolving according to the Schroedinger's equation a well defined initial state. In general, VMC and related methods require a *variational search* on the parameter space in order to find a set of parameters that well approximates the state we are interested in.

In general, the *variational search* can be treated as an optimisation problem, where a *loss function* $\mathcal{L}(\theta)$ is minimised across the parameter space $\mathcal{W}$ with an optimization algorithm. It is of paramount importance that the optimisation algorithm has a bounded complexity and that it can converge on average. Depending on the particular target state, there exist several different optimisation algorithms, which are often related to each other. In this section we will present the single most important one as a starting point to discuss VMC: energy minimisation.

Ground state search and Variational Principle Given an Hamiltonian H, it is easy to show that the ground-state $|\epsilon_0\rangle$ with energy ϵ_0 is the one with the lowest energy across all possible states in

[4]The local estimator of the kinetic term is the trace of the hessian, which, using backward-mode automatic differentiation, is as expensive to compute as the full hessian. In recent years several computational tricks using higher-order forward mode AD have been proposed. Those methods lead to $O(N)$ asymptotical improvements, but are non-standard and may be quite complicated to use in a generic code.

the Hilbert space[5]. This is known as the *ground-state variational principle*, which states that

$$E(\theta) = \frac{\langle\psi_\theta|H|\psi_\theta\rangle}{\langle\psi_\theta|\psi_\theta\rangle} \geq \epsilon_0 \quad \forall\theta \in \mathcal{W} \quad \text{and} \quad E(\theta) = \epsilon_0 \iff |\psi_\theta\rangle = |\epsilon_0\rangle \tag{6.29}$$

This states, in other words, that it is possible to interpret the energy as a loss function, and that by minimising it we can find the ground-state,

$$\theta^\star = \arg\min_\theta E(\theta). \tag{6.30}$$

To find this exact solution we would require that the latent space $\mathcal{W}$ contains a representation of the ground-state[6], whoever this is unlikely for most variational ansatze. In practice, we assume that the lower the energy, the closer we get to the exact ground-state energy and the accuracy of the wave-function as well.

We remark that it is possible to analytically bound, by means of the relative energy error, the fidelity of the variational state with respect to the ground-state. Such a bound, however, grows with a very large power of the relative error and is therefore not very useful in practice. This suggests that while the variational principle is grounded in mathematical motivation, its daily application requires a small leap of faith unless we're able to find very small relative errors[7].

As we generally work with thousands to millions of variational parameters, we cannot use global optimisation algorithms to solve eq. (6.30), and we rely instead of local optimisation techniques such as conjugate[8]-gradient descent and its cousins, where the parameters are iteratively optimised according to

$$\theta(\tau+1) = \theta(\tau) - \eta\nabla_{\theta^\star}E(\theta), \tag{6.31}$$

where τ is an integer labelling the optimisation steps[9] and η is the so-called *learning rate* describing the speed at which we change the parameters. Do notice that it is possible to complicate arbitrarily this equation, accounting for momentum, higher order derivatives and much more in order to get a faster convergence.

In order to perform a gradient-descent optimisation according to eq. (6.32) we must be able to estimate the gradient of the energy. The gradient with respect to the variational parameters of an expectation value is,

$$\nabla_{\theta^\star}E(\theta) = \nabla_{\theta^\star}\frac{\langle\psi_\theta|H|\psi_\theta\rangle}{\langle\psi|\psi\rangle} \tag{6.33}$$

$$= \frac{\langle\nabla_\theta\psi_\theta|H|\psi_\theta\rangle + \langle\psi_\theta|H|\nabla_{\theta^\star}\psi_\theta\rangle}{\langle\psi|\psi\rangle} - E(\theta)\frac{\langle\nabla_\theta\psi_\theta|\psi_\theta\rangle + \langle\psi_\theta|\nabla_{\theta^\star}\psi_\theta\rangle}{\langle\psi_\theta|\psi_\theta\rangle}. \tag{6.34}$$

where we use the notation $|\nabla_\theta\psi_\theta\rangle \equiv \sum_x \nabla_\theta\psi_\theta(x)\,|x\rangle$. The expression contains several brakets, which as we have discussed before incur into an exponential complexity when evaluated exactly. As done previously we must transform those sums over the whole Hilbert space onto expectation values.

[5] The proof relies on its eigendecomposition $H\,|\epsilon_i\rangle = \epsilon_i\,|\epsilon_i\rangle$ where $\epsilon_i < \epsilon_{i+1}$. Then, it is possible to show that given a state $|\psi\rangle = \sum_i c_i\,|\epsilon_i\rangle$ where $c_i = \langle\epsilon_i|\psi_\theta\rangle$

$$\frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} = \sum_i \frac{|c_i|^2}{\sum_i |c_i|^2}\epsilon_i \tag{6.28}$$

[6] More formally, we require that the Image of the latent space under the action of the ansatz contains the ground-state

[7] The standard way this issues shows up in practice is that even if one gets a very good ground-state energy, other observables might be qualitatively wrong.

[8] If your latent space only contains real parameters, the conjugate gradient and the gradient coincide, so we can effectively talk about gradient-descent, which is probably familiar to anyone reading those notes. However, when minimising a real-valued loss function of complex parameters, the direction of steepest descent is given by the conjugate gradient, not the gradient. For more information look at the infobox in the next page.

[9] The choice of τ wants to highlight that it is possible to interpret this discrete optimisation process as some first order discrete solution of an associated ODE

$$\frac{d\theta}{d\tau} = -\eta\nabla E(\theta), \tag{6.32}$$

and if the noise introduced by stochastic sampling is introduced, an extra Wiener noise term could be used to model the ideal dynamics of the training process.

Real, Complex and Holomorphic functions

When deriving the expressions of the gradients of a loss function such as being done here, there are in general two cases that might appear: the case where (i) the parameters of the loss function are real, and the case where (ii) they are complex.
First, we remark that while a loss with real parameters requires the gradient for the optimisation, a loss with complex parameters requires knowledge of the conjugate-gradient. This is because the direction of largest increase of the loss is identified by the conjugate gradient. To prove it, we consider a function

$$E(\theta+\delta\theta) = E(\theta) + \delta\theta\cdot\nabla_\theta E(\theta) + \delta\theta^\star\cdot\nabla_{\theta^\star}E(\theta) + O(\delta\theta^2) \tag{6.35}$$

and therefore the largest change of the loss function is obtained by[a]

$$\arg\max_{\delta\theta}(E(\theta+\delta\theta)-E(\theta)) = 2\arg\max_{\delta\theta}(\mathrm{Re}[\delta\theta\cdot\nabla_\theta E(\theta)]). \tag{6.36}$$

The value of $\delta\theta$ which maximises the condition is that where the argument of $\mathrm{Re}[\cdot]$ is itself real, therefore when $\delta\theta = (\nabla_\theta E(\theta))^\star = \nabla_{\theta^\star}E(\theta)$.

Holomorphicity A complex-valued wave-function is holomorphic if it satisfies the condition $\nabla_{\theta^\star}\psi_\theta(x) = 0$. From this it is easy to prove that $\nabla_\theta\langle\psi_\theta|x\rangle = \langle\nabla_{\theta^\star}\psi|x\rangle = 0$.
For this reason, before doing any calculation involving gradients we should be clear on what case we are considering: wavefunctions with real parameters, holomoprhic-complex, or non-holomorphic complex wavefunctions? Interestingly, the complex non-holomorphic case can be reduced to the real-parameters by splitting every parameter into its real and imaginary part.
Which particular case is considered is not very important when studying the method from an high level, but it is vital to be aware of which case is being considered when implementing them in code.

[a] We repeatedly use here the property that $(\nabla_\theta E(\theta))^\star = \nabla_{\theta^\star}E(\theta)$, which stems from the fact that $E(\theta)\in\mathbb{R}$. This will always be the case, as we do not discuss complex-valued loss functions.

To continue the calculation, we must decide whether we will consider (i) real-valued parameters $\theta\in\mathbb{R}^M$ or (ii) complex-valued parameters with an holomorphic wave-function[10]. Those two cases lead to almost identical expressions, yet have a different derivation.

Gradient derivation: complex parameters with holomorphic wave-function We now consider a complex holomorphic wavefunction, such that $|\nabla_{\theta^\star}\psi\rangle = 0$, therefore developing further eq. (6.34) we get

$$\nabla_{\theta^\star}E(\theta) = \frac{\langle\nabla_\theta\psi_\theta|H|\psi_\theta\rangle}{\langle\psi|\psi\rangle} - E(\theta)\frac{\langle\nabla_\theta\psi_\theta|\psi_\theta\rangle}{\langle\psi_\theta|\psi_\theta\rangle}, \tag{6.37}$$

from here, the two terms can be treated as:

$$\frac{\langle\nabla_\theta\psi_\theta|H|\psi_\theta\rangle}{\langle\psi|\psi\rangle} = \sum_x \frac{1}{\langle\psi|\psi\rangle}\langle\nabla_\theta\psi_\theta|x\rangle\ \langle x|H|\psi_\theta\rangle \tag{6.38}$$

$$= \sum_x \frac{|\psi(x)|^2}{\langle\psi|\psi\rangle}\frac{\langle\nabla_\theta\psi_\theta|x\rangle}{\langle\psi|x\rangle}\frac{\langle x|H|\psi_\theta\rangle}{\langle x|\psi\rangle} \tag{6.39}$$

$$= \sum_x p_\theta(x)\,(\nabla_\theta\log\psi(x))^\star\,H^{\mathrm{loc}}(x) \tag{6.40}$$

$$= \mathbb{E}_{p_\theta(x)}\left[(\nabla_\theta\log\psi(x))^\star\,H^{\mathrm{loc}}(x)\right] \tag{6.41}$$

[10] A third case, (iii) complex-valued parameters with a non-holomorphic wave-function, can be treated as a function with twice the real parameters and be dealt with (i).

we remark that in eq. (6.39) we multiplied and divided by $|\psi(x)|^2$, which might be zero. The formula above is in fact slightly wrong, but we will ignore this problem for now and discuss it further in TODO.

We can also express the second term of eq. (6.37) as an expectation value, obtaining

$$\frac{\langle \nabla_\theta \psi_\theta | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} = \mathbb{E}_{p_\theta(x)} \left[\nabla_\theta \log \psi^\star(x) \right]. \tag{6.42}$$

Inserting the two terms of eqs. (6.41) and (6.42) back into the original gradient eq. (6.37), replacing $E(\theta)$ with it's stochastic estimator $E(\theta) = \mathbb{E}_p \left[H^{\text{loc}}(x) \right] = \bar{H}^{\text{loc}}$,

$$\nabla_{\theta^\star} E(\theta) = \mathbb{E}_{p_\theta(x)} \left[(\nabla_\theta \log \psi(x))^\star \left(H^{\text{loc}}(x) - \bar{H}^{\text{loc}} \right) \right], \tag{6.43}$$

where $\bar{H}^{\text{loc}} = \mathbb{E}_{\theta(x)} \left[H^{\text{loc}}(x) \right]$ is the average of the local estimators.

It can also be easily shown that this estimator is a covariance,

$$\boldsymbol{F} = \nabla_{\theta^\star} E(\theta) = \text{Cov}_{p_\theta(x)} \left(\nabla_\theta \log \psi^\star(x), H^{\text{loc}}(x) \right) \qquad \text{[complex holomorphic } \psi\text{]} \tag{6.44}$$

which highlights the fact that if the variance of $H^{\text{loc}}(x)$ is 0, that is, if the wave-function represents an eigenstate, the estimator of the gradient will be exactly zero.

Gradient derivation: real parameters The derivation in the case of real parameters follows a similar pattern, but we will now use the fact that $|\nabla_{\theta^\star} \psi\rangle = |\nabla_\theta \psi\rangle$ (as the parameters are purely real). Starting from eq. (6.34), it should be easy to see that

$$\frac{\langle \nabla_\theta \psi_\theta | H | \psi_\theta \rangle + \langle \psi_\theta | H | \nabla_{\theta^\star} \psi_\theta \rangle}{\langle \psi | \psi \rangle} = 2 \,\text{Re} \left[\frac{\langle \nabla_\theta \psi_\theta | H | \psi_\theta \rangle}{\langle \psi | \psi \rangle} \right] \tag{6.45}$$

$$\frac{\langle \nabla_\theta \psi_\theta | \psi_\theta \rangle + \langle \psi_\theta | \nabla_{\theta^\star} \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} = 2 \,\text{Re} \left[\frac{\langle \nabla_\theta \psi_\theta | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} \right] \tag{6.46}$$

from which it can easily be shown that the gradient of the energy with real parameters is simply twice the real part of the *variational forces* defined in eq. (6.44). In practice,

$$\nabla_{\theta^\star} E(\theta) = 2 \,\text{Re} \left[\boldsymbol{F} \right] = 2 \,\text{Re} \left[\text{Cov}_{p_\theta(x)} \left(\nabla_\theta \log \psi^\star(x), H^{\text{loc}}(x) \right) \right] \qquad \text{[real parameters } \psi\text{]} \tag{6.47}$$

We also remark that the argument of the real part is, for historical reasons, often called *variational forces*, and indicated with the capital letter $\boldsymbol{F}$. Interestingly, the gradient of the energy coincides with the variational forces when the wavefunction is complex-holomorphic (see eq. (6.44)), while it does not otherwise. The distinction is particularly important when discussing Natural Gradient Descent and the Time Dependent Variational principle.

The zero-gradient property

The estimator of the variational forces becomes identically zero, with zero variance, when the wave-function coincides with that of an eigenstate, because $\Delta H^{loc}(x) = 0$ for every sample. A major consequence is that if, for any reason, we manage to find the ground-state, the optimisation will stop regardless of the number of samples we are using to estimate the gradient. This is in stark contrast to what traditionally happens in a data-driven machine learning context, where the gradient at the global minima would be zero *on average*, and therefore the optimisation would not stop once it hits the global minima, and instead would start to oscillate around it.

Interestingly, it is also possible to show empirically, or by means of some approximate calculations, that the variance of this estimator *goes to zero smoothly*, meaning that as we converge the fluctuation of the gradient due to the sampling are damped and we do not need to increase the number of samples to maintain the same relative precision on the gradient. This is, again, in stark contrast to standard Data-Driven algorithms, where it is instead customary to increase the mini-batch size during calculations to reduce the sampling noise.

TODO.

Mathematical structure of the variational forces

We now want to briefly discuss the mathematical structure of the estimator of the forces $\boldsymbol{F}$. In particular, looking at eq. (6.44), we can define the centered local energies,

$$\Delta H^{loc}(x) = H^{\mathrm{loc}}(x) - \bar{H}^{\mathrm{loc}}. \tag{6.48}$$

This is to all effects a vector, where every entry correspond to the local energy wrt the i-th sample.

The term $\nabla_\theta \log \psi(x)$, instead, can be interpreted as the jacobian matrix of the function $\log \psi(x)$, where the first index runs along the different parameters θ_i and the second index runs along the different samples x. Explicitating those dimensions, we can rewrite the gradient estimate as

6.2.3 Stochastic Reconfiguration and Natural gradient descent

While first-order methods such as gradient descent and related optimisers (AdaGrad, ADAM, YOGI...) is the most widely used optimisation technique in Machine Learning, it is very susceptible to saddle points where it may get stuck[11]. While Adaptive gradient methods such as AdaGrad and ADAM fight against this issue by amplifying directions where the gradient is small, ultimately they fail as well. Second-order methods such as the Newton Method can avoid such issues, but they require performing hessian-vector products which increase the computational cost enourmously and are impractical for optimising even relatively small neural networks with more than a few 10s of thousand parameters.

Natural gradient descent is a first-order method in the *natural gradient* originally proposed by Amari at RIKEN in 1990 [], which exploits ideas from differential geometry to optimise the loss function in a parametrisation-invariant way. The natural gradient shares some properties of the newton method, but without the high computational cost. In the context of Variational Monte Carlo, the simplest flavour of natural gradient was baptized *Stochastic Reconfiguration* **(SR)** by Sandro Sorella, who independently derived it in the early 2000. Natural Gradient/SR is an essential tool of NQS calculations, and is considered necessary to achieve the highest accuracies.

In the following, we will give a modern introduction to this method, connecting the original derivation of SR with the geoemtric interpretation of natural gradient, attempting to highlight the underlying mathematical reasons for the unreasonable effectiveness of natural gradient in the context of VMC.

Quadratic problems and Imaginary time evolution

The energy is a quadratic form in the space of wave-functions, as can be clearly seen by its definition

$$E(\psi) = \frac{\langle\psi|\hat{H}|\psi\rangle}{\langle\psi|\psi\rangle}. \tag{6.49}$$

From this it can easily be shown that the energy is a convex functional of the wave-function ψ, which we know can be minimized exactly with gradient descent methods. A straightforwad approach to solve this problem is to follow the conjugate-gradient with respect to ψ until convergence,

$$\partial_{\psi^\star} E(\psi) = \frac{\hat{H}\,|\psi\rangle}{\langle\psi|\psi\rangle} - E(\psi)\frac{|\psi\rangle}{\langle\psi|\psi\rangle} = \frac{\Delta\hat{H}\,|\psi\rangle}{\langle\psi|\psi\rangle}, \tag{6.50}$$

where we used the shorthand $\Delta\hat{H} \equiv \hat{H} - E(\psi)$. And mathematicians know that this method will converge to the solution exponentially fast in the number of iterations.

This is mathematically equivalent to what physicists call *imaginary-time evolution*, which is equivalent to solving the imaginary-time Schroedinger's equation with respect to time τ,

$$\frac{d\,|\psi\rangle}{d\tau} = -(H - E(\psi))\,|\psi\rangle\,. \tag{6.51}$$

[11] Local minimas, while in theory an issue, are in practice not an issue when the parametrised function has many parameters. A sketch of the proof can be had by assuming that the directional derivative is a random variable with probability to be zero p, then the probability for a point to have all directional derivatives equal to zero would be p^M where M is the number of parameters, which goes to zero exponentially fast in the number of parameters. However, the probability to have a saddle point remains extremely high.

This differential equation can be formally integrated, which leads to a formal solution of the form (ignoring normalisation)

$$|\psi(\tau)\rangle \propto e^{-\hat{H}\tau} |\psi(0)\rangle \propto \sum_i c_i^0 e^{-(\epsilon_i - \epsilon_0)\tau} |\epsilon_i\rangle ,$$

where ϵ_i are the eigenstates and eigenvalues of the Hamiltonian, and c_i^0 are the coefficients that determine the initial state in this basis. It can easily be seen that this equation leads, in the $\tau \to \infty$ limit, to an exponentially fast convergence towards the ground state[12]

We remark that the term "$-E(\psi)$" emerges from the need to conserve the norm of the wave-function to the first order in τ[13], similarly to how it emerges from taking the derivative of the denominator of the left-hand side of eq. (6.50). And, by integrating this equation we can easily see that this equation converges to the ground-state exponentially fast in the imaginary time τ.

A key take-away should be that imaginary-time evolution in the full Hilbert space and minimising a quadratic form are essentially equivalent, and lead to an exponentially fast convergence to the global minima.

[12] Assuming that $|c_0^0| \neq 0$

[13] An alternative derivation of this expression be derived easily by considering the proper imaginary time Schroedinger's equation,

$$\frac{d|\psi\rangle}{d\tau} = -H|\psi\rangle , \tag{6.52}$$

and computing the norm of the wavefunction after an infinitesimal timestep, will give

$$\|\psi(\tau)\| = \|\psi_0\| \sqrt{1 - \tau \langle H \rangle_{\psi_0}}, \tag{6.53}$$

which leads to the expression for the normalised state

$$\frac{|\psi(\tau)\rangle}{(\|\psi(\tau)\|)} = \frac{\mathbb{I} - \tau H}{\sqrt{1 - \tau \langle H \rangle_{\psi_0}}} \frac{|\psi_0\rangle}{\|\psi_0\|} \approx \left(\mathbb{I} - \tau(H - \langle H \rangle_{\psi_0})\right) \frac{|\psi_0\rangle}{\|\psi_0\|}, \tag{6.54}$$

which suggests that integrating eq. (6.52) would lead to a norm incrasing with rate $\langle H \rangle_{\psi_0}$. To address this increase of norm, we consider as the *(Norm preserving) Imaginary time Schroedinger's equation*

The derivative operator $\hat{\partial}_i$

In the following demonstration, and in the rest of this manuscript, we will be often using the derivative operator $\hat{\partial}_i$ intended as the partial derivative with respect to the $i-$th parameter. Formally, the operator behaves like that:

$$\hat{\partial}_i : \mathcal{H} \to \mathcal{H} \tag{6.55}$$

$$|\psi_\theta\rangle \to |\partial_i \psi_\theta\rangle := \hat{\partial}_i |\psi_\theta\rangle \tag{6.56}$$

where $|\partial_i \psi_\theta\rangle$ is defined as

$$|\partial_i \psi_\theta\rangle = \sum_x \langle x|\partial_i \psi_\theta\rangle |x\rangle = \sum_x \frac{\partial \psi_\theta(x)}{\partial \theta_i} |x\rangle . \tag{6.57}$$

The expectation value of the derivative operator is

$$\left\langle \hat{\partial}_i \right\rangle_\theta = \frac{\langle \psi_\theta | \hat{\partial}_i | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} = \frac{\langle \psi_\theta | \partial_i \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} \tag{6.58}$$

While this operator is technically not an observable, because it is not hermitian, it can still be computed by sampling

$$\left\langle \hat{\partial}_i \right\rangle_\theta = \sum_x \frac{|\psi_\theta(x)|^2}{\langle \psi_\theta | \psi_\theta \rangle} \partial_i \log \psi_\theta(x). \tag{6.59}$$

Because of this relation, the $\hat{\partial}_i$ operator was routinely defined in the literature as the *log-derivative operator* $\hat{O}_i$, which has a local estimator $O_i^{\text{loc}}(x) = \partial_i \log \psi_\theta(x)$.
Note that the *log-derivative* operator O_i that is routinely found in the literature and the derivative operator $\hat{\partial}_i$ are formally equivalent. The reason why, historically, we only worked with log-derivatives is because those are needed when computing wavefunctions and can be interpreted as a proper linear operator acting on the wavefunction, while the derivative operator has a more involved formal definition related to tangent spaces.

6.2.4 Stochastic Reconfiguration and following the Imaginary time evolution

In the previous subsection we have shown that if the loss function is a quadratic form, we can quickly converge to the optimum by following the tangent vectors $|\Delta\phi\rangle \equiv \frac{\Delta \hat{H}|\psi\rangle}{\langle \psi|\psi\rangle}$. However, there are two issues:

- This vector lives in the whole Hilbert space, so it cannot be numerically computed as it would involve computations with an exponential cost.
- When using a variational ansatz for the wavefunction, ψ_θ, we must update the parameters, not directly the wave-function amplitudes.

To circumvent those issues, we must project the state evolved according to imaginary time evolution back into the manifold of variational wave-fucntions. The computational trick here is that given a variational state ψ_θ, even if we cannot formally compute the entirety of $H|\psi_\theta\rangle$, we can still query efficiently it's amplitudes, and therefore we can use it to compare it to the projection. This first derivation presented here follows the intuition and formalism of physicists. In a later section we will discuss a deeper mathematical interpretation in terms of differential geometry.

For simplicity we consider here the problem of solving a single infinitesimal imaginary-timestep τ, starting from the initial state $|\psi_\theta\rangle$. The state evolved infinitesimally in imaginary time will be

$$\frac{e^{-H\tau}|\psi_\theta\rangle}{\|e^{-H\tau}|\psi_\theta\rangle\|} \approx \left(\mathbb{I} - \tau(\hat{H} - \langle H\rangle_\theta) + O(\tau^2)\right)|\nu_\theta\rangle , \tag{6.60}$$

where we will be using latin letters like ν to indicate that the state is normalised. We can also write the expression for the state $\psi_{\theta+\tau\dot{\theta}}$ where we have evolved infinitesimally the parameters in the direction of $\dot{\theta}$. This state is

$$\left|\nu_{\theta+\tau\dot{\theta}}\right\rangle = \frac{\left|\psi_{\theta+\tau\dot{\theta}}\right\rangle}{\left\|\psi_{\theta+\tau\dot{\theta}}\right\|} = e^{i\chi(\theta)}\left(\mathbb{I}+\tau\dot{\theta}_i(\hat{\partial}_i - \langle\hat{\partial}_i\rangle_\theta) + O(\tau^2)\right)|\nu_\theta\rangle \tag{6.61}$$

where $\chi(\theta) = \tau\sum_i \mathrm{Im}(\dot{\theta}_i \langle\hat{\partial}_i\rangle_\theta)$ is a global phase. We remark that both eq. (6.60) and **??** are valid whenever $\theta_i \in \mathbb{R}$ or if $\theta_i \in \mathbb{C}$ and the ansatz is holomorphic w.r.t. the parameters (for the complete derivation, see the footnote[14]). As discussed before, a non holomorphic ansatz can be expressed in terms of real-valued parameters.

Having expressed the normalised states, the standard derivation of Stochastic Reconfiguration is to find the $\dot{\theta}_i$ that match, up to an arbitrary complex phase, the states given in eq. (6.60) and eq. (6.61). This is expressed mathematically as

$$\dot{\theta} = \arg\min_{\dot{\theta}}\left[\arg\min_{\alpha}\left[\left\|\frac{e^{-H\tau}|\psi_\theta\rangle}{\|e^{-H\tau}|\psi_\theta\rangle\|} - e^{i\alpha}\frac{\left|\psi_{\theta+\tau\dot{\theta}}\right\rangle}{\left\|\psi_{\theta+\tau\dot{\theta}}\right\|}\right\|^2\right]\right], \tag{6.62}$$

where the innermost arg min determines the arbitrary phase α that minimizes the distance. We remark that an alternative definition of this loss function would be to minimise the Fubini-Study metric, which would lead to

$$\dot{\theta} = \arg\min_{\dot{\theta}}\left[\mathrm{FS}(e^{-H\tau}\psi_\theta, \psi_{\theta+\tau\dot{\theta}})\right], \tag{6.63}$$

[14] The derivation of eq. (6.60) is obtained by linearizing both the numerator and the denominator in τ. In particular, we have

$$\left\|e^{-H\tau}|\psi_\theta\rangle\right\| = \|\psi_\theta\|\sqrt{\frac{\langle\psi_\theta|(\mathbb{I}-2\tau H+O(\tau^2))|\psi_\theta\rangle}{\langle\psi_\theta|\psi_\theta\rangle}} = \|\psi_\theta\|\sqrt{1-2\tau\langle H\rangle_\theta + O(\tau^2)}$$

And recalling that $(1-2x)^{-1/2} \approx 1+x+O(x^2)$ we can expand the state obtaining

$$\frac{e^{-H\tau}|\psi_\theta\rangle}{\|e^{-H\tau}|\psi_\theta\rangle\|} \approx \frac{(\mathbb{I}-\tau H)}{\sqrt{1-2\tau\langle H\rangle_\theta+O(\tau^2)}}\frac{|\psi_\theta\rangle}{\|\psi_\theta\|} = (\mathbb{I}-\tau H)(1-2\tau\langle H\rangle_\theta)|\nu_\theta\rangle + O(\tau^2) = (\mathbb{I}-\tau(H-\langle H\rangle_\theta))|\nu_\theta\rangle,$$

where $|\nu_\theta\rangle$ is the normalized state.

A similar procedure is employed to derive formula eq. (6.61), but some extra care must be taken in order to extract that complex phase at the beginning. Note that the reason why we want to extract that complex phase is to be able to match equivalent states up to a phase.

The derivation is different wether we have real parameters or complex-holomoprhic ansatze. First we remark that for both cases can Taylor expand a state with parameters $\theta+\tau\dot{\theta}$ around θ for small τ, keeping only leading terms in τ, leading to the expression

$$\left|\psi_{\theta+\tau\dot{\theta}}\right\rangle = |\psi_\theta\rangle + \tau\dot{\theta}_i\hat{\partial}_i|\psi_\theta\rangle = (\mathbb{I}+\tau\dot{\theta}_i\hat{\partial}_i)|\psi_\theta\rangle,$$

where we use Einstein's summation convention for repeated indices (if the expression was unclear to the reader, simply braket the left-hand side with $\langle x|$ and work through the Taylor expansion in τ). The expression is also valid for complex holomoprhic ansatze because $\hat{\partial}_i^\star|\psi_\theta\rangle = 0$ in that case.

Then, the denominator of eq. (6.61) is given by

$$\begin{aligned}\left\|\psi_{\theta+\tau\dot{\theta}}\right\| &= \|\psi_\theta\|\sqrt{\frac{\langle\psi_\theta|(\mathbb{I}+\tau\dot{\theta}_i\hat{\partial}_i)^\dagger(\mathbb{I}+\tau\dot{\theta}_i\hat{\partial}_i)|\psi_\theta\rangle}{\langle\psi_\theta|\psi_\theta\rangle}} \\ &= \|\psi_\theta\|\sqrt{1+\tau 2\,\mathrm{Re}[\theta_i^\star\,\dot{\langle}\partial_i\rangle_\theta]},\end{aligned}$$

and then, substituting into the normalized state we get

$$\begin{aligned}\frac{\left|\psi_{\theta+\tau\dot{\theta}}\right\rangle}{\left\|\psi_{\theta+\tau\dot{\theta}}\right\|} &\approx \frac{(\mathbb{I}+\tau\dot{\theta}_i\hat{\partial}_i)}{\sqrt{1+\tau 2\,\mathrm{Re}[\theta_i^\star\,\dot{\langle}\partial_i\rangle_\theta]}}\frac{|\psi_\theta\rangle}{\|\psi_\theta\|} \\ &= (\mathbb{I}+\tau\dot{\theta}_i\hat{\partial}_i)(1+\tau\,\mathrm{Re}[\theta_i^\star\,\dot{\langle}\partial_i\rangle_\theta] + i\tau\,\mathrm{Im}[\theta_i^\star\,\dot{\langle}\partial_i\rangle_\theta] - i\tau\,\mathrm{Im}[\theta_i^\star\,\dot{\langle}\partial_i\rangle_\theta])|\nu_\theta\rangle \\ &\approx (\mathbb{I}+\tau\dot{\theta}_i\hat{\partial}_i)(1+\tau\theta_i^\star\,\dot{\langle}\partial_i\rangle_\theta)(1-i\tau\,\mathrm{Im}[\theta_i^\star\,\dot{\langle}\partial_i\rangle_\theta]+O(\tau^2))|\nu_\theta\rangle \\ &\approx (\mathbb{I}+\tau\dot{\theta}_i\hat{\partial}_i)(1+\tau\theta_i^\star\,\dot{\langle}\partial_i\rangle_\theta)\exp\left[-i\tau\,\mathrm{Im}[\theta\star_i\langle\partial_i\rangle_\theta]\right] = e^{i\chi(\theta)}(\mathbb{I}+\tau\dot{\theta}_i^\star(\hat{\partial}_i - \left\langle\hat{\partial}_i\right\rangle_\theta)|\nu_\theta\rangle.\end{aligned}$$

If the step from the third to the fourth line is unclear, notice that in the last parenthesis we have the first order expansion of the matrix exponential, and as the expression is up to leading order in τ we can simply add the missing terms and complex the exponential.

with $\mathrm{FS}(\psi,\phi) = \arccos\left(\sqrt{F(\psi,\phi)}\right)$ where $F(\psi,\phi)$ is the normalized fidelity.

Substituting the expressions of the linearized states in eq. (6.62), we can easily see that the innermost $\arg\min$ leads to $\alpha = -\chi(\theta)$ and that overall we obtain the quadratic form in $\dot{\theta}$

$$\tau^2 \arg\min_{\dot{\theta}} \left[\left\| \left(\Delta\hat{H} - \dot{\theta}_i \Delta\hat{\partial}_i \right) |\nu_\theta\rangle \right\|^2 \right], \tag{6.64}$$

where we use the short-hand $\Delta H \equiv \hat{H} - \langle H \rangle_\theta$ and $\Delta\hat{\partial}_i \equiv \hat{\partial}_i - \langle \hat{\partial}_i \rangle_\theta$. Which expands to

$$\arg\min_{\dot{\theta}} \left[\langle \Delta\hat{H}^2 \rangle_\theta - \dot{\theta}_i \langle \nu_\theta | \Delta\hat{H}\Delta\hat{\partial}_i | \nu_\theta \rangle - \dot{\theta}_j^\star \langle \nu_\theta | \Delta\hat{\partial}_j^\dagger \Delta\hat{H} | \nu_\theta \rangle + \dot{\theta}_i \dot{\theta}_j^\star \langle \nu_\theta | \Delta\hat{\partial}_i^\dagger \Delta\hat{\partial}_j | \nu_\theta \rangle \right].$$

Note that so far, all the formulas written are always valid in the two cases of real-parameters or complex-holomorphic. Finally, to solve this quadratic form we must pick one of the two cases (even if the solution will have almost the same form).

Complex holomorphic wavefunction: In this case, we can use Wirtinger derivatives and interpret $\dot{\theta}$ and $\dot{\theta}^\star$ as two sets of linearly independent parameters. Then, we stationarize wrt variations in $\dot{\theta}^\star$ finding

$$0 = \frac{\partial}{\partial \dot{\theta}_j^\star} [...] = -\langle \nu_\theta | \Delta\hat{\partial}_j^\dagger \Delta\hat{H} | \nu_\theta \rangle + \dot{\theta}_i \langle \nu_\theta | \Delta\hat{\partial}_i^\dagger \Delta\hat{\partial}_j | \nu_\theta \rangle .$$

By defining the two objects

$$F_j \equiv \langle \nu_\theta | \Delta\hat{\partial}_j^\dagger \Delta\hat{H} | \nu_\theta \rangle = \frac{\langle \partial_j \psi_\theta | H | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} - \frac{\langle \psi_\theta | \hat{H} | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} \frac{\langle \partial_j \psi_\theta | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle}, \tag{6.65}$$

$$S_{ij} \equiv \langle \nu_\theta | \Delta\hat{\partial}_i^\dagger \Delta\hat{\partial}_j | \nu_\theta \rangle = \frac{\langle \partial_i \psi_\theta | \partial_j \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} - \frac{\langle \partial_i \psi_\theta | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} \frac{\langle \psi_\theta | \partial_j \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle}, \tag{6.66}$$

which we will call the *Variational Forces* (F_j) and the *Quantum Geometric Tensor* (S_{ij}). The former is equivalent at the expression of the energy we discussed in the previous section eq. (6.37), while the latter is the geometric tensor, which has several properties that we will discuss later. The stationarity condition is therefore equivalent to finding the vector of $\dot{\theta}_j$ that satisfies the linear system

$$\sum_j S_{ij} \dot{\theta}_j = F_i \tag{6.67}$$

which can be solved formally by inverting S and writing (in matrix notation) $\dot{\boldsymbol{\theta}} = S^{-1}\boldsymbol{F}$.

Real-parameter wavefunction: In this case $\theta = \theta^\star$, and the procedure is identical to before, only that we will set to zero the derivative wrt $\dot{\theta}$.

$$0 = \frac{\partial}{\partial \dot{\theta}_j} [...] = -2\,\mathrm{Re}\left[\langle \nu_\theta | \Delta\hat{\partial}_j \Delta\hat{H} | \nu_\theta \rangle + \dot{\theta}_i \langle \nu_\theta | \Delta\hat{\partial}_i \Delta\hat{\partial}_j | \nu_\theta \rangle \right].$$

from which the following definitions follow:

$$F_j \equiv \langle \nu_\theta | \Delta\hat{\partial}_j^\dagger \Delta\hat{H} | \nu_\theta \rangle = \mathrm{Re}\left[\frac{\langle \partial_j \psi_\theta | H | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} - \frac{\langle \psi_\theta | \hat{H} | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} \frac{\langle \partial_j \psi_\theta | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} \right], \tag{6.68}$$

$$S_{ij} \equiv \langle \nu_\theta | \Delta\hat{\partial}_i^\dagger \Delta\hat{\partial}_j | \nu_\theta \rangle = \mathrm{Re}\left[\frac{\langle \partial_i \psi_\theta | \partial_j \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} - \frac{\langle \partial_i \psi_\theta | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} \frac{\langle \psi_\theta | \partial_j \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} \right]. \tag{6.69}$$

The attentive reader will notice once again that the variational forces here coincide with the energy gradient, but we are considering the real part of the Quantum Geometric Tensor, discarding the imaginary part. A more detailed discussion of those variational principles, in connection with time evolution, can be found in [YEZ+19].

6.2.5 Geometric properties of the Quantum Geometric Tensor

The $N_{\text{parameters}} \times N_{\text{parameters}}$ matrix S defined as

$$S_{ij} = \langle \nu_\theta | \Delta\hat{\partial}_i^\dagger \Delta\hat{\partial}_j | \nu_\theta \rangle$$

is called *Quantum Geometric Tensor* because it is the **metric tensor** of the *variational manifold* that embeds the natural quantum metric. In simple terms, the natural metric of the space of wavefunction must be invariant under a global phase and changes in the norm of a wave-function. The metric of the Hilbert space is therefore the Fubini-Study[15] metric

$$d(\psi, \phi) = \arccos\Big(\sqrt{F(\psi,\phi)}\Big) = \arccos\sqrt{\frac{\langle\psi|\phi\rangle\,\langle\phi|\psi\rangle}{\langle\psi|\psi\rangle\,\langle\phi|\phi\rangle}}, \tag{6.70}$$

which can be used to define the distance between states[16]. When working with variational states parametrised by parameters $\theta \in \mathcal{M}$ in the variational manifold, we would like to embed the same metric between vectors of the variational manifold:

$$d(\theta, \gamma) = \arccos\sqrt{\frac{\langle\psi_\theta|\psi_\gamma\rangle\,\langle\psi_\gamma|\psi_\theta\rangle}{\langle\psi_\theta|\psi_\theta\rangle\,\langle\psi_\gamma|\psi_\gamma\rangle}}.$$

While this expression cannot be simplified further, by considering only infinitesimally distant objects $\gamma = \theta + \tau\dot{\theta}$, and keeping a first order expansion in τ, it is possible to show that the metric has the form

$$d(\theta, \theta + \tau\dot{\theta}) = \tau^2 \dot{\theta}_i^\star S_{ij} \theta_j,$$

which motivates the name.

It should be noted that, as the QGT determines the distance between two wavefunctions that differ by a small change of parameters, its rank provides an information on how degenerate the variational ansatz is for a particular set of parameters. In particular, a non-full-rank QGT signifies that there are some redundant parameters, and the kernel of the Geometric tensor will contain directions in the parameter space that do not change the physical wavefunction. When considering Neural-Network Quantum States, that is customary.

6.2.6 Connection to the Natural Gradient

Natural Gradient descent was introduced by Amari in a seminal 1999 paper [Ama98]. It is a technique that tries to improve on the *dumb* stochastic gradient descent by taking information of the natural metric of the space of distributions that are encoded variationally.

The idea is the following: standard gradient descent of a *loss function* $L(\theta)$ can be seen as updating the parameters in such a way to obtain the largest change in the loss function, while under the constraint that the step has unit norm in the euclidean variational space. This can be written as the solution $\Delta\theta$ to a local optimisation problem with Lagrange multiplier η^{-1}

$$\arg\max_{\Delta\theta} \left| |L(\theta + \Delta\theta) - L(\theta)| - \frac{1}{\eta}\left(\|\Delta\theta\|^2 - 1\right)\right|^2, \tag{6.71}$$

where the term $\eta^{-1}\|\Delta\theta\|^2$ weights large changes of the parameters negatively, effectively setting the learning rate η. This can be verified by linearizing the equation above in $\Delta\theta$, which leads to the solution[17]

$$\Delta\theta = -\frac{\eta}{2}\boldsymbol{\nabla} L(\theta), \tag{6.72}$$

[15] The Fubini-Study corresponds to the Bures metric when the state is pure.

[16] Do note that the Fubini-Study metric corresponds, lossely speaking, to the Fidelity. In fact, the Fidelity is not a valid metric because it does not satisfy the triangle inequality, but it is rather easy to see the $\arccos\sqrt{\cdot}$ as a way to *fix* the problem at large distances. Instead, at small distances, the two are essentially equivalent.

[17] This can be derived easily: consider the Taylor expansion of the loss $L(\theta + \Delta\theta) \approx L(\theta) + \Delta\theta_i \nabla_i L(\theta)$ (assuming real parameters. For complex parameters see the infobox a few pages back). Then, stationarizing the argument of the arg max we get

$$0 = \frac{d}{d\Delta\theta}\left[\Delta\Theta_i \nabla_i L(\theta) - \lambda \Delta\theta_i \Delta\theta_i\right] = \nabla_i L(\theta) - \frac{2}{\eta}\Delta\theta_i$$

from which easily follows eq. (6.71). The precise value of η can be determined by stationarizing it wrt to itself, and will give the constraint that $\|\Delta\theta\|^2 = 1$.

where η is a scale factor such that $\|\Delta\theta\|^2 = 1$.

$\|\Delta\theta\|^2$ is the Euclidean norm of a unit change of parameters. This can be explicitated as a quadratic form with the euclidean metric tensor $\epsilon_{ij} = \mathbb{I}_{ij}$

$$ds^2 = \Delta\theta_i \epsilon_{ij} \Delta_j.$$

Amari remarked that it is possible to replace the metric tensor in the constraint with the geometric tensor of the manifold of parametrised objects, which, in our case, would be the quantum geometric tensor.

Carrying out the calculations, it should be evident that the expression obtained for the Natural Gradient Descent is identical to the Stochastic Reconfiguration already discussed.

However, the meaning of the SR method is much deeper and more powerful than Natural Gradient itself.

6.2.7 Sampling the geometric tensor

The reader will recall that in the previous section we have shown that eq. (6.37) can be estimated efficiently by sampling from the $|\psi_\theta(x)|^2$ distribution (see eq. (6.43)). Likewise, we can use the same trick to estimate efficiently the Quantum Geometric Tensor: by considering[18] eq. (6.66), we can explicitate the identity within the brakets, and multiply and divide by $|\psi(x)|^2$ obtaining

$$\begin{aligned}
S_{ij} &= \sum_x \frac{\langle \partial_i \psi_\theta |\, x\rangle\langle x\, | \partial_j \psi_\theta\rangle}{\langle \psi_\theta | \psi_\theta\rangle} - \left(\sum_x \frac{\langle \partial_i \psi_\theta |\, x\rangle\langle x\, | \psi_\theta\rangle}{\langle \psi_\theta | \psi_\theta\rangle}\right)\left(\sum_x \frac{\langle \psi_\theta |\, x\rangle\langle x\, | \partial_j \psi_\theta\rangle}{\langle \psi_\theta | \psi_\theta\rangle}\right) \\
&= \sum_x \frac{|\psi_\theta(x)|^2}{\langle \psi_\theta | \psi_\theta\rangle} \frac{\langle \partial_i \psi_\theta |\, x\rangle\langle x\, | \partial_j \psi_\theta\rangle}{\langle \psi_\theta | x\rangle\, \langle x | \psi_\theta\rangle} - \left(\sum_x \frac{\langle \partial_i \psi_\theta | x\rangle}{\langle \psi_\theta | x\rangle} \frac{|\, \langle x | \psi_\theta\rangle\, |^2}{\langle \psi_\theta | \psi_\theta\rangle}\right)\left(\sum_x \frac{|\psi_\theta(x)|^2}{\langle \psi_\theta | \psi_\theta\rangle} \frac{\langle x\, | \partial_j \psi_\theta\rangle}{\langle x | \psi_\theta\rangle}\right) \\
&= \mathbb{E}_{|\psi_\theta(x)|^2}\left[\partial_i \log \psi^\star_\theta(x) \partial_j \log \psi_\theta(x)\right] - \mathbb{E}_{|\psi_\theta(x)|^2}\left[\partial_i \log \psi^\star_\theta(x)\right] \mathbb{E}_{|\psi_\theta(x)|^2}\left[\partial_i \log \psi_\theta(x)\right] \\
&= \mathbb{C}\mathrm{ov}_{|\psi_\theta(x)|^2}\left(\partial_i \log \psi^\star_\theta(x), \partial_j \log \psi_\theta(x)\right)
\end{aligned}$$

we remark that, like for the gradient of the energy, we had to multiply and divide by $|\psi_\theta(x)|^2$ which in principle might be zero. This means that this estimator for the Quantum Geometric Tensor potentially has a bias and/or a large variance, even if the natural gradient calculation might be less sensible to such issues.

6.2.8 Algorithmic complexity and implementation of gradients and geometric tensors

[18] We will now stop discussing both real-valued and complex-valued formulas to give a more compact discussion. However, everything we discuss is valid for both cases.

6.2.9 Complements and details

Asymptotical signal to noise ratio of the energy estimator

We want to show that

$$\mathrm{SNR}_{p_\theta(x)}\left[H_\theta^{\mathrm{loc}}(x)\right] \propto |E(\theta) - E_0|^{-1}, \tag{6.73}$$

when converging to the ground state. For simplicity, we will consider a state that is a combination of the ground state with a small component of another excited state

$$|\psi_\delta\rangle = |\psi_0\rangle + \delta\,|\psi_1\rangle\,, \tag{6.74}$$

where we consider $\hat{H}\,|\psi_i\rangle = \epsilon_i\,|\psi_i\rangle$ and $\langle\psi_i|\psi_j\rangle = \delta_{ij}$. This case should be general and I believe (even though i have not done it) that it should be possible to prove that more complicated cases where more excited states are present can be reconducted to this particular case. We will make the calculation in the limit of small δ to leverage linear expansions.

The energy, given δ, is

$$E(\delta) = \frac{\langle\psi_\delta|\hat{H}|\psi_\delta\rangle}{\langle\psi_\delta|\psi_\delta\rangle} = \epsilon_0\left(\frac{1+\delta^2 r}{1+\delta^2}\right) \approx \epsilon_0\left(1+(r-1)\delta^2\right) + O(\delta^4),$$

where we defined $r \equiv \epsilon_1/\epsilon_0$ for a more compact notation. The local estimator, instead, is

$$E^{loc}(x) = \frac{\langle x|H|\psi_\delta\rangle}{\langle x|\psi_\delta\rangle} = \epsilon_0\frac{\psi_0(x)+\delta r\psi_1(x)}{\psi_0(x)+\delta\psi_1(x)} \approx \epsilon_0\left(1+\delta(r-1)\frac{\psi_1(x)}{\psi_0(x)} + \delta^2(1-r)\frac{\psi_1^2(x)}{\psi_0^2(x)}\right) + O(\delta^3),$$

To compute the variance of the energy, we must also compute

$$\mathbb{E}_{|\psi_\delta(x)|^2}\left[\left|E^{loc}(x)\right|^2\right] \approx \epsilon_0^2\left[1+\delta^2(r^2-1)\right],$$

from which it follows that the variance is

$$\mathbb{V}\mathrm{ar}\left[E^{loc}(x)\right] \approx \epsilon_0^2\delta^2 r(r-2). \tag{6.75}$$

The signal to noise ratio can then be easily computed and it is

$$\mathrm{SNR}_{p(x)}\left[E^{loc}(x)\right] \approx \frac{\epsilon_0\left(1+(r-1)\delta^2\right)}{\epsilon_0\delta|r(r-2)|} \propto \delta^{-1}, \tag{6.76}$$

which diverges in the limit of $\delta \to 0$.

Chapter 7

The eletronic structure problem

In this chapter we will discuss what is probably the most important problem in quantum mechanics: electronic structure. This is relevant to virtually all of chemistry and material science, and solving it allows us to predict properties of everything we interact with in our daily life. But the problem is extremely complex: not only we'd like to deal with an Avogadro number $O(N^{23})$ of particles, which is factually impossible to handle numerically.

Most importantly, the equations we will discuss in this chapter are the rigorous mathematical foundations of the *classical chemistry* you have studied in high school. Indeed, for several systems quantum effects are relatively weak, and we can get a decent phenomenological understanding by means of some simple rules that can be motivated rigorously with what we'll discuss in this chapter.

However, a major limitation of classical chemistry is that it is 'phenomenological': it can explain some phenomena, but it cannot predict it. Most importantly, it cannot be used to compute the binding and activation energies of several molecules and reactions, and completely fails to explain some catalytic pathways. To compute those energy levels, unfortuantely, quantum mechanics must be used, and complicated equations must be solved. This is industrially important: if we want to find a catalyser for a reaction, we can either test millions of possible candidates, or we can try to use computational tools to predict which ones are more likely to work.

7.1 From the Molecular Hamiltonian to a Discrete Electronic Problem

In chemistry we work at the level of nuclei and electrons. Both are particles with a given mass, electric charge and spin. While the nucleus itself is composed by neutrons and protons, in chemistry we assume that it's simply a point particle. Only in the field of nuclear chemistry, where we are interested in much finer processes of heavier atoms, we would look at the inner structure.

We seek a quantum-mechanical description of a molecular system composed of nuclei and electrons. The total wavefunction $\Psi(\mathbf{R}, \mathbf{r})$ depends on all nuclear $\mathbf{R}$ and electronic $\mathbf{r}$ coordinates. In non-relativistic quantum mechanics (and atomic units), the full Hamiltonian reads

$$\hat{H}_{\text{tot}} = -\sum_I \frac{1}{2M_I}\nabla_I^2 - \sum_i \frac{1}{2}\nabla_i^2 + \sum_{I<J} \frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|} - \sum_{I,i} \frac{Z_I}{|\mathbf{R}_I - \mathbf{r}_i|} + \sum_{i<j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}. \tag{7.1}$$

where ∇_I^2 is the laplacian w.r.t. the position of a nuclei I and ∇_i^2 is the laplacian w.r.t. the position of the electron i. The last 3 terms represent the nucleus-nucleus, nucleus-electron and electron-electron Coulomb interaction.

We remark that this is the standard Hamiltonian in absence of spin-orbit couplings and magnetic fields. But those effects could be integrated into this Hamiltonian by adding additional terms. In particular, we remark that while the Hamiltonian eq. (7.1) only has operators that act on the position of electrons and nuclei, electrons and nuclei are spin-full particles! But as the Hamiltonian does not depend on it, the two parts of the wave-function (spin and spatial) factor.

7.1.1 Born–Oppenheimer separation

Because nuclei are much heavier than electrons, their motion is much slower. The Born–Oppenheimer (BO) approximation exploits this separation of scales by approximating the wave-function as

$$\Psi(\mathbf{R}, \mathbf{r}) \approx \Psi_{\text{nuc}}(\mathbf{R})\, \Psi_{\text{el}}(\mathbf{r}; \mathbf{R}), \tag{7.2}$$

where the electronic state is solved for fixed nuclear positions.

This partition the whole problem into two different sub-problems: a nuclear problem, which can often be treated effectively with strong approximations, and the electronic problem. The electronic hamiltonian $\hat{H}_{\text{elec}}(\mathbf{R}_I)$, which acts only on the electronic degrees of freedom, is defined to be

$$\hat{H}_{\text{elec}}(\mathbf{R}_I) = -\sum_i \frac{1}{2}\nabla_i^2 - \sum_{I,i} \frac{Z_I}{|\mathbf{R}_I - \mathbf{r}_i|} + \sum_{i<j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}. \tag{7.3}$$

I remark that while the Hamiltonian's structure depends on nuclear coordinates $\mathbf{R}_I$, it does not act on them. The electronic eigenvalue equation then becomes

$$\hat{H}_{\text{el}}(\mathbf{R})\Psi_{\text{el}}(\mathbf{r}; \mathbf{R}) = E(\mathbf{R})\Psi_{\text{el}}(\mathbf{r}; \mathbf{R}), \tag{7.4}$$

and $E(\mathbf{R})$ defines a potential energy surface for the nuclei.

7.1.2 Born-Oppenheimer molecular dynamics

When studying dynamical properties, a common approximation is to exploit the large mass ratio between nuclei and electrons. Because electrons have a smaller mass and therefore move on timescales much faster than the nuclei. We therefore assume that for each nuclear configuration $\mathbf{R}(t)$ the electrons relax instantaneously to the ground state of the electronic Hamiltonian,

$$\hat{H}_{\text{el}}(\mathbf{R})\Psi_{\text{el}}^{(0)}(\mathbf{r}; \mathbf{R}) = E(\mathbf{R})\Psi_{\text{el}}^{(0)}(\mathbf{r}; \mathbf{R}).$$

The nuclei can then be treated as classical particles evolving on the potential energy surface $E(\mathbf{R})$. The force acting on nucleus I is

$$\mathbf{F}_I(\mathbf{R}) = -\nabla_{R_I} E(\mathbf{R}), \tag{7.5}$$

leading to the classical molecular dynamics equations

$$M_I \frac{d^2\mathbf{R}_I}{dt^2} = -\nabla_{R_I} E(\mathbf{R}). \tag{7.6}$$

This framework is the basis of *Born–Oppenheimer molecular dynamics*: the electronic ground state is recomputed at each nuclear configuration, and nuclei move under the resulting forces. We will return to this point later.

This picture can break down when the electronic state cannot relax adiabatically to a single eigenstate, for example near avoided crossings or conical intersections, during ultrafast processes, or when several electronic states are energetically close. In such situations non-adiabatic effects become important and transitions between electronic states must be considered. Additionally, the picture of the nuclei being classical particles is sometimes too crude, and quantum effects must also be included.

7.1.3 Fermionic antisymmetry and spin structure

Electrons are indistinguishable particles that obey fermionic statistics[1]. As a consequence, the total electronic wavefunction must be antisymmetric under exchange of any two electrons:

$$\Psi_{\text{el}}(\dots, \mathbf{x}_i, \dots, \mathbf{x}_j, \dots) = -\Psi_{\text{el}}(\dots, \mathbf{x}_j, \dots, \mathbf{x}_i, \dots), \tag{7.7}$$

[1] Because identical particles are indistinguishable, observable physics must be invariant under exchange of two particles. The many-body wavefunction may therefore change at most by a phase under exchange, namely

$$\Psi(\dots, x_i, \dots, x_j, \dots) = e^{i\theta}\Psi(\dots, x_j, \dots, x_i, \dots),$$

as this leaves the observable Born-Amplitude invariant. Consistency requires this phase to be ± 1, corresponding to symmetric (bosonic) or antisymmetric (fermionic) wavefunctions. In non-relativistic quantum mechanics the nature of a particles must be postulated. However, within relativistic Quantum Field Theory, the Spin-Statistics theorem proves that all particles with half-spin (1/2, 3/2, etc..) always obey antisymmetric statistics, while particles with integer spin $(0, 1, \dots)$ obey Bosonic statistics.

where the coordinate

$$\mathbf{x} = (\mathbf{r}, s)$$

includes both spatial and spin variables. This antisymmetry ensures that physical observables, such as the Born Amplitude, are invariant under particle exchange.

Role of spin. Electron spin is an intrinsic two-level degree of freedom ($s = \uparrow, \downarrow$). In the absence of spin–orbit coupling and magnetic fields, the electronic Hamiltonian does not act on spin and commutes with the total spin operators. Eigenstates may therefore be chosen as simultaneous eigenstates of

$$\hat{S}^2\Psi = S(S+1)\Psi, \qquad \hat{S}_z\Psi = M_S\Psi. \tag{7.8}$$

In this regime, the spin part of the wavefunction can be treated independently from the spatial part. If spin–orbit interactions become important, spin is no longer separately conserved and this separation breaks down. The antisymmetry requirement, however, remains unchanged.

Pauli exclusion principle. A consequence of the antisimmetry of the wavefunction is that it must vanish if two electrons with the same spin occupy the same position

$$\Psi_{\text{el}}(\dots, \mathbf{x}, \dots, \mathbf{x}, \dots) = -\Psi_{\text{el}}(\dots, \mathbf{x}, \dots, \mathbf{x}, \dots) \qquad \Rightarrow \qquad \Psi_{\text{el}}(\dots, \mathbf{x}, \dots, \mathbf{x}, \dots) = 0 \tag{7.9}$$

This leads to the Pauli exclusion principle, which states that no two electrons can occupy the same quantum state. In practice, this allows two electrons to share the same spatial state only if their spin states differ.

7.1.4 Mean field states: Slater determinants

In spin systems, the most crude approximation we have looked at for a system is the mean-field state $\Psi(x_1, x_2, \dots) = \prod_{i=1}^{N} \phi_i(x_i)$. Here we'd like to generalise the concept of a mean-field product state to fermionic systems. Indeed, the product state written like that cannot be a physical state because it is not anti-symmetric. To make it antisimmetric, we can take the 'product state' and manually antisimmetrize it by summing over all possible permutations:

- For 2 electrons, the possible permutations of indices $(1, 2)$ are $p_1 = (1, 2)$ and $p_2 = (2, 1)$, with parity respectively $+$ and $-$. Therefore,

  $$\Psi(\mathbf{x}_1, \mathbf{x}_2) = \frac{1}{\sqrt{2}} \left(\phi_1(\mathbf{x}_1)\phi_2(\mathbf{x}_2) - \phi_1(\mathbf{x}_2)\phi_2(\mathbf{x}_1)\right),$$

  and one can easily verify that this wavefunction changes sign upon exchanging $\mathbf{x}_1$ and $\mathbf{x}_2$. The square root term at the beginning is merely there to ensure normalization.

- For 3 electrons, there are $3! = 6$ permutations. Antisymmetrization gives

  $$\begin{aligned}\Psi(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) = \frac{1}{\sqrt{6}} &\phi_1(\mathbf{x}_1)\phi_2(\mathbf{x}_2)\phi_3(\mathbf{x}_3) + \phi_1(\mathbf{x}_2)\phi_2(\mathbf{x}_3)\phi_3(\mathbf{x}_1) + \phi_1(\mathbf{x}_3)\phi_2(\mathbf{x}_1)\phi_3(\mathbf{x}_2) \\ &- \phi_1(\mathbf{x}_1)\phi_2(\mathbf{x}_3)\phi_3(\mathbf{x}_2) - \phi_1(\mathbf{x}_2)\phi_2(\mathbf{x}_1)\phi_3(\mathbf{x}_3) - \phi_1(\mathbf{x}_3)\phi_2(\mathbf{x}_2)\phi_3(\mathbf{x}_1),\end{aligned}$$

  where even permutations appear with a plus sign and odd permutations with a minus sign. The attentive student will immediately recognize that the simple equation above corresponds to the 3x3 determinant

  $$\Psi(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) = \frac{1}{\sqrt{3!}} \det \begin{vmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \phi_3(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \phi_3(\mathbf{x}_2) \\ \phi_1(\mathbf{x}_3) & \phi_2(\mathbf{x}_3) & \phi_3(\mathbf{x}_3) \end{vmatrix}$$

  .

- For N electrons, there are $N!$ permutations p of the index set $(1, 2, \dots, N)$. We define the antisymmetrized product (often called the *Slater determinant state*) as

  $$\Psi(\mathbf{x}_1, \dots, \mathbf{x}_N) = \frac{1}{\sqrt{N!}} \sum_{p \in S_N} \text{sgn}(p) \prod_{i=1}^{N} \phi_i\big(\mathbf{x}_{p(i)}\big),$$

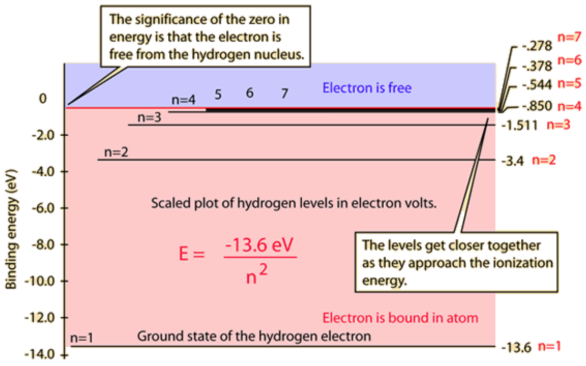


Figure 7.1: Energy levels for the hydrogen atom, as a function of the radial quantum number $n \in [1, 7]$. The system also allows for a continuum of energy levels for $E > 0$, which corresponds to a non-quantized (non-binded) nucleus-electron system.

where S_N is the permutation group on N elements and $\mathrm{sgn}(p) = +1$ (-1) for even (odd) permutations. This construction guarantees antisymmetry: exchanging any two particle coordinates flips the sign of Ψ.

The same wavefunction can be written compactly as a determinant [2]:

$$\Psi(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \frac{1}{\sqrt{N!}} \det \begin{pmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \cdots & \phi_N(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \cdots & \phi_N(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_N) & \phi_2(\mathbf{x}_N) & \cdots & \phi_N(\mathbf{x}_N) \end{pmatrix}.$$

We highlight that swapping two particles exchanges two rows of the matrix, which changes the sign, ensuring antisymmetry.

The simplest antisimmetric state, which generalizes a product state, is therefore called a *Slater Determinant* because it's amplitude is computed by means of a determinant. We remark that writing it as a determinant is not only a useful mathematical construct that allows for more easily manipulating those states, but is also computationally essential, as doing the same over $N!$ permutations would have complexity $O(N!)$ while computing the determinant has complexity $O(N^3)$.

This construction shows that antisymmetric many-electron states can be built from a set of one-electron functions $\{\phi_i(r)\}$, usually called **single particle orbitals**. Therefore an important point is **which single-particle orbitals to choose?**

7.1.5 Example: The hydrogen atom

Consider two hydrogen atoms very far apart, so that they essentially do not interact. The electronic states of each atom are then well approximated by the hydrogenic orbitals obtained by solving the Schroedinger's equation with a $V(r) = 1/r$ potential, which you did in your first year courses. The solution is a product of Laguerre polynomials for the radial part and spherical harmonics for the angular component. In general, we'll write them as

$$H_{\text{hydrogen}} \psi_{n\ell m}(r) = E_n \psi_{n\ell m}(r),$$

which are exact eigenfunctions of the hydrogen atom whose energy only depends on the radial part.

The electronic ground-state of an hydrogen atom will therefore be $\psi_{1,0,0}(r)$, which is also known as the $1s$ orbital, while higher excited states like $\psi_{2,1,0}$, $\psi_{2,1,\pm1}$, etc will have higher energy.

[2] This follows from the Leibniz formula for the determinant,

$$\det(A) = \sum_{p \in S_N} \mathrm{sgn}(p) \prod_{i=1}^{N} A_{i,p(i)},$$

which reproduces exactly the antisymmetrized sum above when $A_{i,j} = \phi_j(\mathbf{x}_i)$.

If now we consider the joint system with the two hydrogen atoms at positions R_1 and R_2 and two electrons, it is natural to think that if[3] $|R_1 - R_2| >> a_0$ the ground-state of the joint system should be the slater determinant of the two separate ground-states

$$\psi_{2H,\,gs}(r_1, r_2) = \det \begin{vmatrix} \phi_{100}(\mathbf{r}_1 - R_1) & \phi_{100}(\mathbf{r}_1 - R_2) \\ \phi_{100}(\mathbf{r}_2 - R_1) & \phi_{100}(\mathbf{r}_2 - R_2) \end{vmatrix}. \tag{7.10}$$

To shorten the notation, we usually refer to the the two *centered atomic orbitals* $\phi_{1s,1}(r) = \phi_{100}(\mathbf{r} - R_1)$ and $\phi_{1s,2}(r) = \phi_{100}(\mathbf{r} - R_2)$ and build the slater determinant out of those.

$$\psi_{2H,\,gs}(r_1, r_2) = \det \begin{vmatrix} \phi_{1s,1}(\mathbf{r}_1) & \phi_{1s,2}(\mathbf{r}_1) \\ \phi_{1s,1}(\mathbf{r}_2) & \phi_{1s,2}(\mathbf{r}_2) \end{vmatrix}. \tag{7.11}$$

Building a basis of many-electron states. But this state will be a 'good' ground-state only if th two atoms are sufficiently far. How can we improve this approximation as the two atoms are closer and closer and $|R_1 - R_2| \approx a_0$, and the two atoms will hybridize?

Given a set of orbitals $\{\phi_p\}$, one can construct many Slater determinants by occupying different subsets of these orbitals. For example, in the 2H molecule above one could consider the centered atomic orbitals $\{1s, 2s\}$ for each atom, so for a total of 4 atomic orbitals $\{\phi_{1s,1}, \phi_{1s,2}, \phi_{2s,1}, \phi_{2s,s}\}$.

Then, one builds different slater determinants, assuming that different orbitals are occupied, obtaining something that looks like

$$\Psi = \sum_I c_I \, |D_I\rangle \, . \tag{7.12}$$

This is effectively a variational wave-function with variational parameters c_I. If one considers all possible determinants one recover the full hilbert space and essentially a state vector, but in general only few determinants are kept in the sum, which limits the expressive power of the ansatz.

In our example of $2H$ molecule, we could for example consider only 3 determinants where the occupied orbitals are $(1s, 1s)$, $(1s, 2s)$ and $(2s, 1s)$, and ignore the $(2s, 2s)$ occupation as unlikely. In that case the wavefunction will be

$$\psi_{2H,\,gs}(r_1, r_2) = c_{1s,1s} \det \begin{vmatrix} \phi_{1s,1}(\mathbf{r}_1) & \phi_{1s,2}(\mathbf{r}_1) \\ \phi_{1s,1}(\mathbf{r}_2) & \phi_{1s,2}(\mathbf{r}_2) \end{vmatrix} + c_{1s,2s} \det \begin{vmatrix} \phi_{1s,1}(\mathbf{r}_1) & \phi_{2s,2}(\mathbf{r}_1) \\ \phi_{1s,1}(\mathbf{r}_2) & \phi_{2s,2}(\mathbf{r}_2) \end{vmatrix} + c_{2s,1s} \det \begin{vmatrix} \phi_{2s,1}(\mathbf{r}_1) & \phi_{1s,2}(\mathbf{r}_1) \\ \phi_{2s,1}(\mathbf{r}_2) & \phi_{1s,2}(\mathbf{r}_2) \end{vmatrix}. \tag{7.13}$$

Choice of orbital sets. The quality and efficiency of this expansion depend crucially on two things: (i) the choice of orbitals and (ii) the number of determinants, corresponding to the number of possible combinations of those orbitals.

Importantly, a good orbital basis can dramatically reduce the number of determinants needed (think for example: if instead of the hydrogenic orbitals i was using as orbitals some random gaussians, i would need more orbitals to express the hydrogenic ones).

In molecular systems, orbitals are typically expanded in localized atomic-centered functions called *atomic orbitals*. These provide a flexible and chemically intuitive basis for constructing molecular orbitals and many-electron states.

We therefore turn to the question of how such one-electron basis functions are chosen.

7.2 Beyond hydrogenic orbitals

Hydrogenic orbitals are exact eigenfunctions of the Coulomb potential created by a single proton. They provide a natural starting point and a useful qualitative language (shells, $s/p/d$ orbitals), but they are nor accurate nor reasonable for other atoms with more protons and electrons. In heavier atoms each electron feels not only the nuclear attraction but also the repulsion from all other electrons. This electron–electron interaction screens the nuclear charge and modifies the effective potential experienced by an electron.

[3] a_0 is the bohr radius, the radius of the hydrogen atom's ground-state.

A useful physical picture is that of an *effective one–electron problem*

$$\left[-\frac{1}{2}\nabla^2 + V_{\text{eff}}(r)\right]\phi_i(\mathbf{r}) = \varepsilon_i\phi_i(\mathbf{r}), \tag{7.14}$$

where V_{eff} includes the nuclear attraction and an average repulsion from the other electrons. Close to the nucleus the electron feels almost the full nuclear charge Z, while far away the inner electrons screen the nucleus and the potential decays approximately as $-1/r$. This leads to orbitals that resemble hydrogenic ones but with contracted inner shells and expanded outer shells.

In practice, the effective potential is not known a priori and must be determined self–consistently. This is the idea behind mean–field methods such as Hartree–Fock and Kohn–Sham density functional theory, which determine orbitals by solving an effective one–electron equation whose potential depends on the electron density itself. The resulting orbitals are often called *atomic orbitals* or *Hartree–Fock orbitals*. They retain the familiar angular structure given by spherical harmonics,

$$\phi_{n\ell m}(\mathbf{r}) = R_{n\ell}(r)\, Y_{\ell m}(\theta,\varphi),$$

but their radial functions $R_{n\ell}(r)$ differ significantly from the hydrogenic ones.

Basis–set representation. To compute $R_{n\ell}(r)$ efficiently, and to later used it effectively, we must be able to compute it rapidly. While for the particular case of the Hydrogen atom we knew the form to be a Laguerre polynomial, the form of general atomic orbitals is generally not know. The orbitals are therefore written in a given *finite functional basis* of radial functions $\chi_u(\mathbf{r})$

$$\phi_i(\mathbf{r}) = \sum_\mu C_{\mu i}\,\chi_\mu(\mathbf{r}), \tag{7.15}$$

where $\{\chi_\mu\}$ are predefined basis functions and the coefficients $C_{\mu i}$ are optimized variationally. This converts the continuous problem of eq. (7.14) into a finite matrix eigenvalue problem.

The basis functions are chosen to resemble atomic orbitals and to reproduce the correct qualitative behavior (a cusp at the nucleus and exponential decay at large distance). Two important families of basis functions are widely used in quantum chemistry.

7.2.1 Slater–type orbitals (STOs).

Slater–type orbitals mimic the exact hydrogenic radial behavior,

$$\chi(\mathbf{r}) \propto r^{n-1}e^{-\zeta r}Y_{\ell m}(\theta,\varphi), \tag{7.16}$$

where ζ is an effective long-range screening parameter. They reproduce the correct cusp at the nucleus ($r \to 0$) and the exponential decay of real atomic orbitals. STOs can describe atoms accurately with relatively small bases (few functions).

However, integrals involving STOs cannot be evaluated analytically in closed form for general molecular geometries, making them computationally expensive as they require that every matrix element of the operator be computed with MC integration or other analytical methods.

7.2.2 Gaussian–type orbitals (GTOs).

In most practical calculations STOs are replaced by Gaussian–type orbitals,

$$\chi(\mathbf{r}) \propto x^a y^b z^c\, e^{-\alpha r^2}, \tag{7.17}$$

with non–negative integers a, b, c determining the angular character. Gaussian functions are apparently *bad*: they do *not* reproduce the correct cusp at the nucleus and decay too rapidly at large distance. But products of Gaussians centered on different atoms are gaussians, their derivatives are known analytically as well as their moments, making integral evaluation extremely fast and precise.

To recover the correct physical shape, several Gaussian functions are combined to approximate a single Slater orbital,

$$\chi_{\text{STO}} \approx \sum_{k=1}^{K} d_k\, e^{-\alpha_k r^2}. \tag{7.18}$$

This idea leads to *contracted Gaussian basis sets*. For instance, the well–known STO–3G basis represents each Slater orbital by a contraction of three Gaussians.

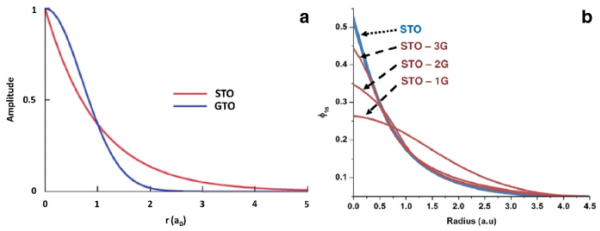


Figure 7.2: (**left**) Comparison between a Slater Type Orbital (STO) and a gaussian type orbital (GTO). (**right**) comparison between an STO and its gaussian-based approximation.

Molecular orbitals Once atomic basis functions are chosen, molecular orbitals are constructed as linear combinations of basis functions centered on all nuclei,

$$\psi_i(\mathbf{r}) = \sum_{\mu} C_{\mu i}\, \chi_\mu(\mathbf{r} - \mathbf{R}_\mu). \tag{7.19}$$

This linear–combination–of–atomic–orbitals (LCAO) picture provides a bridge between chemistry intuition (localized atomic orbitals) and quantum mechanics (delocalized molecular orbitals). These orbitals are then used to build Slater determinants and correlated many–electron wavefunctions.

The accuracy of an electronic structure calculation is therefore controlled by two ingredients: the quality of the many–electron ansatz (single determinant, multi–determinant, correlated methods) and the flexibility of the chosen basis set. Increasing the basis size systematically improves the description and allows convergence toward the complete basis–set limit.

7.3 Constructing a discrete Hamiltonian (2nd quantisation)

Up to now we have described electrons through a continuous wavefunction $\psi(\mathbf{x}_1, \ldots, \mathbf{x}_N)$. However, once a set of single–particle orbitals $\{\phi_p(\mathbf{x})\}$ is chosen, a powerful simplification occurs.

Any antisymmetric N–electron wavefunction can be expanded as a linear combination of Slater determinants built from these orbitals,

$$\psi(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \sum_I c_I\, D_I(\mathbf{x}_1, \ldots, \mathbf{x}_N), \tag{7.20}$$

where each D_I corresponds to occupying a specific set of orbitals.

If the set $\{\phi_p\}$ is complete, this expansion is exact. With a finite basis, it becomes a controlled approximation whose accuracy improves as more orbitals are included. Therefore, choosing a basis of orbitals transforms the continuous many–electron problem into a finite linear space spanned by determinants. The problem becomes algebraic.

Occupation representation. A determinant is fully specified by listing which orbitals are occupied. For example, if we have four spin–orbitals,

$$|1, 0, 1, 0\rangle$$

denotes the determinant in which orbitals 1 and 3 are occupied. Instead of representing wavefunctions in the position basis, it is common to encode wavefunctions in the basis of determinants (known as *Fock space* in this context). So, in a sense, the wavefunction of a system with N electrons can be written either as

$$|\psi\rangle = \int_{\mathbb{R}^{3N}} d\mathbf{x}_1 d\mathbf{x}_2 \ldots d\mathbf{x}_N\, \psi(\mathbf{x}_1 \mathbf{x}_2 \ldots \mathbf{x}_N)\; |\mathbf{x}_1 \mathbf{x}_2 \ldots \mathbf{x}_N\rangle \tag{7.21}$$

or in the Fock basis

$$|\psi\rangle = \sum_{n_1 \in \{0,1\}} \sum_{n_2 \in \{0,1\}} \cdots \sum_{n_M \in \{0,1\}} c_{n_1 n_2 \ldots n_M}\, |n_1, n_2 \ldots n_M\rangle\,. \tag{7.22}$$

Because electrons are fermions, each spin–orbital may be occupied by *at most one electron*,

$$n_p \in \{0, 1\}.$$

This description is called the *occupation-number basis*. Each occupation vector corresponds exactly to one Slater determinant. This point of view allows us to manipulate many–electron states without writing explicit determinants.

7.3.1 Creation and destruction operators

We now want to obtain a representation like that of creation and destruction operators for the harmonic oscillator, only that it will be for creating and destroying electrons in given orbitals.

We now introduce an operator that adds an electron in orbital ϕ_i.

Conceptually, given an antisymmetric N–electron wavefunction ψ_N, the operation of adding an electron in orbital ϕ_i produces the antisymmetrized $(N+1)$–electron state

$$\hat{c}_i^\dagger \psi_N \;\propto\; \mathcal{A}[\phi_i(\mathbf{x}_{N+1})\psi_N(\mathbf{x}_1, \ldots, \mathbf{x}_N)]\,, \tag{7.23}$$

where $\mathcal{A}$ denotes antisymmetrization over all particle coordinates.

Similarly, the annihilation operator $\hat{c}_i$ removes an electron from orbital i if present.

These operators act on occupation states as

$$\hat{c}_i^\dagger |\ldots, n_i = 0, \ldots\rangle = |\ldots, n_i = 1, \ldots\rangle, \tag{7.24}$$

$$\hat{c}_i |\ldots, n_i = 1, \ldots\rangle = |\ldots, n_i = 0, \ldots\rangle. \tag{7.25}$$

They play a role analogous to ladder operators of the harmonic oscillator: they raise and lower occupation numbers.

Anticommutation The central algebraic property of fermionic operators follows directly from antisymmetry. Consider applying two creation operators to an N–electron state.

First apply $\hat{c}_j^\dagger$, then $\hat{c}_i^\dagger$.

- $\hat{c}_j^\dagger$ adds one electron in orbital j and antisymmetrizes over all $N+1$ coordinates.
- $\hat{c}_i^\dagger$ then adds one electron in orbital i and antisymmetrizes over all $N+2$ coordinates.

The resulting state is the fully antisymmetrized $(N+2)$–electron wavefunction built from

$$\phi_i(\cdot)\,\phi_j(\cdot)\,\psi_N(\cdots).$$

Now observe the key point:

in a fully antisymmetrized wavefunction, exchanging the labels of the particle occupying orbital i and the one occupying orbital j changes the sign.

But performing the two creations in the opposite order corresponds exactly to exchanging these two particles.

Therefore, for any ψ_N,

$$\hat{c}_i^\dagger \hat{c}_j^\dagger \psi_N = -\,\hat{c}_j^\dagger \hat{c}_i^\dagger \psi_N, \tag{7.26}$$

which implies

$$\boxed{\{\hat{c}_i^\dagger, \hat{c}_j^\dagger\} = 0}. \tag{7.27}$$

As a special case, taking $i = j$ gives

$$(\hat{c}_i^\dagger)^2 = 0,$$

meaning that adding two identical fermions to the same orbital gives zero. This is the Pauli exclusion principle in operator form.

A similar argument shows

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

These relations encode fermionic antisymmetry in a compact algebraic form.

vacuum Let $|0\rangle$ denote the vacuum (no electrons). A determinant occupying orbitals $p_1, \ldots, p_N$ is constructed as

$$|D\rangle = \hat{c}^\dagger_{p_1} \hat{c}^\dagger_{p_2} \cdots \hat{c}^\dagger_{p_N} |0\rangle. \tag{7.28}$$

Exchanging the order of two operators changes only the sign, reflecting the antisymmetry of the determinant.

7.3.2 Second quantised electronic Hamiltonian

The electronic Hamiltonian is naturally written in the coordinate representation. For N electrons,

$$\hat{H} = \sum_{i=1}^{N} \hat{h}(i) + \frac{1}{2} \sum_{i \neq j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}, \tag{7.29}$$

where $\hat{h}(i)$ contains the kinetic energy and the attraction to the nuclei, and the second term describes Coulomb repulsion.

To obtain a discrete representation, we introduce an orthonormal set of spin-orbitals $\{\phi_p(\mathbf{x})\}$ and insert the resolution of the identity in the one-particle Hilbert space,

$$\hat{\mathbb{I}} = \sum_{p} |\phi_p\rangle\langle\phi_p| \tag{7.30}$$

This states that the orbitals form a complete basis for the single-electron space.

One–electron term. Consider the operator $\hat{h}(i)$ acting on electron i. Inserting the identity on both sides gives

$$\hat{h}(i) = \sum_{pq} |\phi_p(i)\rangle \underbrace{\langle\phi_p|\hat{h}|\phi_q\rangle}_{h_{pq}} \langle\phi_q(i)|, \tag{7.31}$$

where the matrix elements

$$h_{pq} = \int d\mathbf{x}\; \phi^*_p(\mathbf{x})\, \hat{h}\, \phi_q(\mathbf{x}) \tag{7.32}$$

are the familiar one-electron integrals.

Thus the continuous operator $\hat{h}(i)$ is replaced by a matrix acting on orbital labels.

Two–electron interaction. We proceed similarly for the Coulomb interaction. Inserting the identity for each electron,

$$\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} = \sum_{pqrs} |\phi_p(i)\phi_q(j)\rangle \underbrace{\langle pq|\frac{1}{r_{12}}|rs\rangle}_{(pq|rs)} \langle\phi_r(i)\phi_s(j)|, \tag{7.33}$$

with the two-electron integrals

$$(pq|rs) = \int d\mathbf{x}_1 d\mathbf{x}_2\; \frac{\phi^*_p(\mathbf{x}_1)\phi_q(\mathbf{x}_1)\phi^*_r(\mathbf{x}_2)\phi_s(\mathbf{x}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|}. \tag{7.34}$$

Bibliography

[Ama98] Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

[CPGSV21] J. Ignacio Cirac, David Pérez-García, Norbert Schuch, and Frank Verstraete. Matrix product states and projected entangled pair states: Concepts, symmetries, theorems. *Reviews of Modern Physics*, 93(4), December 2021.

[NW06] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*, volume 2 of *Springer Series in Operations Research and Financial Engineering*. Springer New York, 2006.

[Or 14] Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349:117–158, October 2014.

[Sch05] U. Schollwöck. The density-matrix renormalization group. *Reviews of Modern Physics*, 77(1):259–315, April 2005.

[Sch11] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, January 2011.

[Vid03] Guifré Vidal. Efficient classical simulation of slightly entangled quantum computations. *Physical Review Letters*, 91(14):147902, 2003.

[Vid04] Guifré Vidal. Efficient simulation of one-dimensional quantum many-body systems. *Physical Review Letters*, 93(4):040502, 2004.

[Whi92] Steven R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69:2863–2866, November 1992.

[Whi93] Steven R. White. Density-matrix algorithms for quantum renormalization groups. *Phys. Rev. B*, 48:10345–10356, October 1993.

[YEZ+19] Xiao Yuan, Suguru Endo, Qi Zhao, Ying Li, and Simon C. Benjamin. Theory of variational quantum simulation. *Quantum*, 3:191, October 2019.