University of Trento
Department of Physics

PhD Thesis

~ · ~

Academic Year 2024–2025

Quantum Algorithms for many-body systems simulation

Supervisor
Prof. Alessandro Roggero

PhD Student
Luca Vespucci

Final examination date: October 10, 2025

Abstract

In this thesis, two novel quantum algorithms are developed and presented, both of which focus on practical feasibility for near-term quantum hardware and the physics of many-body systems. The first project exploits the Hubbard-Stratonovich transformation to obtain an estimate for many-body quantum observables, proposing a method that could lead to a simplification of quantum circuits by reducing the need for multi-qubit gates. This approach is particularly suited to devices with limited qubit coherence times and gate fidelity, though its applicability is mainly limited to observables that can be mapped onto quadratic forms. The second project implements a Givens rotation-based decomposition for the Quantum METTS (Minimally Entangled Typical Thermal States) algorithm, trying to extend its applicability beyond finite-range Hamiltonians. This method is applied to a small test system of 4 supernovae neutrinos and demonstrates potential for efficient thermal state sampling in systems with high entanglement growth. Although neither method outperforms existing techniques, both contribute new perspectives and emphasize the importance of mathematical decomposition in optimizing quantum algorithms for near-term devices.

Contents

Introduction

During recent years, great interest in the field of *quantum computing* has developed across both academic and industrial areas, motivated by the growing recognition that quantum mechanical systems could be at the base of new paradigms of computation. While classical computers rely on deterministic operations over binary variables, known as *bits*, quantum computers manipulate information that is encoded in quantum bits (*qubits*), these being two-level quantum systems governed by the laws of quantum mechanics. The quantum nature of qubits enables access to phenomena such as *superposition*, *entanglement*, and *interference*, which gives quantum computers the potential to outperform classical machines for a broad class of computational problems.
The advantages that quantum computing offers are far-reaching and can be of help in many different disciplines. Some of the main applications are the following:

- **Quantum simulation**, in particular of many-body quantum systems, chemical compounds and complex materials;
- **Cryptanalysis**, the most relevant result consisting of Shor's algorithm [1], a theoretically efficient way of factoring large integer numbers, which, if it could be efficiently implemented, would threaten the security of the best cryptography schemes used nowadays;
- **Optimization problems**, typically mapped on analog quantum computers and tackled by quantum heuristic algorithms;
- **Quantum machine learning**, which promises to help with the acceleration of high-dimensional data analysis.

Despite quantum computing today still being at a developmental stage, there is strong evidence that it could contribute to speeding up the best classical algorithms even exponentially, especially when dealing with problems characterized by state spaces having an exponentially large dimension or involving explicit quantum dynamics.
It was *Richard Feynman* (1982) [2] and *David Deutsch* (1985) [3] that for the first

time proposed the idea that quantum systems could be more efficiently simulated by another quantum system, rather than a classical computer, so to exploit the fact that both systems obeyed the same rules of quantum mechanics. This key insight gave rise to the field of *quantum information theory*, which soon introduced quantum bits as the fundamental unit of quantum information [4]. The next decade saw the development of the most famous and outstanding results in quantum computing: in 1994 *Shor's algorithm* [1] for the factorization of prime numbers was developed, which demonstrated an exponential speedup over classical algorithms, while in 1996 Grover published his *search algorithm* [5], achieving a quadratic improvement for unstructured search problems over classical alternatives.
To these results followed the formulation of the *quantum circuit model*, as well as the introduction of *quantum error correction codes* and the theoretical proof of *fault tolerance* with the assumption of realistic noise models [6–8]. All these results together contribute to giving strong evidence that large-scale quantum computation is indeed physically realizable, at least in principle. More recently, a great variety of quantum computing architectures and platforms were explored, among which *superconducting qubits* [9], *trapped ions* [10], *photonic systems* [11], and *neutral atoms* [12], each demonstrating different (yet always increasing) levels of control over qubit coherence and gate fidelity.
Despite these incredibly noticeable results, there is still much to improve in nowadays quantum computing and milestones to achieve: just to mention some examples, qubit *coherence times* must be improved and *gate error rates* reduced, in order to guarantee for meaningful and reliable computations; quantum hardware have to reach bigger scales and a more complete connection topology, so that enough resources become available; also, *fault-tolerant quantum computation* should be achieved. Furthermore, lots of efforts must be put into finding new classes of problems that can be efficiently mapped and solved by quantum means.
With this thesis, I offer my own contribution to this ongoing scientific effort, as I investigate two distinct yet related approaches to quantum algorithm development. Both these projects have a focus on the context of many-body system physics, and specifically address the problem keeping in mind practical feasibility on near-term quantum hardware.
The first project is based on the exploitation of the *Hubbard-Stratonovich transformation* [13, 14] as a way of estimating many-body quantum observables. Hubbard-Stratonovich transformation is an exact mathematical identity usually employed in many-body statistical physics to separate the contributions of different interacting terms appearing in the system Hamiltonian [15]. Here, it is exploited in a similar fashion, but with the novel intent of re-expressing the expectation value of certain observables as an average over auxiliary fields and possibly few-qubit gates only, removing the need for the execution of costly multi-qubit operators.

This approach is not fully universal, but can be applied to all those observables that can be mapped onto quadratic forms, and proposes as a method to reduce the complexity of the circuit required for the computation. These properties make the method particularly suitable to those devices characterized by more limited qubit coherence times and gate fidelity. Despite the applicability limitations of the developed algorithm and the fact that it may not offer particular advantages in performance when compared to other standard methods, it still constitutes a distinct route to the measurement strategy and can claim a rather simple quantum implementation, making it a useful tool for exploring alternative formulations of quantum estimation techniques.
The second project consists of a specific implementation of the *Quantum METTS (Minimally Entangled Typical Thermal States)* algorithm [16], which is a method allowing for the efficient sampling of thermal expectation values through the construction of a statistically representative ensemble made of low-entanglement pure states. The main idea is that of decomposing the unitaries involved in the QMETTS procedure on a basis of *Givens rotations*, which are a class of two-level unitary transformations known to be universal for decomposing number-conserving unitaries [17]. The motivation behind the development of this approach is that of extending the applicability of QMETTS beyond the class of finite-range k-local Hamiltonians: we tested the method following an heuristic perspective, applying it to a small system of interacting neutrinos in the plane wave approximation, described by the usual *all-to-all* Hamiltonian (see Eq. (3.19) and [18]) and characterized by an high entanglement growth, properties which makes them challenging for traditional analyses. The Givens rotations formulation relies on the neutrino Hamiltonian symmetries to circumvent the scalability issues associated with densely interacting Hamiltonians, and its formulation associated to the employment of the SWAP network provides an adaptable way to implement the algorithm on hardware having arbitrary qubit connectivity. The small size of the test system was mainly chosen to simplify the computation, since it only serves as a minimal benchmark for the method's feasibility, with no intent of discovering new physical insights or testing model limits. I stress that we do not seek to give a deep analysis of the algorithm performance, rather we try to show that Givens-rotation-based decompositions can have a useful impact for thermal state sampling in systems beyond locally interacting models.
Together, these two projects contribute novel perspectives on the design and implementation of quantum algorithms for observable estimation and system simulation. While neither method claims superiority over existing approaches, both expand the repertoire of available techniques and highlight the importance of mathematical structure and decomposition strategies in optimizing quantum computation for near-term devices.

Chapter 1

Quantum computing basics

1.1 Quantum computing introduction

Quantum computing proposes itself as an extension to classical computation, being formulated for the exploitation of phenomena which are intrinsic of quantum mechanics, like *superposition* and *entanglement*, so to enable the processing of information beyond what can be achieved with traditional bits. Classical computers, in fact, manipulate binary states which are deterministic, while quantum computers operate on *qubits*: these are two-level quantum systems whose states can be used to represent multiple classical states simultaneously, potentially offering exponential speedups for certain specific problems.
Today, applications of quantum computing are very broad and various, including quantum simulation of complex physical systems, optimization, cryptography, and machine learning. The main challenges of this computation paradigm include the development of efficient algorithms, managing noise influence and errors in hardware implementations, as well as designing efficient measurement protocols for the extraction of useful information.
This chapter outlines the fundamental principles of quantum computation, contrasts them with classical approaches, and discusses core topics relevant to this thesis, including quantum algorithms, circuit models, and observable estimation techniques.

1.1.1 Comparison with classical computing

To give a basic understanding of the new aspects introduced by quantum computing and appreciate its potential, it is essential to first compare it with the well-established paradigm of classical computation.
The fundamental working principle of the latter consists of the manipulation of binary variables called *bits*, the basic units of information in all conventional com-

puting devices, each of which can be in one of two possible states, either 0 or 1, implying that the single-bit state belongs to the space $\mathbb{Z}_2 \doteq \{0, 1\}$. A string of N classical bits can represent one out of 2^N possible configurations at any time, the ensemble of these possible realizations constituting the set of possible states for the collection of N bits. Computations are then performed by applying sequences of logical operations, some of the most frequent being the AND, OR, and NOT operations, on these bitstrings, where each operation is mathematically described by a Boolean function f taking a bitstring of a certain length i as an input and transforming it to a new bitstring, possibly having a different length j

$$f : \{0, 1\}^i \to \{0, 1\}^j. \tag{1.1}$$

These operations are called *logical gates*, and are usually implemented by decomposing them on a set of elementary and possibly universal logic gates, which are represented by different graphic symbols connected by lines representing input and output bits (an example in Fig. 1.1).

Figure 1.1: Examples of typical elementary classical logic gates: On the left, an AND gate is represented, characterized by two input bits and a single output: it acts by setting the output bit state to 1 whenever both input bits states are 1, otherwise it sets the output bit state to 0. On the right, there is a NOT gate, having only one input and one output bits: it acts by setting the output bit state to the opposite value than the input one.

Complex classical algorithms are consequently realized as compositions of such gates, forming *digital circuits*: these are represented by a collection of the symbols corresponding to the involved logic gates, connected by lines representing input and output bits (an example is given in Fig. 1.2).

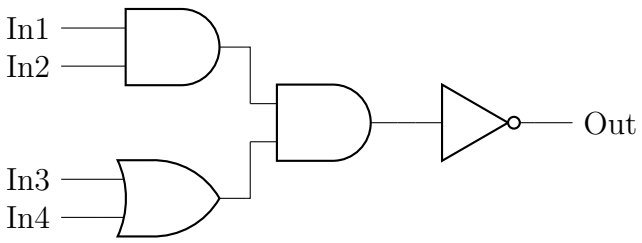


Figure 1.2: Example of the representation of a generic random classical logic circuit, realized as a composition of the elementary AND, OR and NOT gates, having four input bits and one output bit.

Quantum computation instead employs *qubits*, i.e. two-level quantum systems that can individually exist in a coherent linear superposition of the two states $|0\rangle$ and $|1\rangle$ (see Section 1.2.1). The state of a single qubit is represented by a unit vector in a two-dimensional complex Hilbert space (see Fig. 1.3 for a graphical representation), and the state space of a set of N qubits resides in a 2^N-dimensional space. It is evident that the number of classical resources needed for a full description of the state of a group of qubits scales exponentially with the number of qubits included in the group: this exponential scaling is exactly the main difference between quantum and classical computing, and what enables, in principle, quantum computers to represent and process information in ways that are fundamentally inaccessible to classical devices.
Quantum gates are the generalization to the quantum realm of classical gates, corresponding to operations which map an initial state to a final one: as such, they correspond to unitary transformations and are realized by letting the qubits evolve under controlled interactions (see Section 1.2.2). Importantly, quantum operations must preserve coherence and allow for the development of many purely quantum phenomena like *interference* and *entanglement*, which can be regarded as resources enabling quantum algorithms to outperform their classical counterparts for specific computational tasks. In general, in fact, quantum computers do not accelerate all problems indiscriminately, rather they offer computational advantages when addressing specific classes of problems, particularly those involving linear algebra, quantum simulation, and high-dimensional optimization.

1.1.2 Quantum computers advantages for simulating complex many-body quantum systems

Classical computers face an exceptional challenge in simulating the dynamics of complex quantum systems, since the needed computational resources, both in time and memory, for addressing such problems scale exponentially with the number of particles in the system and soon become unmanageable. This exponential scaling issue becomes more and more critic the more entangled are the states of the system under analysis [19].
This consideration lies at the ground of the motivations behind quantum computing, which is exactly suited to deal with this exponential scaling by relying on intrinsic quantum resources. Neutrino flavor evolution in dense astrophysical environments like supernovae and neutron star mergers is a typical example of such a highly entangled quantum many-body system: these systems are characterized by nonlinear interactions, which contribute to a fast and wide entanglement growth and to the build-up of intricate quantum correlations, making their simulation

extremely challenging for classical computers.
Similarly, thermal quantum systems require the realization and handling of mixed quantum states, a task that becomes quickly intractable on classical machines as system size starts to increase.
On the contrary, quantum computers propose themselves as a natural platform to address these challenges [2, 20], since they encode information directly into quantum states, which are naturally subject to the same behavior of the systems to be simulated: in principle, one could map the state of each particle to that of a suitable group of qubits, so that the needed quantum resources would scale just linearly with the number of particles to be simulated. For example, an N $\frac{1}{2}$-spin particle system can be represented with exactly N qubits, whereas a classical computer must handle 2^N bits to represent the corresponding full state space.
Furthermore, the realization of specific operations can be intrinsically easier on quantum computers. Let's go back to the context of neutrino flavor evolution: this system is characterized by a Hamiltonian involving many contributions (see Eq. (3.19)), among which the most commonly considered ones are vacuum mixing, matter effects, and neutrino self-interactions, and it gives rise to complex quantum phenomena like neutrino flavor oscillations. On quantum hardware, the time evolution of this system can be performed by decomposing it in an efficient sequence of quantum gates acting on the mapped qubit states (see Section 1.2.2), relying on multi-qubit operations to directly generate entanglement and complex correlation in the system. Another relevant example concerns once again thermal system evolution: quantum algorithms can rely on methods like the *Quantum Imaginary-Time Evolution* (QITE) (see Section 3.3), which allows for an efficient preparation of the needed mixed states, while classical approaches face an exponential overhead when dealing with the sampling of thermal distributions or the computation of partition functions.
It is important to notice that, to the best of my knowledge, today there is yet no evidence of a classical algorithm that can tackle the simulation of generic many-body quantum physical systems characterized by states showing high levels of entanglement. This is for sure one of the principal fields where quantum supremacy is expected to emerge and where quantum computers have the highest probability to prove their exponentially better performances with respect to classical machines.

1.1.3 Challenges in QC: noise and decoherence

We just saw some of the main motivations that support the theoretical quantum computing exponential advantage over classical computation, but we didn't focus much on their limitations yet, which is indeed a particularly relevant and delicate topic. The principal sources of potential limitations to quantum computing capabilities are *noise*, *decoherence*, *limited qubit connectivity*, and *imperfect gate*

realization: all of these phenomena contribute to limit the scalability gain of quantum algorithms and reduce the results fidelity.
The main source of problems to quantum computation is the noise due to coupling with the environment. Due to their nature, a complete isolation of quantum systems from external interactions is practically impossible to achieve, so that these influences will modify in a more or less random and for sure uncontrollable way the operations performed on the qubits, leading to an unavoidable decoherence of their state. This means that the different components of the quantum superpositions acquire spurious phases, disrupting the same intended superposition [21]. Generating correct entanglement and superposition is crucial to a good quantum algorithm, so that a limitation to their realization puts strong constraints to the maximum level of accuracy that can be achieved.
Different architectures may show different levels of coupling with the environment, but usually the role of noise is always non-negligible, and typically the most noise-resistant technologies have different drawbacks, like for example much longer times needed for the gate realization. There is no universally acknowledged best architecture for qubits; rather, choosing the best one is usually part of the problem, and algorithms tipically have to be specifically optimized for each choice: some possible alternatives include superconducting circuits, trapped ions, photons and neutral atoms.
Another sore topic affecting ideal computation is the imperfection of the gate realization: rather than performing the ideal desired quantum operation, real hardware will be affected by fluctuations in electronics and loss of control over the generated pulses, which spoils the correct implementation of a quantum circuit. The longer the circuit, the more these imperfections will add up and become relevant, with the consequence that algorithms relying on many iterations or longer circuits will be more affected by this kind of error source.
Especially relevant to this discussion is also the topic of qubit connectivity in quantum hardware: oftentimes, the machine infrastructure only allows for gate operations between neighboring qubits on specific graphs, rather than allowing complete freedom in the choice of qubits onto which performing a given operation. The full connectivity is usually effectively restored by moving around qubit states through the application of SWAP instruction sequences [22], which theoretically allow for arbitrary long-range entanglement generation at the price of substantially increasing the number of operations required, i.e. the total circuit depth.
This is fundamental to the practical feasibility of many quantum simulations, since lots of the most interesting physical systems are characterized by extremely non-local interactions (collective neutrino flavor oscillations are a typical example). Hardware architectures with richer connectivity graphs, or techniques to compile circuits that minimize qubit movement, are thus vital for scalable and accurate

simulations.
An ideal cure to many of these problems would consist of *quantum error correction* [23–25], which expands from its classical formulation and promises to equip quantum computing with extreme resilience to errors. Nowadays though, it is not a reliable approach, since it usually requires far more quantum resources than the ones typically available.
The alternative, already vastly employed and consisting of a great number of different techniques, is what's known under the name of *noise mitigation*: this is an always increasing collection of procedures which can be safely realized on NISQ (Noisy Intermediate-Scale Quantum) devices available today that employ a limited amount of quantum resources to obtain relevant advantages in the limitation of the effect of noise on quantum algorithms. As an example, *Dynamical decoupling* [26, 27] is one of the most common methodologies, consisting of the application of carefully timed control pulses on idle qubits, to suppress certain types of decoherence.
Overall, despite the current generation of available quantum hardware not being yet sufficient for large-scale and high-fidelity quantum simulations, noise mitigation, improvements in connectivity, and control fidelity are rapidly improving the range of problems that can be tackled effectively. Each of these efforts contributes to extending the reach of quantum simulation, bringing within grasp a class of problems that remain fundamentally intractable for classical computation.

1.2 QC building blocks

There exist several paradigms for quantum computation, including *analog* quantum computing [28], where the system evolves under a Hamiltonian controlled via continuous parameters, and *digital*, or *gate-based*, or also *circuit-based* quantum computing [29], which is based on the sequential application of discrete unitary operations. Other less common models are also possible, like the measurement-based [30] or adiabatic quantum computing [31]. In this thesis though, the focus will exclusively be on the digital, gate-based paradigm, which has been proven to be universal [3] and constitutes the most commonly considered framework for developing general-purpose quantum algorithms.
As already introduced, quantum computing is based on the manipulation of information that is stored at the quantum level, mapped to the state of controllable systems. These states are modified by the action of quantum operations, which rely on the laws of quantum mechanics to exploit non-classical resources like entanglement and superposition of states, aiming to reach better computational performances and to overcome classical computer limits.
The typical steps in a quantum computation are essentially three:

- the **initialization** of the system in a well-determined quantum state;

- the **transformation** of the initial state through the application of a certain set of quantum operators called gates, analogous to the logical gates in classical computing;
- the **measurement** of some observables on the final state, allowing for the recovery of the processed information stored in the quantum system.

In this section, I'll introduce the main aspects of these three fundamental procedures.

1.2.1 Qubits and quantum states

Quantum computation fundamental characteristic is that of manipulating the states of a quantum system, which encode the relevant information. In the following, I introduce the mathematical framework suitable for the description of the state of a set of qubits, I discuss the state representation both in state-vector and density-matrix formalisms, and distinguish pure from mixed states. I'll also comment on the Bloch sphere picture.

Single-qubit pure states

First of all, I recall that the typical fundamental building block of a universal quantum computer is the *qubit*, a two-level quantum system whose state can be identified as a *ray* in a two-dimensional Hilbert space $\mathcal{H} \doteq \mathbb{C}^2$, where a ray is a unit vector having an arbitrary global phase. This means that any two unit elements of $\mathbb{C}^2$ differing only by a global phase factor describe the same qubit state.
The typical notation used to represent a state is called *Dirac notation* [32] and employs the *ket* symbol $|\cdot\rangle$, so that a general state of a qubit can generically be written down as $|\psi\rangle$.
To further characterize qubit states, they are usually represented in the so-called *computational basis*, consisting of the eigenstates of the Pauli $\hat{Z}$ operator, better introduced in the next Section 1.2.2

$$\left\{ |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}. \tag{1.2}$$

The states of this basis are called $|0\rangle$ and $|1\rangle$ and are defined by the following relations

$$\hat{Z}\,|0\rangle = |0\rangle \qquad \hat{Z}\,|1\rangle = -\,|1\rangle \tag{1.3}$$

On this basis, the general qubit state can be expressed as the following

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle = \begin{pmatrix} a \\ b \end{pmatrix} \qquad |\ a, b \in \mathbb{C},\ |a|^2 + |b|^2 = 1, \tag{1.4}$$

where a and b are probability complex amplitudes, their modulus squared representing the probability of finding the qubit in states $|0\rangle$ and $|1\rangle$ after performing a measurement in the computational basis (for the measurement procedure see Section 1.2.3). Written in this form, the state of a qubit is evidently different with respect to that of a classical bit: the first in fact can in principle be any possible (normalized) complex linear superposition of the two computational basis states, while the latter is only either one or the other, at any time. This superposition is an intrinsic quantum property, which is exploited to gain advantages with respect to classical computations.
Thanks to the global phase invariance of $|\psi\rangle$, parameter a in Eq. (1.4) can always be taken to be real and positive: in this case usually the following parametrization of $|\psi\rangle$, involving two real angles θ and ϕ, is used to express the state

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right) \qquad | \ \theta \in [0,\pi),\ \phi \in [0,2\pi), \tag{1.5}$$

where i is the imaginary unit.
Eq. 1.5 suggests that a qubit state can also be represented as the oriented radius of a unit tridimensional sphere: such a sphere is called *Bloch sphere*, where θ and ϕ represent the *polar* and *azimuthal* angles on it, respectively.
This parallelism between unit vectors in $\mathbb{R}^3$ and qubit states is possible because there is a one-to-one mapping between the two, which is evident by looking at the general form for a unit 3D real vector $\vec{r}$ in cartesian coordinates

$$\begin{aligned} \vec{r} &= \{r_x,\, r_y,\, r_z\} \\ &= \{\sin(\theta)\cos(\phi),\, \sin(\theta)\sin(\phi),\, \cos(\theta)\} \qquad | \ \theta \in [0,\pi),\ \phi \in [0,2\pi). \end{aligned} \tag{1.6}$$

This relation between $\mathbb{R}^3$ unit vectors and single-qubit states $|\psi\rangle$ is even more striking when looking at the *density matrix* $\hat{\rho}_\psi$ associated to the state $|\psi\rangle$, which can be defined as the outer product of $|\psi\rangle$ with itself: this quantity is then an operator, which can be written in the following way

$$\hat{\rho}_\psi \doteq |\psi\rangle\langle\psi| = \frac{1}{2}\left(\hat{\mathbb{I}} + r_x\hat{X} + r_y\hat{Y} + r_z\hat{Z}\right), \tag{1.7}$$

where I introduced the identity operator $\hat{\mathbb{I}}$ and the Pauli $\hat{X}$ and $\hat{Y}$ operators, which will be better described in the next Section 1.2.2 and that on the computational basis have the following representation

$$\hat{\mathbb{I}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ \hat{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\ \hat{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},\ \hat{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{1.8}$$

In Eq. (1.7) I also used the symbol $\langle\cdot|$, called *bra* in Dirac notation: this symbol associates any element $|\psi\rangle$ of the qubit Hilbert space $\mathcal{H}$ to a functional, belonging

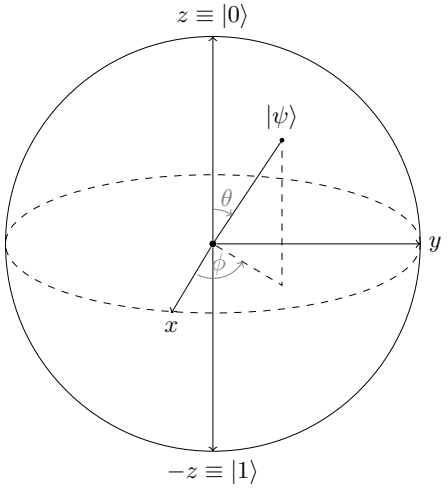


Figure 1.3: Bloch vector visualization of a general single-qubit state $|\psi\rangle$

to the dual space $\mathcal{H}^*$, such that when it acts on a ket, it gives the internal product with that element. To be more clear, the following relation defines the bra of any element belonging to the Hilbert space $\mathcal{H}$

$$\langle\phi| \ : \ |\psi\rangle \mapsto \langle\phi|\psi\rangle \qquad \forall \ |\phi\rangle, |\psi\rangle \in \mathcal{H}, \tag{1.9}$$

with $\langle\phi|\psi\rangle$ being the internal product between the two elements $|\psi\rangle$ and $|\phi\rangle$.
From Eqs. (1.5), (1.6) and (1.8) one can easily check that r_x, r_y and r_z in Eq. (1.7) correspond to the expectation values of the respective Pauli operators on the quantum state $|\psi\rangle$ (see Section 1.2.3). Harvesting from this parallelism, one can deduce that any general unitary operator transforming the state of a single qubit can always be seen as a rotation on the Bloch sphere surface, which can always be decomposed as a combination of rotations around the three orthogonal directions, each generated by the appropriate Pauli operator (see Section 1.2.2).

Single-qubit mixed states

The states considered until now are known as *pure states*, but they are not the only kind of states possible for a quantum system: some processes characterized by an intrinsic random nature, may in fact evolve a system to a configuration corresponding to a statistical mixture of pure states, rather then to a single, fixed, well-determined, pure state. This situation can be interpreted as if the considered process projects the system in a state that has the probability p_j of being $|\psi_j\rangle$: this kind of state is called *mixed state*, since it is a statistical mixture of pure states,

and can be described by a density matrix $\hat{\rho}$ written in the following form

$$\hat{\rho} \doteq \sum_j p_j \hat{\rho}_j = \sum_j p_j \left|\psi_j\right\rangle \left\langle\psi_j\right| \qquad \mid\ p_j \in \mathbb{R},\ 1 > p_j \geq 0\ \forall j,\ \sum_j p_j = 1. \tag{1.10}$$

The density matrices of pure and mixed states are not identical: both have unit trace, are positive semidefinite and Hermitian, but while a pure density matrix $\hat{\rho}_j$ can always be written as the outer product of a single pure state $\left|\psi_j\right\rangle$ with itself and is such that $\hat{\rho}_j^2 = \hat{\rho}$, the same is not true for a mixed density matrix $\hat{\rho}$, which instead satisfies $0 \leq \mathrm{Tr}[\hat{\rho}^2] < 1$.
In general, the computational basis in Eq. (1.2) can always be used to decompose any single-qubit density matrix, which takes the following form

$$\hat{\rho} = \sum_{i,j=0}^{1} \rho_{ij} \left|i\right\rangle \left\langle j\right| \qquad \mid\ \rho_{i,j} \in \mathbb{C},\ \hat{\rho} = \hat{\rho}^\dagger,\ \hat{\rho} \geq 0,\ \mathrm{Tr}[\hat{\rho}] = 1 \tag{1.11}$$

The Bloch sphere representation is also still valid for mixed states, with the slight difference that they correspond to vectors starting from the sphere center and having norm smaller than 1

$$\hat{\rho} = \frac{1}{2}\left(\hat{\mathbb{I}} + \vec{r}\cdot\vec{\hat{\sigma}}\right) \qquad \mid\ \|\vec{r}\| \leq 1, \tag{1.12}$$

where I introduced the Pauli vector $\vec{\hat{\sigma}}$

$$\vec{\hat{\sigma}} \doteq \left(\hat{X}, \hat{Y}, \hat{Z}\right). \tag{1.13}$$

Many-qubit states

The generalization to the description of a system of many qubits, let's say N, is quite standard. First of all, let's introduce the notion of *separable states* $\left|\psi_S\right\rangle_N$: these are pure states of the full system such that each n-th single particle is in a well-defined pure state $\left|\psi_n\right\rangle$. The full separable state can then be built as the Kronecker product of the single-particle ones and belongs to the Hilbert space obtained by taking the tensor product of all the single-particle Hilbert spaces

$$\left(\mathbb{C}^2\right)^{\otimes N} \ni \left|\psi_S\right\rangle_N \doteq \bigotimes_{n=1}^{N} \left|\psi_n\right\rangle. \tag{1.14}$$

The elements of the computational basis for a system of N qubits are separable states such that each qubit is in a computational basis state, $\left|0\right\rangle$ or $\left|1\right\rangle$: the sequence of single-qubit states constituting the system can then be represented by a bitstring

of zeros and ones, which can also be interpreted as a binary number and thus converted to the corresponding decimal integer, which is comprised between 0 and $2^N - 1$. Here I present a simple example

$$|5\rangle_4 = |0101\rangle = |0\rangle \otimes |1\rangle \otimes |0\rangle \otimes |1\rangle \,. \tag{1.15}$$

The full set of computational basis states for a system of N qubits can then be written in the following way

$$\left\{ |n\rangle_N \;\middle|\; n \in \{0, \ldots, 2^N - 1\} \right\}. \tag{1.16}$$

A general pure state of a system of N qubits is then a ray in the space $(\mathbb{C}^2)^{\otimes N}$ built by the tensor product of the Hilbert spaces $\mathbb{C}^2$ of the single-qubit states and as such can always be represented as a complex linear combination of states from the computational basis state

$$|\psi\rangle_N = \sum_{n=0}^{2^N-1} c_n \, |n\rangle_N \qquad \mid c_n \in \mathbb{C} \;\forall n, \; \sum_{n=0}^{2^N-1} |c_n|^2 = 1 \tag{1.17}$$

If the state $|\psi\rangle_N$ cannot be written in the form of Eq. (1.14), than it is called *entangled state*: this kind of states are characterized by some intrinsically non-classical correlations between qubits, which are an important resource exploited in quantum computation to try to overcome classical limitations.
The generalization of mixed states to the case of N qubits is still obtained through to the introduction of density matrices $\hat{\rho}_N$, which can still be decomposed in the many-qubit computational basis of Eq. (1.16), but also as a linear combination of tensor products of all the possible Pauli strings:

$$\hat{\rho}_N = \sum_{i,j=0}^{2^N-1} \rho_{ij} \, |i\rangle \langle j| \qquad \mid \rho_{ij} \in \mathbb{C}, \; \hat{\rho}_N = \hat{\rho}_N^\dagger, \; \hat{\rho}_N \geq 0, \; \mathrm{Tr}[\hat{\rho}_N] = 1, \tag{1.18}$$

$$= \frac{1}{2^N} \sum_{i=0}^{2^N-1} c_i \hat{P}_i^{(N)} \qquad \mid \hat{P}_i^{(N)} \doteq \bigotimes_{j=1}^{N} \hat{P}_{i_j}, \; \hat{P}_{i_j} \in \left\{ \hat{\mathbb{I}}, \hat{X}, \hat{Y}, \hat{Z} \right\}, \tag{1.19}$$

where $c_i = \mathrm{Tr}\left[\hat{\rho}_N \hat{P}_i^{(N)} \right]$.

1.2.2 Operations

Having established the mathematical framework used for the description of quantum states, let's now turn to the study of *quantum operations*, also referred to as *quantum gates*. The closed-system (unitary) model of quantum computation

describes these objects as transformations changing a quantum state of a system to a new, usually different, quantum state of the same system. Consequently, they correspond to unitary transformations, which in finite dimensions can always be represented by matrices acting on the Hilbert space of one or more qubits. These operations must preserve the inner product structure of the state space and are therefore constrained to be linear and norm-preserving.
Quantum gates can be distinguished depending on the number of qubits onto which they act: later we will see how usually gates acting on single and two qubits are considered to be enough for allowing the implementation of a generic operation, so now I'll move to present the most important gates of these types.

Single-qubit gates

Single-qubit gates are represented by elements of the unitary group $\mathcal{U}(2)$ and therefore act on a two-dimensional Hilbert space $\mathcal{H} = \mathbb{C}^2$, $\mathcal{U}(2)$ grouping all possible 2x2 unitary matrices.
In the following I'll refer to gates both as operators and matrices, having in mind the fact that I'm considering quantum operations acting on a well-defined number of qubits, which means that their representation in matrix form is also well defined and only depends on the choice of the basis.
Whenever the specific information about the global phase of the state after the action of the operation is not relevant, meaningful single-qubit gates reduce to the elements of the special unitary group $\mathcal{SU}(2)$, which is the group of all 2x2 unitary matrices having determinant equal to 1.
Thanks to this parallelism with group theory, it is possible to deduce that all single-qubit operations can be fully characterized by analyzing a small set of fundamental gates, which can be taken to consist of the already rapidly introduced Pauli operators (see Section 1.2.1 and Eq. (1.8)), all of which are Hermitian and therefore associated with physical observables.

The identity and Pauli gates. *Pauli gates* are three operators usually called $\hat{X} \equiv \hat{\sigma}_x$, $\hat{Y} \equiv \hat{\sigma}_y$ and $\hat{Z} \equiv \hat{\sigma}_z$, which certainly are among the most important and common single-qubit gates. They possess lots of useful properties, among which I list the following ones: they are

- Traceless: $\mathrm{Tr}\left[\hat{\sigma}_j\right] = 0 \qquad \forall j \in \{x, y, z\}$;
- Hermitian: $\hat{\sigma}_j^\dagger = \hat{\sigma}_j \qquad \forall j \in \{x, y, z\}$;
- Involutory: $\hat{\sigma}_j^2 = \hat{\mathbb{I}} \qquad \forall j \in \{x, y, z\}$;
- Unitary: $\hat{\sigma}_j^\dagger \hat{\sigma}_j = \hat{\mathbb{I}} \qquad \forall j \in \{x, y, z\}$.

In the definitions above I made use of yet another extremely important operator, the *identity operator* $\hat{\mathbb{I}}$: this operator is defined regardless of the dimension of the Hilbert space $\mathcal{H}$ onto which it acts and corresponds to the trivial transformation, the one which does not change the state onto which it is applied

$$\hat{\mathbb{I}}\,|\psi\rangle \doteq |\psi\rangle \qquad \forall\,|\psi\rangle \in \mathcal{H}. \tag{1.20}$$

Sometimes, when explicitly identifying the identity operator as acting on a Hilbert space of a given dimension d, the following notation is used: $\hat{\mathbb{I}} \to \hat{\mathbb{I}}_d$.
The identity and Pauli operators together constitute a basis for the decomposition of any 2x2 complex matrix $\hat{M} \in \mathbb{C}^{\mathbf{2x2}}$, even those which are not Hermitian or unitary

$$\hat{M} = m_i\hat{\mathbb{I}} + m_x\hat{X} + m_y\hat{Y} + m_z\hat{Z} \qquad \forall\hat{M} \in \mathbb{C}^{\mathbf{2x2}}, \tag{1.21}$$

where the coefficients of the expansion are given by the following formula

$$\mathbb{C} \ni m_j = \frac{1}{2}\,\mathrm{Tr}\left[\hat{\sigma}_j\hat{M}\right] \qquad \forall j \in \{i, x, y, z\}, \tag{1.22}$$

and where I defined $\hat{\sigma}_i \doteq \hat{\mathbb{I}}$.
This can be further generalized to represent any complex 2^Nx2^N matrix $\hat{M}_N$

$$\hat{M}_N = \sum_{j=1}^{4^N} m_j\hat{P}_j^{(N)} \quad \forall\ \hat{M}_N \in \mathbb{C}^{2^N\mathbf{x}2^N} \quad |\ \mathbb{C} \ni m_j = \frac{1}{2^N}\,\mathrm{Tr}\Big[\hat{P}_j^{(N)}\hat{M}_j\Big], \tag{1.23}$$

where in this case the basis is constituted by all the possible different Kronecker products of N operators taken from the set of Pauli operators plus the identity

$$\hat{P}_j^{(N)} \in \{\bigotimes_{k=1}^{N}\hat{P}_k \mid \hat{P}_k \in \{\hat{I}, \hat{X}, \hat{Y}, \hat{Z}\}\}. \tag{1.24}$$

The matrix representation of the Identity and Pauli operators on the computational basis was already introduced in Eq. (1.8).
Another extremely significant result consists in the fact that the three Pauli operators also constitute a basis for the generators of the $\mathcal{SU}(2)$ group (formally up to a constant scalar rescaling factor), which is directly connected to the rotations on a 3D-sphere: a generic rotation around the axis identified by the tridimensional unit vector $\vec{n}$ can in fact always be represented in the following way

$$\hat{R}_{\vec{n}}(\theta) = e^{i\frac{\theta}{2}\vec{n}\cdot\vec{\sigma}} = \cos\left(\frac{\theta}{2}\right)\hat{\mathbb{I}} + i\sin\left(\frac{\theta}{2}\right)\vec{n}\cdot\vec{\hat{\sigma}}, \tag{1.25}$$

where $\vec{\hat{\sigma}}$ is the Pauli vector introduced in Eq. (1.13).
This means that each Pauli operator generates a rotation of the qubit state around one out of three orthogonal axes

$$\hat{R}_x(\theta) \doteq e^{i\frac{\theta}{2}\hat{X}}, \qquad \hat{R}_y(\theta) \doteq e^{i\frac{\theta}{2}\hat{Y}}, \qquad \hat{R}_z(\theta) \doteq e^{i\frac{\theta}{2}\hat{Z}}. \tag{1.26}$$

Particularly relevant is also the realization that the Pauli $\hat{X}$ gate corresponds to the generalization of the classical NOT gate (see Fig. 1.1): recalling its representation given in Eq. (1.8) in fact, it is straightforward to verify the following identities

$$\hat{X}\,|0\rangle = |1\rangle\,, \qquad \hat{X}\,|1\rangle = |0\rangle\,. \tag{1.27}$$

The Hadamard gate. The Hadamard gate is often called $\hat{H}$, possibly generating some confusion with the symbol usually used for the Hamiltonian operator, and on the computational basis it is represented by the following matrix

$$\hat{H} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{1.28}$$

This gate plays a central role in the generation of quantum interference, because it transforms single-qubit computational basis states into an equal superposition of them:

$$\hat{H}\,|0\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) \doteq |+\rangle\,, \qquad \hat{H}\,|1\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right) \doteq |-\rangle\,. \tag{1.29}$$

$|+\rangle$ and $|-\rangle$ states together constitute another possible basis for the single-qubit Hilbert space, usually called X-basis since these states are the eigenvalues of the Pauli $\hat{X}$ operator

$$\hat{X}\,|+\rangle = |+\rangle\,, \qquad \hat{X}\,|-\rangle = -\,|-\rangle\,. \tag{1.30}$$

The phase gate. The phase gate is usually called $\hat{S}$ and its effect is that of introducing a phase on the $|1\rangle$ component of a single-qubit state. It is therefore represented on the computational basis by the following matrix.

$$\hat{S} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \tag{1.31}$$

corresponding to a $\pi/2$ rotation about the z-axis on the Bloch sphere and therefore satisfying $\hat{S}^2 = \hat{Z}$.

Two-qubit gates

Two-qubit gates are operators acting on the Hilbert space of two qubits, therefore corresponding to the elements of the unitary $\mathcal{U}(4)$ group, the group of unitary 4x4 matrices. These gates are fundamental in quantum algorithms, as they are responsible for the generation of entanglement in the qubit system, which is an extremely valuable resource to approach quantum exponential advantage.
Let's see a few of the typical gates belonging to this category

The CNOT gate CNOT gate, the Controlled-NOT gate, is also sometimes called $\hat{CX}$, as it corresponds to the application of the $\hat{X}$ gate on a qubit, conditioned to another qubit being in the state $|1\rangle$. Usually, it is assumed that the controlling qubit is the first one, meaning the one that would appear more on the left in the Kronecker product of separable states (refer to Eq. (1.14)): in this case, the CNOT gate is represented on the computational basis by the following matrix

$$\hat{CX} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{1.32}$$

As I will better introduce later, a sequence of quantum gates is usually represented graphically via some circuit, similarly to what happens in classical computing (see Fig. 1.2): each horizontal line represents a qubit, while gates are represented by precise symbols appearing along these lines. The symbol corresponding to the CNOT gate is the one reported in Fig. 1.4, where the top qubit is the first, the controlling one, while the bottom qubit is the one conditionally subject to the $\hat{X}$ operation.

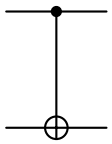


Figure 1.4: Symbol used for the representation of the CNOT gate on quantum circuits: the top line corresponds to the controlling circuit, the bottom one to the qubit conditionally subject to the $\hat{X}$ gate.

The SWAP gate The swap gate is another very relevant two-qubit gate, as it allows for exchanging the state of two qubits: this is of key importance for quantum computers with limited connectivity, as it can be exploited to shuffle the

stored information around, moving it to a spot where it can be processed.
On the computational basis, the SWAP gate is represented by the following matrix

$$\text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{1.33}$$

As a side note, I stress here the fact that the SWAP gate is evidently a special case of a Givens rotation (see Section 3.3.2 and compare with Eq. (3.74)).
The quantum circuit symbol associated to the SWAP gate is illustrated in Fig. 1.5

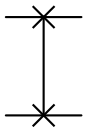


Figure 1.5: Symbol used for the representation of the SWAP gate on quantum circuits.

Universality, gate decomposition and quantum circuits

As I briefly pointed out previously, one-qubit and two-qubit gates are generally enough to implement an arbitrary quantum operation $\hat{U} \in \mathcal{U}(2^N)$, acting on any number N of qubits.
This important result is implied by the Solovay-Kitaev theorem [33, 34], which states that any unitary operation can always be accurately approximated by a limited and 'short' sequence of operations taken from a certain set, which can be chosen to consist of only one-qubit and two-qubit operations. More precisely, the theorem states that for any positive real number $\varepsilon > 0$ and unitary $\hat{U} \in \mathcal{U}(d)$, d being a positive integer, there will always exist a real positive constant c such that there exists at least one product $\hat{P}$ of $\mathcal{O}(\log^c(\frac{1}{\varepsilon}))$ operators taken from a given set that will be at least ε-close to $\hat{U}$ in operatorial norm, i.e. such that the following inequality holds

$$\left\| \hat{P} - \hat{U} \right\|_{\text{op}} \leq \varepsilon, \tag{1.34}$$

where the operatorial norm of a quantum operator $\hat{O}$ is defined in the following way through the usual Hilbert space 2-norm

$$\left\| \hat{O} \right\|_{\text{op}} \doteq \sup_{|\psi\rangle \ | \ \||\psi\rangle\| \neq 0} \left[\frac{\left\| \hat{O} |\psi\rangle \right\|}{\||\psi\rangle\|} \right]. \tag{1.35}$$

There are several different sets which constitute a *universal set* in this sense, as an example the one made by arbitrary rotations about the z-axis and y-axis plus

the CNOT, $\{\hat{R}_z(\theta), \hat{R}_y(\phi), \hat{CX}\}$, but many more exist. Operations in quantum hardware are realized by exploiting this principle: there usually is just a small set of few-qubits *native* gates that can be physically implemented on the qubits. All other more complex operations are decomposed on this set of native gates and implemented as a sequence of these operations. Finding the optimal decomposition is quite a big problem yet, and often times a lot of effort in the formulation of a new quantum algorithm is put into this task.
A general sequence of gates, native or not, is usually represented graphically with a scheme inspired to the classical logic circuits, which is therefore called *quantum circuit*: this is constituted by many horizontal lines, each one representing a qubit, while the gates are inserted as specific symbols, drawn on the lines corresponding to the qubits onto which they act, starting from the left with the gates acting first. Generic squares represent generic gates, usually specified by writing the gate name inside the square, while more common gates are associated to specific symbols (see Figs. 1.4, 1.5). At the starting point, on the far left of the circuit, a system of N qubits is assumed to be in the $|0\rangle_N$ computational basis state (see Eq. (1.16)), with the first qubit at the top corresponding to the left-most in the Kronecker product of separable states (refer to Eq. (1.14)). An example of a quantum circuit is represented in Fig. 1.6.

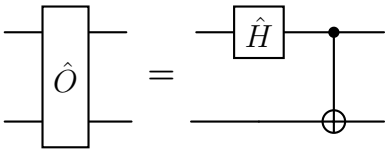


Figure 1.6: Example of a quantum circuit, corresponding to one of the easiest alternatives to produce a maximally entangled state on two qubits: to the left I represented the circuit as a generic operator $\hat{O}$ applied on two qubits, while to the right I decomposed it as an Hadamard operator applied on the first qubit followed by a CNOT gate controlled on the first qubit.

Exponential operator and time evolution

The state of any physical quantum system naturally changes with time, in a way that directly depends on the specific interactions among its constituents. If $\hat{H}$ is the Hamiltonian describing these interactions and $|\psi_0\rangle$ is the system state at time $t = 0$, then according to quantum mechanics the system state evolution will be governed by the following equation

$$|\psi(t)\rangle = e^{-it\hat{H}} |\psi_0\rangle , \tag{1.36}$$

where $|\psi(t)\rangle$ is the system state at time $t \in \mathbb{R}$.
Since the Hamiltonian is an Hermitian operator, $\hat{H} = \hat{H}^\dagger$, it follows that the ex-

ponential operator $\hat{U}(t) \doteq e^{-it\hat{H}}$ must be unitary: it is called *Real-Time Evolution* (RTE) operator and it lies at the base of the way that native gates are typically implemented in quantum hardware. Quantum computers are in fact realized by allowing a certain degree of control over the system Hamiltonian, so that certain terms can be modified or switched on or off: by carefully selecting and controlling these terms as well as the duration of the evolution, it is possible to engineer the real-time evolution operator so that it effectively implements the desired native gate $\hat{G}$ on the target qubits

$$\hat{G} = e^{-it_G \hat{H}_G}. \tag{1.37}$$

This is also the practical reason why only unitary operations are physically allowed on a real quantum computer.
In principle though, one could also be interested in realizing some non-unitary operations: as some examples, these operations are necessary to build thermal states, which I will show in Chapter 3, but they could also be useful for extracting the ground state of a system.
In general, non-unitary operations are described by an *Imaginary-Time Evolution* (ITE) operator, obtained by substituting the real time t in Eq. (1.36) with a completely imaginary time: this is called *Wick rotation* [35], as it corresponds to a $\pi/2$ rotation on the complex plane of the time axis

$$\mathbb{R} \ni t \to -i\tau \mid \tau \in \mathbb{R} \qquad \Longrightarrow \qquad \text{RTE } e^{-it\hat{H}} \to e^{-\tau\hat{H}} \text{ ITE}. \tag{1.38}$$

In order to implement non-unitary operations on quantum computers, one has to rely on different tricks: a possibility is that of performing intrinsic non-unitary operations like measurements [36], which, due to their projective nature, do not conserve the information stored in the state. An alternative consists in increasing the Hilbert space of the problem by relying on more qubits than the ones needed for the non-unitary gate action: these additional qubits are called *ancillas* and their role is to allow the application of a unitary gate on the full system, which reduces to the desired non-unitary gate on the original subspace [37]. In Chapter 3 I will also introduce the QITE algorithm, which is a way of generating approximate imaginary-time evolved states by only acting with unitary operators and that does not rely on the introduction of ancillas: the idea is to construct the unitary generating the normalized time evolved state $e^{-\tau\hat{H}} |\psi\rangle / \mathcal{N}$, with $\mathcal{N} \doteq \left\| e^{-\tau\hat{H}} |\psi\rangle \right\|$, and at the same time find an estimate of $\mathcal{N}$.
As a last note, I state that RTE and ITE can be put together by defining a complex time $\mathbb{C} \ni \mathfrak{t} \doteq t - i\tau$: the exponential operator generated by a given Hamiltonian $\hat{H}$ will be then generically called *Time Evolution* (TE) operator

$$e^{-i\mathfrak{t}\hat{H}} \mid \mathfrak{t} \in \mathbb{C} \text{ TE} \tag{1.39}$$

1.2.3 Measurements

In this thesis, all quantum measurements are performed in the computational basis (see Eq. (1.16) and [32]), which means that after the measurement, the system is found in one of the computational pure states, with probability depending on the decomposition in such basis of the system state before the measurement (see Eq. (1.18)). More explicitly, considering a system of N particles in the state described by the density matrix $\hat{\rho}_N$, the probability for the measurement to produce state $|j\rangle$ is given by the following expression

$$P(j) = \mathrm{Tr}\left[|j\rangle\langle j|\,\hat{\rho}_N\right] = \rho_{jj} \;\forall j \in \{0,\ldots,2^{N-1}\}, \tag{1.40}$$

where I recall that density matrices have unit trace, $\sum_{j=0}^{2^N-1}\rho_{jj} = 1$.
The nature of quantum measurement is then inherently probabilistic: single measurements yield stochastic outcomes, which individually are not sufficient to fully characterize the diagonal components on the computational basis of the quantum state. Let's consider an operator $\hat{O}$ that is diagonal in the computational basis, and as such can be written in the following form

$$\hat{O} = \sum_{i=0}^{2^N-1} O_i\,|i\rangle\langle i|\,, \tag{1.41}$$

with O_i being the eigenvalue of $\hat{O}$ corresponding to eigenvector $|i\rangle$, one of the computational basis elements.
Its expectation value $\langle\hat{O}\rangle_{\rho_N}$ on the density matrix $\hat{\rho}_N$ is the following

$$\begin{aligned}\langle\hat{O}\rangle_{\rho_N} = \mathrm{Tr}\left[\hat{O}\hat{\rho}_N\right] = \sum_{i=0}^{2^N-1}\langle i|\,\hat{O}\hat{\rho}_N\,|i\rangle = \sum_{i,j,k,l=1}^{2^N-1} O_j\rho_{kl}\langle i|j\rangle\langle j|k\rangle\langle l|i\rangle = \\ = \sum_{i,j,k,l=1}^{2^N-1} O_j\rho_{kl}\delta_{ij}\delta_{jk}\delta_{li} = \sum_{i=1}^{2^N-1} O_i\rho_{ii},\end{aligned} \tag{1.42}$$

where I used the definitions given in Eqs. (1.41) and (1.18) and exploited the fact that computational basis states are orthonormal, so that $\langle i|j\rangle = \delta_{ij}$ with δ_{ij} the Kronecker delta, having value 1 when its arguments are identical and 0 otherwise.
In order to get a faithful estimate of $\langle\hat{O}\rangle_{\rho_N}$, the measurement procedure must be repeated many times under identical and independent state preparations, so that enough statistics can be collected.
Each repetition of the same measurement is called a *shot*, and, as already said, projects the system in one computational basis state $|j\rangle$, which corresponds to the $\hat{O}$ eigenvalue O_j. Performing a measurement can then be interpreted as taking a

sample from $\langle \hat{O} \rangle_{\rho_N}$, which can be estimated as the sample average $\bar{O}_{\rho_N}$ of all the measurements. If N_S is the total number of shots and N_j is the number of shots producing state $|j\rangle$, then the expectation value is estimated as the following

$$\langle \hat{O} \rangle_{\rho_N} \cong \bar{O}_{\rho_N} \doteq \frac{1}{N_S} \sum_{j=0}^{2^N-1} N_j O_j. \tag{1.43}$$

As $N_S \to \infty$, this sample average converges to the true expectation value due to the law of large numbers. For a finite number of samples instead, the error on such quantity is estimated as the standard error $\sigma\left(\bar{O}_{\rho_N}\right)$

$$\sigma\left(\bar{O}_{\rho_N}\right) \doteq \frac{1}{N_S} \sqrt{\sum_{j=0}^{2^N-1} N_j \left(O_j - \bar{O}_{\rho_N}\right)^2}. \tag{1.44}$$

In experimental implementations, typical values of N range from a few thousands to tens of thousands to achieve sufficient statistical precision.

Let's now consider the estimation of expectation values of operators that are not diagonal in the computational basis, starting from the two Pauli operators $\hat{X}$ and $\hat{Y}$. In this case one exploits the following identities

$$\hat{X} = \hat{H}\hat{Z}\hat{H} \qquad\qquad \hat{Y} = \hat{S}\hat{H}\hat{Z}\hat{H}\hat{S}^\dagger, \tag{1.45}$$

where I exploited the Hadamard gate $\hat{H}$ and the phase gate $\hat{S}$ introduced in Eqs. (1.28) and (1.31).
The identities in Eq. (1.45) imply in fact that these two Pauli operators expectation values can be obtained as the $\hat{Z}$ expectation value on a rotated state

$$\langle \psi | \hat{X} | \psi \rangle = \langle \psi | \hat{H}\hat{Z}\hat{H} | \psi \rangle = \langle \psi' | \hat{Z} | \psi' \rangle, \tag{1.46}$$

$$\langle \psi | \hat{Y} | \psi \rangle = \langle \psi | \hat{S}\hat{H}\hat{Z}\hat{H}\hat{S}^\dagger | \psi \rangle = \langle \psi'' | \hat{Z} | \psi'' \rangle, \tag{1.47}$$

where $|\psi\rangle$ is a generic one-qubit state and I defined $|\psi'\rangle \doteq \hat{H}|\psi\rangle$ and $|\psi''\rangle \doteq \hat{H}\hat{S}^\dagger|\psi\rangle$.
The expectation value estimations for $\hat{X}$ and $\hat{Y}$ are then obtained as previously described for the case of observables that are diagonal in the computational basis, with the only difference that the correct transformation must be added to the circuit before taking each measurement.
The generalization to the measurement of strings of Pauli operators on many-qubit states is trivial: everything works as already described, where each qubit corresponding to the measurement of an $\hat{X}$ or $\hat{Y}$ operator must be properly transformed before the measurement is performed.

This is at the base of the procedure to get estimations of expectation values of generic operators: if $\hat{O}_N$ is a random Hermitian operator acting on N qubits, it can always be decomposed in the Pauli basis, i.e. as a real linear combination of all the possible strings of Pauli operators on the N qubits

$$\hat{O}_N = \sum_{j=0}^{2^N-1} c_j \hat{P}_j^{(N)}, \tag{1.48}$$

where $c_j \in \mathbb{R}$ and $\hat{P}_j^{(N)}$ are the Pauli operator strings defined in Eq. (1.19).
This means that the expectation value of $\hat{O}_N$ on a given state can always be obtained as the proper linear combination of the expectation values of all the Pauli operator strings on the same state.
To reduce experimental overhead, mutually commuting Pauli operator strings are usually grouped into common measurement settings, each requiring only a single basis rotation, minimizing in this way the number of distinct circuits while preserving unbiased estimation of $\langle \hat{O}_N \rangle_{\rho_N}$.

Chapter 2

Hubbard-Stratonovich transformation for quantum observables

2.1 Introduction

In this chapter, I present a method that relies on the Hubbard-Stratonovich transformation (HST, see Eq. (2.2)) to recast the action of the exponential of an operator into combinations of possibly simpler gates. Rather than allowing for the direct gate implementation, I argue that this formulation is more suitable for decomposing observables and estimating their expectation values.
This idea draws inspiration from the use of HST in statistical mechanics and condensed matter physics [38–40], where it is exploited to decouple non-linear, higher-order interaction terms by introducing an auxiliary classical field which couples linearly with the original fields: thanks to this classical field, the action becomes quadratic in the original fields so that they can be integrated out.
At the beginning of the chapter, in Section 2.2, the HST is introduced in its most generic form, while in Section 2.3 I explicitly show how to use it to formulate operator exponentials, arguing that the implementation of operators in these terms is generally not efficient, as it requires building linear combinations of operators. In Section 2.4 I propose an alternative way of exploiting the HST, as a procedure to estimate the expectation value of operators which can be written as exponentials of squared Hermitian operators: in the section, I show how this is indeed a viable possibility, but I also point out that there are severe limitations to a general formulation of observables by means of this transformation, reducing the range of applicability of the proposed algorithm to small systems or cases where symmetries can be efficiently exploited. In Sections 2.4.1 and 2.4.2 I focus on the case

of completely real and imaginary exponential observables, respectively, showing some simple examples.

2.2 The Hubbard-Stratonovich transformation (HST)

The Hubbard-Stratonovich transformation (HST) was developed by the Russian physicist R. L. Stratonovich [13] and popularized by the British physicist J. Hubbard [14], who applied it to the evaluation of the grand partition function of a system with a general pairwise interaction.
The HST is an exact mathematical identity, whose most straightforward definition is the following

$$e^{\frac{tx^2}{2}} = \int_{-\infty}^{\infty} \frac{e^{-\frac{y^2}{2t}}}{\sqrt{2\pi t}} e^{\pm xy} dy, \tag{2.1}$$

where $t \in \mathbb{R}^+$ is a real parameter, that must be positive to ensure the integral convergence, x is a real scalar field, y is another auxiliary scalar field, coupling linearly to x, and the $\pm$ sign makes the independence of the integral from this sign explicit, since it can be chosen at will. This identity is very easily demonstrated by completing the square at the exponent inside the integral and by solving the resulting Gaussian integral.
The HST can also be formulated in a Fourier-like form, though, which is more suitable to deal with complex quantities and more general mathematical objects

$$e^{-i\frac{\mathfrak{t}x^2}{2}} = \int_{-\infty}^{\infty} \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}} e^{\pm i\sqrt{i\mathfrak{t}}xy} dy \doteq I_g\left[e^{\pm i\sqrt{i\mathfrak{t}}xy}\right], \tag{2.2}$$

where $y \in \mathbb{R}$ is again an introduced auxiliary real scalar field, $\mathfrak{t} \doteq t - i\tau \in \mathbb{C}$ is a complex parameter (with t, τ being real quantities), x may be equivalently interpreted as both a field or, as it is the case in the context of quantum computing, a Hermitian operator, and I introduced the notation $I_g[A]$ to represent the integral of quantity A with Gaussian weight $g(y) \doteq e^{-y^2/2}/\sqrt{2\pi}$.
With these premises, Eq. (2.2) can be interpreted as a way of reformulating time-evolution operators, or more generally complex exponential operators, which we saw in Eq. (1.37) is the standard way that native operators are implemented in quantum hardware: by exploiting the reduction on the x order induced by the HST, one could in principle think of relying on it to replace the direct implementation of Eq. (2.2) LHS with a series of operators parametrized by y, $e^{\pm i\sqrt{\mathfrak{t}}\hat{x}y}$, which may be individually easier to implement. In principle, $\hat{x}$ could also be a sum of commuting terms, which would decouple thanks to the HST, possibly reducing many-qubit gates into combinations of fewer-qubit gates.

2.3 HST formulation of quantum operators

Let's consider a generic quantum many-body operator $\hat{G}_N(\mathfrak{t})$ acting on N qubits that can be written as a complex-time exponential of a Hermitian operator $\hat{S}_N$ in the following way

$$\hat{G}_N(\mathfrak{t}) \doteq e^{-i\frac{\mathfrak{t}\hat{S}_N}{2}} \quad | \quad \mathfrak{t} \in \mathbb{C}, \hat{S}_N = \hat{S}_N^\dagger. \tag{2.3}$$

Operators of this kind may be arbitrarily difficult to implement or to decompose as products of simpler, one-qubit and two-qubit operators. If $\hat{S}_N$ can be written as the square of a Hermitian operator, $\hat{S}_N = \hat{L}_N^2$, then the HST would be a valid alternative way of formulating $\hat{G}_N(\mathfrak{t})$ by introducing a continuously parametrized set of complex exponential operators in $\hat{L}_N$

$$\hat{G}_N(\mathfrak{t}) = e^{-i\frac{\mathfrak{t}\hat{L}_N^2}{2}} = \int_{-\infty}^{\infty} \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}} \hat{O}_N(\mathfrak{t}, y)dy = I_g\left[\hat{O}_N(\mathfrak{t}, y)\right], \tag{2.4}$$

where I defined the parametrized operators $\hat{O}_N(\mathfrak{t}, y)$ as the following

$$\hat{O}_N(\mathfrak{t}, y) \doteq e^{\pm i\sqrt{i\mathfrak{t}}\hat{L}_N y}. \tag{2.5}$$

In principle, operators $\hat{O}_N(\mathfrak{t}, y)$ may be easier to implement than $\hat{G}_N(\mathfrak{t})$, especially if $\hat{L}_N$ can be written as a sum of commuting operators, each acting on a fewer number of particles: in this case $\hat{O}_N(\mathfrak{t}, y)$ separates into a product of few-qubit operators, which is typically much simpler to realize than a general many-qubit operator.
One drawback of this formulation concerns unitarity: as it was briefly pointed out at the end of Section 1.2.2, only unitary operators may formally be executed on quantum hardware. If $\hat{G}_N(\mathfrak{t})$ is taken to be a unitary operator, though, Eq. (2.3) requires the parameter $\mathfrak{t}$ to be real, $\mathbb{C} \ni \mathfrak{t} \to t \in \mathbb{R}$, which in turn implies that $\hat{O}_N(t, y)$ is not a unitary operator

$$t \in \mathbb{R} \Longrightarrow \hat{O}_N(t, y) = e^{\pm i\sqrt{it}\hat{L}_N y} = e^{\pm\frac{1+i}{\sqrt{2}}\sqrt{t}\hat{L}_N y} = e^{\pm\sqrt{\frac{t}{2}}\hat{L}_N y} e^{\pm i\sqrt{\frac{t}{2}}\hat{L}_N y}. \tag{2.6}$$

This makes the HST formulation of unitary operators much more delicate to approach from a realistic perspective than it would be for the non-unitary operators defined by setting the $\mathfrak{t}$ parameter in Eq. 2.3 to a completely imaginary value $\mathbb{C} \ni \mathfrak{t} \to -i\tau \mid \tau \in \mathbb{R}$: in this second case, possibly less common in typical quantum simulation algorithms, all the operators $\hat{O}_N(-i\tau, y)$ would in fact be unitary and therefore more easily implementable on quantum machines

$$\tau \in \mathbb{R} \Longrightarrow \hat{O}_N(-i\tau, y) = e^{\pm i\sqrt{\tau}\hat{L}_N y}. \tag{2.7}$$

For the time being, I won't elaborate more on this topic, but I will pick it up again later in Section 2.4.2.

A second highly relevant drawback of the HST resides in its integral structure. While products of operators may be straightforwardly implemented by consecutively applying the terms appearing in the product to the qubits onto which they act, integrals of operators are much more difficult to realize, since they can be approximately regarded as operator sums, or in general linear combinations of operators.

A naïve Monte Carlo-inspired approach to the implementation of the HST formulation of operators would be that of sampling several values $\{\bar{y}_j\}_j$ from the Gaussian weight $g(y)$ in Eq. (2.8), building for each sample a different quantum circuit implementing the corresponding operator $\hat{O}_N(\mathfrak{t}, \bar{y}_j)$, computing the expectation value of some observable on the generated state and finally taking the average over the estimates from all the circuits

$$g(y) \doteq \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}}. \tag{2.8}$$

This algorithm would not work in general, though, since it only considers expectation values over states evolved by diagonal terms in the linear combination of operators $\hat{O}_N(\mathfrak{t}, y)$, neglecting the contributions coming from all the mixed terms. To be more clear, let us analyze what the action of the general operator $\hat{G}(\mathfrak{t})$ (I drop the N index, to favor readability) on the system density matrix $\hat{\rho}_0 \doteq \hat{\rho}(\mathfrak{t} = 0)$ would be

$$\begin{aligned} \rho_0 \to \rho(\mathfrak{t}) = \hat{G}(\mathfrak{t})\hat{\rho}_0\hat{G}^\dagger(\mathfrak{t}) = I_g\left[\hat{O}(\mathfrak{t}, y_1)\right]\hat{\rho}_0 I_g\left[\hat{O}^\dagger(\mathfrak{t}, y_2)\right] = \\ = \iint_{-\infty}^{\infty} \frac{e^{-\frac{y_1^2+y_2^2}{2}}}{2\pi}\hat{O}(\mathfrak{t}, y_1)\hat{\rho}_0\hat{O}^\dagger(\mathfrak{t}, y_2)dy_1dy_2. \end{aligned} \tag{2.9}$$

Eq. (2.9) is the properly evolved density matrix, correctly featuring all possible combinations of operators $\hat{O}(\mathfrak{t}, y)$. Differently, each operator $\hat{O}(\mathfrak{t}, \bar{y}_j)$, built from the sample $\bar{y}_j$ obtained through the previously described Monte Carlo-like strategy, would generate the state $\hat{O}(\mathfrak{t}, \bar{y}_j)\hat{\rho}_0\hat{O}^\dagger(\mathfrak{t}, \bar{y}_j)$, so that averaging over all samples would correspond to estimating the expectation value computed on the density matrix $\hat{\rho}'(\mathfrak{t})$ defined below, which lacks all the terms involving couples of operators with different values of the y parameter

$$\begin{aligned} \rho_0 \to \rho'(\mathfrak{t}) \doteq I_g\left[\hat{O}(\mathfrak{t}, y)\hat{\rho}_0\hat{O}^\dagger(\mathfrak{t}, y)\right] = \\ = \int_{-\infty}^{\infty} \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}}\hat{O}(\mathfrak{t}, y)\hat{\rho}_0\hat{O}^\dagger(\mathfrak{t}, y)dy. \end{aligned} \tag{2.10}$$

This means that in general the HST formulation of quantum gates cannot be easily implemented on quantum computers, and one should rely on more complicated and resource-demanding methods like Linear Combination of Unitaries (LCU) [41], which allow approximate realizations of the full linear combination of operators $\hat{O}(\mathfrak{t}, y)$ appearing inside the integral in Eq. (2.4): these methods in general require considerable qubit overhead and problematic circuit depths, which unfortunately renders them not viable for present day quantum computers.

2.4 HST formulation of quantum observables

If the HST appears particularly problematic to be realistically exploited as a way of implementing quantum operators, due to the reasons presented in the last Section 2.3, it is still possible to recast it as a way of estimating the expectation value of quantum observables. Eq. (2.9) in fact, states that, in order to faithfully evolve a state through a gate implemented with the HST formulation, it is not enough to act separately with the individual $\hat{O}_N(\mathfrak{t}, y)$ operators (see Eq. (2.6)), rather the combined action of all of them must be realized on the system. This issue is avoided when the focus is moved from the evolution of a quantum state via the HST formulation of an operator to the task of computing the expectation value on a given state of an observable expressed by means of the HST formula (see Eq. (2.4)).
Let us consider the expectation value of the observable $\hat{G}(\mathfrak{t})$ in Eq. (2.3) (I drop the N subscript, to favor readability) evaluated on an arbitrary density matrix $\hat{\rho}$: this can be expressed as the trace over the system of the product between $\hat{G}(\mathfrak{t})$ and $\hat{\rho}$, and by expanding the observable with the HST I get the following formula

$$\langle \hat{G}(\mathfrak{t}) \rangle_\rho = \mathrm{Tr}\Big[\hat{G}(\mathfrak{t})\hat{\rho}\Big] = I_g\left[\langle \hat{O}(\mathfrak{t}, y)\rangle_\rho\right] = \int_{-\infty}^{\infty} \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}} \langle \hat{O}(\mathfrak{t}, y)\rangle_\rho \, dy, \tag{2.11}$$

where I made use of the linearity of the trace to bring it inside the integral.
The expectation value in Eq. (2.11) is now expressed in a form that only contains one instance of the $\hat{O}(\mathfrak{t}, y)$ operators, which means that it can be correctly approximated by the algorithm proposed in the previous section: by taking a set of M samples $\{\bar{y}_j\}_{j\in\{1,\dots,M\}}$ from the Gaussian probability distribution $g(y)$ (see Eq. (2.8)), the observable expectation value $\langle \hat{G}(\mathfrak{t})\rangle_\rho$ may be approximated by $\mu(\{\langle \hat{\bar{O}}_j(\mathfrak{t})\rangle_\rho\}_j)$, i.e. by the average over the expectation values on state $\hat{\rho}$ of the set of operators $\{\hat{\bar{O}}_j(\mathfrak{t})\}_j \doteq \{\hat{O}(\mathfrak{t}, \bar{y}_j)\}_j$

$$\mu\left(\left\{\langle \hat{\bar{O}}_j(\mathfrak{t})\rangle_\rho\right\}_{j\in\{1,\dots,M\}}\right) \doteq \frac{1}{M}\sum_{j=1}^{M} \langle \hat{\bar{O}}_j(\mathfrak{t})\rangle_\rho \,. \tag{2.12}$$

In the limit of a large number of samples M we then obtain

$$\langle \hat{G}(\mathfrak{t}) \rangle_\rho = \lim_{M\to\infty} \left[\mu(\{\langle \hat{\tilde{O}}_j(\mathfrak{t}) \rangle_\rho\}_j) \right]. \tag{2.13}$$

This formulation in principle requires applying each operator in the set $\{\hat{\tilde{O}}_j(\mathfrak{t})\}_j$ to the qubit system and evaluating the correspondent expectation values: this operation is quite delicate, since $\hat{\tilde{O}}_j(\mathfrak{t})$ is in general not Hermitian, and we also saw that it may even be non-unitary.
In the following section, I present some examples of different possible typical situations and describe the HST performances in estimating observable expectation values.

2.4.1 HST estimation of non-unitary observables

Let us start by specifying the general operator $\hat{G}(\mathfrak{t})$ in Eq. (2.3) to be a non-unitary observable, which is more suitable for an HST formulation, as it is described in such terms by an integral over unitary operators. More specifically, let's define as an example such an operator to be the following purely real two-body exponential operator

$$\hat{G}(\mathfrak{t}) \to \hat{G}(-i\tau) = e^{-\frac{\tau}{2}(\hat{P}_1+\hat{P}_2)^2}, \tag{2.14}$$

where $\tau \in \mathbb{R}$ and $\hat{P}_1$, $\hat{P}_2$ are two identical Pauli operators (see Section 1.2.2) acting on qubits 1 and 2 respectively. Operators of this kind can be easily obtained from the well-known "σ-σ" interaction in Eq. (2.15)

$$\begin{aligned} e^{-\tau\vec{\hat{\sigma}}_1\cdot\vec{\hat{\sigma}}_2} = e^{-\tau\sum_{P\in\{X,Y,Z\}}\hat{P}_1\hat{P}_2} = \prod_{P\in\{X,Y,Z\}} e^{-\tau\hat{P}_1\hat{P}_2} = \\ = \prod_{P\in\{X,Y,Z\}} e^{-\tau\left(\frac{1}{2}\left(\hat{P}_1+\hat{P}_2\right)^2-\hat{\mathbb{I}}\right)} = e^{-\frac{3}{2}\tau} \prod_{P\in\{X,Y,Z\}} e^{-\frac{\tau}{2}\left(\hat{P}_1+\hat{P}_2\right)^2}, \end{aligned} \tag{2.15}$$

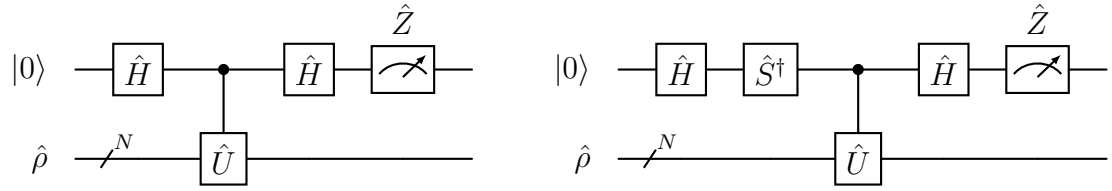
where I defined the Pauli vector $\vec{\hat{\sigma}} \doteq \{\hat{X}, \hat{Y}, \hat{Z}\}$ and exploited both the fact that the square of each Pauli operator is the identity operator and the following commutation property

$$[\hat{P}_1\hat{P}_2, \hat{P}'_1\hat{P}'_2] = 0 \quad \forall\ P, P' \in \{X, Y, Z\}. \tag{2.16}$$

Specifying Eq. (2.11) for the operator $\hat{G}(-i\tau)$ in Eq. (2.14) we get the following formula

$$\begin{aligned} \langle e^{-\frac{\tau}{2}(\hat{P}_1+\hat{P}_2)^2} \rangle_\rho = I_g\left[\langle e^{-i\sqrt{\tau}(\hat{P}_1+\hat{P}_2)y} \rangle_\rho \right] = \int_{-\infty}^{\infty} \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}} \langle e^{-i\sqrt{\tau}(\hat{P}_1+\hat{P}_2)y} \rangle_\rho \, dy = \\ = \int_{-\infty}^{\infty} g(y) \, \langle e^{-i\sqrt{\tau}\hat{P}_1 y} e^{-i\sqrt{\tau}\hat{P}_2 y} \rangle_\rho \, dy, \end{aligned} \tag{2.17}$$

where I introduced the Gaussian weight $g(y)$ defined in Eq. (2.8), I separated the exponential operator $\hat{O}(-i\tau, y) = e^{-i\sqrt{\tau}(\hat{P}_1+\hat{P}_2)y}$ exploiting the fact that $[\hat{P}_1, \hat{P}_2] = 0$, since these act on different particles, and I explicitly chose the symbol $\pm$ to be $-$.
The implementation of Eq. (2.17) through the Monte Carlo-like algorithm described in the last section is straightforward: a set of M samples $\{\bar{y}_j\}_{j\in\{1,\dots,M\}}$ is taken from the Gaussian weight $g(y)$ and for each sample different quantum circuits are built, measuring the unitary operators $\hat{\bar{O}}_j(\tau) \doteq \hat{O}(-i\tau, \bar{y}_j)$ on some density matrix state $\hat{\rho}$. With the retrieval of these expectation values, the Monte Carlo estimate $\mu(\{\langle\hat{\bar{O}}_j(\tau)\rangle_\rho\}_j)$ (see Eq. (2.12)) can be computed, obtaining an approximation to Eq. (2.17).
Among these steps, particular attention has to be paid to what concerns the measurement of operators $\hat{\bar{O}}_j(\tau)$: these operators are unitary but not necessarily Hermitian, which means that their expectation value is, in general, complex. The standard way to compute these expectation values would consist in decomposing each operator in the Pauli basis (see Eq. (1.23)), but this would in principle require an exponential number of measurements in the dimension of the system. As an alternative, it is possible to exploit the unitarity of the operators $\hat{\bar{O}}_j(\tau)$ to set up a circuit known as *Hadamard test* (HT): this exactly allows for estimating the real or imaginary part of the expectation value of generic unitary operators and only requires the addition of a single qubit to the system, called *ancilla*.
Figs. 2.1a and 2.1b show the circuit to compute the real and imaginary part, respectively, of the expectation value $\langle\hat{U}\rangle_\rho$, with $\hat{U}$ a generic unitary operator acting on N qubits. Both $\mathrm{Re}[\langle\hat{U}\rangle_\rho]$ and $\mathrm{Im}[\langle\hat{U}\rangle_\rho]$ are obtained by measuring, as described in Section 1.2.3, the Pauli $\hat{Z}$ operator on the ancilla at the end of the circuit.



(a) HT for measuring $\mathrm{Re}[\langle\hat{\bar{O}}_j(\tau)\rangle_\rho]$.

(b) HT for measuring $\mathrm{Im}[\langle\hat{\bar{O}}_j(\tau)\rangle_\rho]$.

Figure 2.1: The two Hadamard test (HT) circuits used to measure the real and imaginary part of the expectation value of a generic unitary operator $\hat{U}$ on the state $\hat{\rho}$ of a system of N qubits. $\hat{H}$ and $\hat{S}^\dagger$ are the Hadamard gate and the adjoint of the phase gate, respectively (see Eqs. (1.28) and (1.31)).

In the simple cases like the one considered here, where each operator $\hat{\bar{O}}_j(\tau)$ can be

written as the real-time evolution of a sum of K Hermitian operators $\hat{A}^{(k)}$ acting each on a different, single qubit, so that it can be separated as a product of K commuting operators $\hat{\hat{O}}_j^{(k)}(\tau)$

$$\hat{\hat{O}}_j(\tau) = e^{-i\sqrt{\tau}\bar{y}_j \sum_{k=1}^{K} \hat{A}^{(k)}} = \prod_{k=1}^{K} \hat{\hat{O}}_j^{(k)}(\tau), \tag{2.18}$$

where $\hat{\hat{O}}_j^{(k)}(\tau) \doteq e^{-i\sqrt{\tau}\bar{y}_j \hat{A}^{(k)}}$, then the controlled operators in the Hadamard tests can be realized as K consecutive $\hat{\hat{O}}_j^{(k)}(\tau)$ gates acting each on a different qubit k and all controlled on the ancilla, so that only K two-qubit controlled operators are required.
Keeping in mind this HST perspective, Eq. (2.17) can be explicitly formulated by separating the real and imaginary part of the expectation values to be measured

$$\langle e^{-\frac{\tau}{2}(\hat{P}_1+\hat{P}_2)^2} \rangle_\rho = I_g \left[\mathrm{Re}[\langle e^{-i\sqrt{\tau}\hat{P}_1 y} e^{-i\sqrt{\tau}\hat{P}_2 y} \rangle_\rho] \right] + i\, I_g \left[\mathrm{Im}[\langle e^{-i\sqrt{\tau}\hat{P}_1 y} e^{-i\sqrt{\tau}\hat{P}_2 y} \rangle_\rho] \right], \tag{2.19}$$

where the two terms in the RHS can be individually approximated by the sample averages $\mu_R(\{\langle \hat{\hat{O}}_j(\tau) \rangle_\rho\}_j)$ and $\mu_I(\{\langle \hat{\hat{O}}_j(\tau) \rangle_\rho\}_j)$, obtained through the previously described Monte Carlo-like algorithm

$$\mu_R(\{\langle \hat{\hat{O}}_j(\tau) \rangle_\rho\}_j) \doteq \frac{1}{M} \sum_{j=1}^{M} \mathrm{Re}[\langle \hat{\hat{O}}_j(\tau) \rangle_\rho]; \tag{2.20}$$

$$\mu_I(\{\langle \hat{\hat{O}}_j(\tau) \rangle_\rho\}_j) \doteq \frac{1}{M} \sum_{j=1}^{M} \mathrm{Im}[\langle \hat{\hat{O}}_j(\tau) \rangle_\rho]. \tag{2.21}$$

At this point, let us address the problem of understanding how to choose the number M of samples needed to be able to get a meaningful Monte Carlo approximation of the correct integral. To simplify the notation, let us define the following quantities

$$I_g[f(y)] \doteq \int_{-\infty}^{\infty} g(y) f(y) dy, \qquad \mu \doteq \frac{1}{M} \sum_{j=1}^{M} \bar{f}_j, \tag{2.22}$$

where $g(y)$ is a probability distribution, $f(y)$ a real function, and $\bar{f}_j \doteq f(\bar{y}_j)$, with $\{\bar{y}_j\}_{j\in\{1,\dots,M\}}$ a set of M samples taken from the probability distribution $g(y)$.
Primarily, the choice of the value of M depends on the desired relative error r of the Monte Carlo estimate, this being defined as the ratio between $\sigma[\mu]$, the standard deviation of the Monte Carlo estimate μ, and the absolute value of the full integral, $|I_g[f(y)]|$

$$r \doteq \frac{\sigma[\mu]}{|I[f(y)]|}, \tag{2.23}$$

where the standard deviation $\sigma[\mu]$ of the Monte Carlo estimate is defined through the following formula

$$\sigma[\mu] \doteq \frac{\sigma_g[f(y)]}{\sqrt{M}} \qquad | \qquad \sigma_g[f(y)] \doteq \sqrt{I_g[f^2(y)] - I_g[f(y)]^2}. \tag{2.24}$$

I note here that, for the time being, I'm considering the simplified picture in which samples $\bar{f}_j$ do not carry any measurement error: in the more realistic case where the samples $\{\bar{f}_j\}_j$ are measures performed on a quantum device, each of them will be characterized by an error $\sigma_m[\bar{f}_j]$ due to the finite number of shots performed for estimating the measurement (see Section 1.2.3), so that $\sigma[\mu]$ will take the following form

$$\sigma[\mu] \doteq \sqrt{\frac{\sigma_g[f(y)]^2 + \mathbb{E}\left[\{\sigma_m[\bar{f}_j]^2\}_j\right]}{M}} \quad | \quad \mathbb{E}\left[\{\sigma_m[\bar{f}_j]^2\}_j\right] \doteq \frac{1}{M}\sum_{j=1}^{M} \sigma_m[\bar{f}_j]^2. \tag{2.25}$$

Collecting these relations, one finds that the value of M corresponding to a certain relative error r can be obtained with the following relation

$$M = \left\lceil \left(\frac{\sigma_g[f(y)]}{|I_g[f(y)]|\, r} \right)^2 \right\rceil \propto \left(\frac{\sigma_g[f(y)]}{|I_g[f(y)]|} \right)^2 \doteq R_g(f(y)). \tag{2.26}$$

The ratio $R_g(f(y))$ is critical to the HST formulation of observable measurements: when the observable $\hat{G}(\mathfrak{t})$ is an exponential operator with completely imaginary parameter $\mathfrak{t} = -i\tau$, its operator norm will typically decay exponentially in τ, so that usually also $\left|\langle \hat{G}(-i\tau)\rangle_\rho\right| = \left|I_g\left[\langle \hat{O}(-i\tau, y)\rangle\right]\right|$ will have the same behavior. On the contrary, operators $\hat{O}(-i\tau, y)$ are unitary, which means that their operator norm will be of order unity, and therefore that the error $\sigma_g\left[\langle \hat{O}(-i\tau, y)\rangle_\rho\right]$ should behave differently, making this Monte Carlo approach effectively unfeasible for realistic τ values. This would lead to the ratio $R_g\left(\langle \hat{O}(-i\tau, y)\rangle_\rho\right)$ being exponential in τ, which in turn means that the number M of samples needed to estimate $\langle \hat{G}(-i\tau)\rangle_\rho$ up to a desired precision will also scale exponentially in τ.
This scenario is quite general, and would limit the applicability of the HST-based algorithm to small values of τ only, but it can still be avoided when the observable is the exponential of an operator $\hat{S}$ whose smallest eigenvalues, among those associated with eigenvectors that have non-zero overlap with the state $\hat{\rho}$ used in the measurement, are zero: in such a case, the exponential of such eigenvalues are 1 and do not decay as $\tau \to \infty$, which means that the HST and the proposed algorithm would in principle be able in this case to extract from $\hat{\rho}$ the superposition with these eigenstates.

As an example, let us consider again the observable $\hat{G}(-i\tau)$ in Eq. 2.14, specifying now the Pauli operator $\hat{P}$ as $\hat{Y}$

$$\hat{G}(-i\tau) = e^{-\frac{\tau}{2}(\hat{Y}_1+\hat{Y}_2)^2}. \tag{2.27}$$

Since the computational state $|00\rangle$ can be written in terms of the $\hat{S} \doteq (\hat{Y}_1 + \hat{Y}_2)^2$ eigenstates $|\psi_j\rangle$ as the following

$$|00\rangle = \frac{1}{\sqrt{2}} \left(|\psi_0\rangle + |\psi_1\rangle\right), \tag{2.28}$$

with the following definitions for the $\hat{S}$ eigenstates $|\psi_j\rangle$ and associated eigenvalues λ_j

$$\begin{aligned} |\psi_0\rangle \doteq |+\rangle &= \frac{1}{\sqrt{2}} \left(|00\rangle + |11\rangle\right), \quad \lambda_0 = 0, \\ |\psi_1\rangle \doteq |-\rangle &= \frac{1}{\sqrt{2}} \left(|00\rangle - |11\rangle\right), \quad \lambda_1 = 4, \end{aligned} \tag{2.29}$$

according to the reasoning made before, measuring $\hat{G}(-i\tau)$ on state $\hat{\rho} = |00\rangle\langle 00|$ will produce at greater τ values the overlap between $\hat{\rho}$ and $|\psi_0\rangle$

$$\langle e^{-\frac{\tau}{2}(\hat{Y}_1+\hat{Y}_2)^2} \rangle_\rho = I_g \left[\langle e^{-i\sqrt{\tau}\left(\hat{Y}_1+\hat{Y}_2\right)y} \rangle_\rho \right] \xrightarrow{\tau\to\infty} \left(\frac{1}{\sqrt{2}}\right)^2 = \frac{1}{2}, \tag{2.30}$$

and the error $\sigma_g \left[\langle e^{-i\sqrt{\tau}\left(\hat{Y}_1+\hat{Y}_2\right)y} \rangle_\rho \right]$ should also remain constant. As a note, I point out here that in this case all expectation values are real, so that there is no need to consider separately the imaginary part.

The described features are correctly reproduced in Fig. 2.2, where $I_g \left[\langle e^{-i\sqrt{\tau}\left(\hat{Y}_1+\hat{Y}_2\right)y} \rangle_\rho \right]$ and $\sigma_g \left[\langle e^{-i\sqrt{\tau}\left(\hat{Y}_1+\hat{Y}_2\right)y} \rangle_\rho \right]$ are represented as function of τ.

The ratio $R_g \left(\langle e^{-i\sqrt{\tau}\left(\hat{Y}_1+\hat{Y}_2\right)y} \rangle_\rho \right)$ calculated from these results is represented in Fig. 2.3: it is manifestly convergent, implying that the Monte Carlo estimate of $I_g \left[\langle e^{-i\sqrt{\tau}\left(\hat{Y}_1+\hat{Y}_2\right)y} \rangle_\rho \right]$ can be obtained with a number of samples M which is constant at large τ values. More precisely, $R_g \left(\langle e^{-i\sqrt{\tau}\left(\hat{Y}_1+\hat{Y}_2\right)y} \rangle_\rho \right) \xrightarrow{\tau\to\infty} 0.5$, which implies that $M = (2r^2)^{-1}$ samples are needed to approximate the exact value up to a relative error r (neglecting the error on the samples due to the finite statistics of the quantum measurements).

As soon as the operator $\hat{S}$ is modified so that its spectrum doesn't include zero eigenvalues anymore, the general exponential behavior predicted for R_g is correctly recovered. To show this, let us consider, for instance, the following operator $\hat{S}'$

$$\hat{S}' \doteq \left(\hat{Y}_1 + \hat{Y}_2 + \hat{\mathbb{I}}\right)^2, \tag{2.31}$$

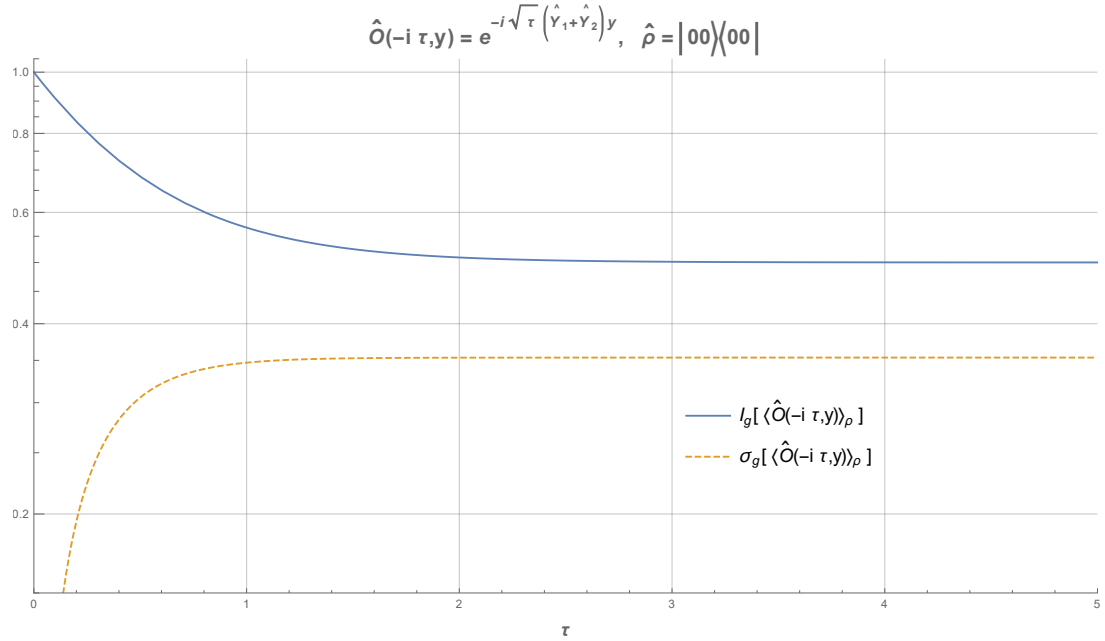


Figure 2.2: Exact values computed for $I_g\left[\langle e^{-i\sqrt{\tau}(\hat{Y}_1+\hat{Y}_2)y}\rangle_\rho\right]$ (solid blue, see Eq. (2.2)) and $\sigma_g\left[\langle e^{-i\sqrt{\tau}(\hat{Y}_1+\hat{Y}_2)y}\rangle_\rho\right]$ (dashed orange, , see Eq. (2.24)), with $\hat{\rho} = |00\rangle\langle 00|$ and $\hat{Y}_j$ being the Pauli operator $\hat{Y}$ acting on qubit j.

where $\hat{\mathbb{I}}$ is the identity operator acting on both qubits. The spectrum of this operator consists of the following eigenstates $|\psi'_j\rangle$ and eigenvalues λ'_j

$$\begin{aligned}
|\psi'_0\rangle &\doteq |\psi_0\rangle = |+\rangle\,, & \lambda'_0 &= 4,\\
|\psi'_1\rangle &\doteq \frac{1}{\sqrt{2}}\left(i\,|00\rangle + |10\rangle\right), & \lambda'_1 &= 4,\\
|\psi'_2\rangle &\doteq \frac{1}{\sqrt{2}}\left(i\,|00\rangle + |01\rangle\right), & \lambda'_2 &= 4,\\
|\psi'_3\rangle &\doteq \frac{1}{2}\left(|11\rangle - |00\rangle - i\left(|01\rangle + |10\rangle\right)\right), & \lambda'_3 &= 9.
\end{aligned} \tag{2.32}$$

When computing $\langle e^{-\frac{\tau}{2}\hat{S}'}\rangle_\rho = I_g\left[\langle e^{-i\sqrt{\tau}\left(\hat{Y}_1+\hat{Y}_2+\hat{\mathbb{I}}\right)y}\rangle_\rho\right]$ on the state $\hat{\rho} = |00\rangle\langle 00|$ then, the expectation value is expected to decay exponentially fast in τ, unlike the error $\sigma_g\left[\langle e^{-i\sqrt{\tau}\left(\hat{Y}_1+\hat{Y}_2+\hat{\mathbb{I}}\right)y}\rangle_\rho\right]$ which should always converge to a fixed value. In Fig. 2.4 these two quantities are represented, showing exactly the described behavior, which would prevent an efficient observable estimation through the HST formulation.

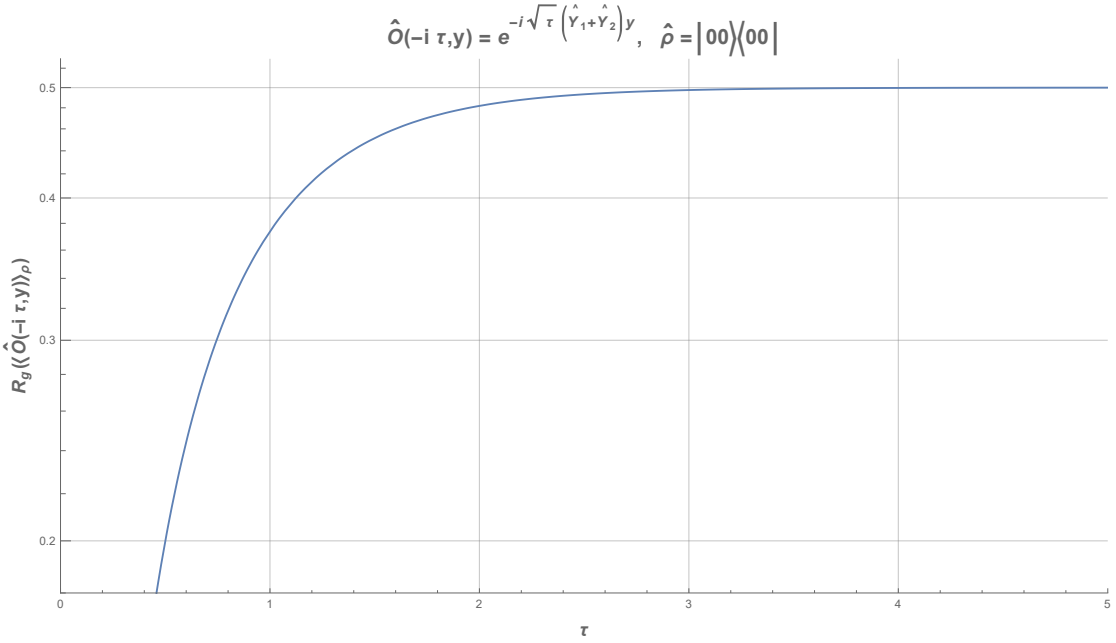


Figure 2.3: Exact computation of the ratio $R_g\left(\langle e^{-i\sqrt{\tau}(\hat{Y}_1+\hat{Y}_2)y}\rangle_\rho\right)$ (see Eq. (2.25)), with $\hat{\rho} = |00\rangle\langle 00|$ and $\hat{Y}_j$ being the Pauli operator $\hat{Y}$ acting on qubit j.

2.4.2 HST estimation of unitary observables

In the last Section 2.4.1, I presented an explicit way of implementing the HST formulation of observables written as completely imaginary-time evolutions of squared Hermitian operators. Here, I will instead focus on observables $\hat{G}(t \in \mathbb{R})$ that are real-time evolutions of squared Hermitian operators, and as such also unitary

$$\hat{G}(\mathfrak{t} \in \mathbb{C}) \to \hat{G}(t \in \mathbb{R}) = e^{-i\frac{t}{2}\hat{S}} = e^{-i\frac{t}{2}\hat{L}^2} \quad | \quad \hat{L} = \hat{L}^\dagger. \tag{2.33}$$

This case is less suitable for an HST formulation, since I already showed how the transformation reduces to a linear combination of non-unitary gates, which in principle cannot be applied on a real quantum device: specifying Eq. (2.11) to this case in fact, we get the following identity

$$\begin{aligned}\langle\hat{G}(t)\rangle_\rho = I_g\left[\langle\hat{O}(t,y)\rangle_\rho\right] = \int_{-\infty}^{\infty}\frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}}\langle e^{-i\sqrt{it}y\hat{L}}\rangle_\rho\, dy = \\ = \int_{-\infty}^{\infty} g(y)\,\langle e^{\sqrt{\frac{t}{2}}y\hat{L}}e^{-i\sqrt{\frac{t}{2}}y\hat{L}}\rangle_\rho\, dy,\end{aligned} \tag{2.34}$$

$g(y)$ being the zero-mean and unit-variance Gaussian probability density and where I defined $\hat{O}(t,y) \doteq e^{-i\sqrt{it}y\hat{L}} = e^{\sqrt{\frac{t}{2}}y\hat{L}}e^{-i\sqrt{\frac{t}{2}}y\hat{L}} \doteq \hat{O}_{\mathrm{u}}(t,y)\hat{O}_{\mathrm{nu}}(t,y)$, explicitly sep-

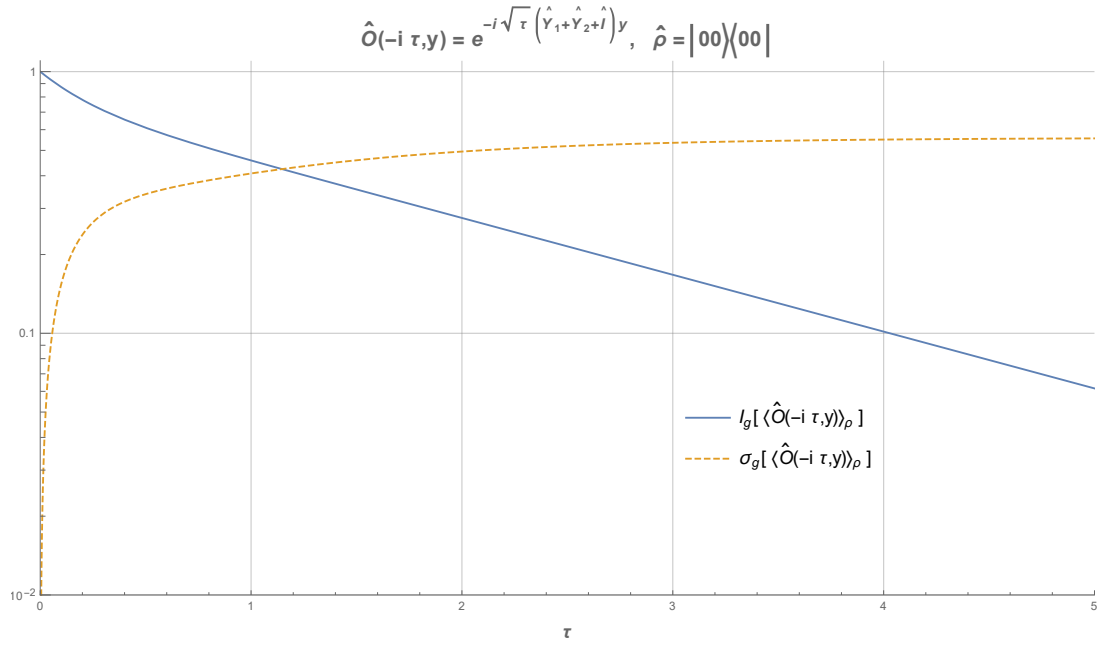


Figure 2.4: Exact values computed for $I_g\left[\langle e^{-i\sqrt{\tau}(\hat{Y}_1+\hat{Y}_2+\hat{\mathbb{I}})y}\rangle_\rho\right]$ (solid blue, see Eq. (2.2)) and $\sigma_g\left[\langle e^{-i\sqrt{\tau}(\hat{Y}_1+\hat{Y}_2+\hat{\mathbb{I}})y}\rangle_\rho\right]$ (dashed orange, see Eq. (2.24)), with $\hat{\rho} = |00\rangle\langle 00|$ and $\hat{Y}_j$ being the Pauli operator $\hat{Y}$ acting on qubit j.

arating the unitary contribution $\hat{O}_{\mathrm{u}}(t,y) \doteq e^{-i\sqrt{\frac{t}{2}}y\hat{L}}$ from the non-unitary one $\hat{O}_{\mathrm{nu}}(t,y) \doteq e^{\sqrt{\frac{t}{2}}y\hat{L}}$, and where I also arbitrarily chose the $\pm$ sign to be $-$ and $\sqrt{i} \doteq (1+i)/\sqrt{2}$.
I already explained how to deal with measuring the expectation values of unitary operators $\hat{O}_{\mathrm{u}}(t,y)$ (see Fig 2.1), but now there is the additional complication of the presence of the non-unitary term $\hat{O}_{\mathrm{nu}}(t,y)$: this in general is extremely non trivial, and, at the best of my knowledge, there isn't any efficient solution to compute the expectation value $\langle\hat{O}(t,y)\rangle_\rho$ for a generic operator $\hat{L}$.
Despite this strong limitation, I continue my analysis by proposing a method that allows the estimation of such expectation value when operator $\hat{L}$ is a real linear combination of N commuting Hermitian involutory operators, i.e. $\hat{L} = \sum_{n=1}^{N} a_n \hat{A}_n$ such that $\hat{A}_n = \hat{A}_n^\dagger$, $\hat{A}_n^2 = \hat{\mathbb{I}}$, $a_n \in \mathbb{R}$ and $\left[\hat{A}_n, \hat{A}_m\right] = 0 \quad \forall n,m$: this is useful as it

allows to rephrase the non-unitary part $\hat{O}_{\mathrm{nu}}(t,y)$ into the following form

$$\begin{aligned}\hat{O}_{\mathrm{nu}}(t,y) = \prod_{n=1}^{N} e^{\sqrt{\frac{t}{2}} y a_n \hat{A}_n} = \prod_{n=1}^{N} \cosh\left(y\, a_n \sqrt{\frac{t}{2}}\right) \hat{\mathbb{I}} + \left|\sinh\left(y\, a_n \sqrt{\frac{t}{2}}\right)\right| \hat{A}'_n = \\ \doteq \prod_{n=1}^{N} c_n(y,t)\hat{\mathbb{I}} + s_n(y,t)\hat{A}'_j,\end{aligned} \tag{2.35}$$

where I defined $c_n(y,t) \doteq \cosh\left(y\, a_n \sqrt{\frac{t}{2}}\right)$, $s_n(y,t) \doteq \left|\sinh\left(y\, a_n \sqrt{\frac{t}{2}}\right)\right|$ and $\hat{A}'_n \doteq \mathrm{sgn}\left[s_n(y,t)\right] \hat{A}_n$.
At this point, by extracting the factor $\mathcal{N}_n(y,t) \doteq c_n(y,t) + s_n(y,t)$ from each term in the product, it is possible to identify the coefficients in front of the operators as probabilities

$$\begin{aligned}\hat{O}_{\mathrm{nu}}(t,y) = \prod_{n=1}^{N} \mathcal{N}_n(y,t) \left(pc_n(y,t)\hat{\mathbb{I}} + ps_n(y,t)\hat{A}'_n\right) = \\ = \mathcal{N}(y,t) \prod_{n=1}^{N} pc_n(y,t)\hat{\mathbb{I}} + ps_n(y,t)\hat{A}'_n,\end{aligned} \tag{2.36}$$

where $\mathcal{N}(y,t) \doteq \prod_{n=1}^{N} \mathcal{N}_n(y,t)$ and both $pc_n(y,t) \doteq c_n(y,t)/\mathcal{N}_n(y,t)$ and $ps_n(y,t) \doteq s_n(y,t)/\mathcal{N}_n(y,t)$ take real values between 0 and 1 and are such that $pc_n(y,t) + ps_n(y,t) = 1 \quad \forall n$.
It is useful to further expand Eq. (2.36) by explicitly multiplying all the N pairs of terms, so that it reduces to the following

$$\hat{O}_{\mathrm{nu}}(t,y) = \mathcal{N}(y,t) \sum_{n=1}^{2^N} p_n(y,t)\hat{B}_n, \tag{2.37}$$

where $p_n(y,t) \doteq \prod_{m=1}^{N} x_m \quad | \, x_m \in \{pc_m(y,t), ps_m(y,t)\}$ and $\hat{B}_n \doteq \prod_{m=1}^{N} \hat{x}_m \quad | \, \hat{x}_m \in \{\hat{\mathbb{I}}, \hat{A}'_m\}$. With these definitions, $\{p_n(y,t)\}_n$ are still real numbers in the range $[0,1]$ and such that the following holds

$$\sum_{n=1}^{2^N} p_n(y,t) = \prod_{n=1}^{N} \left(pc_n(y,t) + ps_n(y,t)\right) = 1\ , \tag{2.38}$$

since $pc_n(y,t) + ps_n(y,t) = 1 \quad \forall\, n \in \{1,\ldots,N\}$, so that the set can be interpreted as a discrete probability distribution with 2^N possible outcomes, while $\{\hat{B}_n\}_n$ are still commuting Hermitian involutory operators.
To show this, consider two operators $\hat{\alpha}$ and $\hat{\beta}$ satisfying the following properties

$$\hat{\alpha} = \hat{\alpha}^\dagger, \quad \hat{\beta} = \hat{\beta}^\dagger, \quad \hat{\alpha}^2 = \hat{\beta}^2 = \hat{\mathbb{I}}, \quad \left[\hat{\alpha}, \hat{\beta}\right] = 0, \tag{2.39}$$

then, the operator $\hat{\gamma} \doteq \hat{\alpha}\hat{\beta}$ will also satisfy the same properties:

$$\hat{\gamma}^\dagger = \left(\hat{\alpha}\hat{\beta}\right)^\dagger = \hat{\beta}^\dagger\hat{\alpha}^\dagger = \hat{\beta}\hat{\alpha} = \hat{\alpha}\hat{\beta} = \hat{\gamma}, \tag{2.40}$$

$$\hat{\gamma}^2 = \hat{\gamma}\hat{\gamma} = \hat{\alpha}\hat{\beta}\hat{\alpha}\hat{\beta} = \hat{\alpha}\hat{\alpha}\hat{\beta}\hat{\beta} = \hat{\mathbb{I}}\hat{\mathbb{I}} = \hat{\mathbb{I}}. \tag{2.41}$$

Substituting Eq. (2.37) into Eq. (2.34) and introducing the following definition

$$I_{g\cdot\{p\}}\left[f_n(y)\right] \doteq \int_{-\infty}^{\infty} g(y) \sum_{n=1}^{2^N} p_n(y) f_n(y) dy, \tag{2.42}$$

the identity reported below is obtained

$$\begin{aligned}
\langle \hat{G}(t) \rangle_\rho &= \int_{-\infty}^{\infty} g(y)\mathcal{N}(y,t) \sum_{n=1}^{2^N} p_n(y,t) \left\langle \hat{B}_n \hat{O}_{\mathrm{u}}(t,y) \right\rangle_\rho dy = \\
&= \int_{-\infty}^{\infty} g(y)\mathcal{N}(y,t) \sum_{n=1}^{2^N} p_n(y,t) \left\langle \hat{O}'_{\mathrm{u}_n}(t,y) \right\rangle_\rho dy = \\
&= I_{g\cdot\{p\}} \left[\mathcal{N}(y,t) \ \left\langle \hat{O}'_{\mathrm{u}_n}(t,y) \right\rangle_\rho \right],
\end{aligned} \tag{2.43}$$

where the operators $\hat{O}'_{\mathrm{u}_n}(t,y) \doteq \hat{B}_n \hat{O}_{\mathrm{u}}(t,y)$ are still unitary and therefore measurable through the Hadamard test described in last Section 2.4.1 (always separating the estimation of the real and imaginary parts of the expectation value)

$$\hat{O}'_{\mathrm{u}_n}(t,y)\hat{O}'_{\mathrm{u}_n}(t,y)^\dagger = \hat{B}_n \hat{O}_{\mathrm{u}}(t,y)\hat{O}_{\mathrm{u}}(t,y)^\dagger \hat{B}_n^\dagger = \hat{B}_n \hat{B}_n = \hat{\mathbb{I}}, \tag{2.44}$$

having used the fact that operators $\hat{O}_{\mathrm{u}}(t,y)$ are unitary, $\hat{O}_{\mathrm{u}}(t,y)^{-1} = \hat{O}_{\mathrm{u}}(t,y)^\dagger$ and operators $\hat{B}_n$ are Hermitian, $\hat{B}_n = \hat{B}_n^\dagger$ and involutory, $\hat{B}_n^2 = \hat{\mathbb{I}}$.
At this point, Eq. (2.43) can be approximated by two different Monte Carlo algorithms, sharing the same average but differing in their expected variance.
The first alternative consists of sampling a set of M values $\{\bar{y}_j\}_{j\in\{1,\dots,M\}}$ from the Gaussian distribution $g(y)$, using each $\bar{y}_j$ sample to build the corresponding discrete probability distribution $\{p_n(\bar{y}_j,t)\}_{n\in\{1,\dots,2^N\}}$. From each of these M, 2^N-valued, discrete probability distributions, a single sample $\bar{n}_j$ is extracted, which is then used to build the unitary operator $\hat{\bar{O}}'_{u_j}(t) \doteq \hat{O}'_{\mathrm{u}_{\bar{n}_j}}(t,\bar{y}_j)$, whose expectation value on state ρ is measured on the quantum computer by means of the Hadamard test, separating the real and imaginary part if needed. The Monte Carlo approximation is finally obtained by taking the average of the product between each measurement sample $\langle \hat{\bar{O}}'_{u_j}(t) \rangle_\rho$ and the corresponding normalization

factor $\bar{\mathcal{N}}_j(t) \doteq \mathcal{N}(\bar{y}_j, t)$

$$\begin{aligned}\langle \hat{G}(t) \rangle_\rho = I_{g\cdot\{p\}} \left[\mathcal{N}(y,t) \langle \hat{O}'_{\mathrm{u}_n}(t,y) \rangle_\rho \right] = \lim_{M\to\infty} \mu \left(\left\{ \bar{\mathcal{N}}_j(t) \langle \hat{\bar{O}}'_{u_j}(t) \rangle_\rho \right\}_{j\in\{1,\dots,M\}} \right) = \\ \doteq I_{g\cdot\{p\}} \left[x_n(y,t) \right] \doteq \lim_{M\to\infty} \mu \left(\{ \bar{x}_j(t) \}_{j\in\{1,\dots,M\}} \right),\end{aligned} \tag{2.45}$$

with $\mu(\cdot) \doteq \mu_R(\cdot) + i\mu_I(\cdot)$, $\mu_R(\cdot)$ and $\mu_i(\cdot)$ being defined in Eqs. (2.20) and (2.21), respectively, and where I simplified the notation by defining $x_n(y,t) \doteq \mathcal{N}(y,t) \langle \hat{O}'_{\mathrm{u}_n}(t,y) \rangle_\rho$ and $\bar{x}_j(t) \doteq \bar{\mathcal{N}}_j(t) \langle \hat{\bar{O}}'_{u_j}(t) \rangle_\rho$.

Adapting Eq. (2.24) to the described algorithm, the standard deviation on this Monte Carlo estimate is the following, always assuming the simplified picture of no error coming from the quantum measurements

$$\sigma_1 \doteq \frac{\sigma_{g\cdot\{p\}} \left[x_n(y,t) \right]}{\sqrt{M}} \qquad | \qquad \sigma_{g\cdot\{p\}} \left[x_n(y,t) \right] \doteq \sqrt{I_{g\cdot\{p\}} \left[x_n^2(y,t) \right] - I_{g\cdot\{p\}} \left[x_n(y,t) \right]^2}. \tag{2.46}$$

Going back to Eq. (2.43), as already anticipated it is possible to devise a second Monte Carlo algorithm for the approximation of such an expectation value, differing in the expected error: everything works the same way as the first proposed alternative, with the only difference that the $\{\bar{y}_j\}_j$ samples are taken from the following normalized continuous probability distribution $d(y,t)$ rather than from the Gaussian $g(y)$

$$d(y,t) \doteq \frac{g(y)\mathcal{N}(y,t)}{\mathcal{M}(t)} \qquad | \qquad \mathcal{M}(t) \doteq \int_{-\infty}^{\infty} g(y)\mathcal{N}(y,t)dy, \tag{2.47}$$

where I introduced the normalization term $\mathcal{M}(t)$. Eq. (2.43) can in fact be formulated in terms of this probability distribution $d(y,t)$ by explicitly extracting the factor $\mathcal{M}(t)$ from the integral

$$\begin{aligned}\langle \hat{G}(t) \rangle_\rho &= \int_{-\infty}^{\infty} g(y)\mathcal{N}(y,t) \sum_{n=1}^{2^N} p_n(y,t) \langle \hat{O}'_{\mathrm{u}_n}(t,y) \rangle_\rho \, dy = \\ &= \int_{-\infty}^{\infty} \mathcal{M}(t) \frac{g(y)\mathcal{N}(y,t)}{\mathcal{M}(t)} \sum_{n=1}^{2^N} p_n(y,t) \langle \hat{O}'_{\mathrm{u}_n}(t,y) \rangle_\rho \, dy = \\ &= \int_{-\infty}^{\infty} \mathcal{M}(t) d(y,t) \sum_{n=1}^{2^N} p_n(y,t) \langle \hat{O}'_{\mathrm{u}_n}(t,y) \rangle_\rho \, dy = \\ &\doteq I_{d\cdot\{p\}} \left[\mathcal{M}(t) \langle \hat{O}'_{\mathrm{u}_n}(t,y) \rangle_\rho \right] = I_{d\cdot\{p\}} \left[x'_n(y,t) \right],\end{aligned} \tag{2.48}$$

where for conciseness I introduced the new quantity $x'_n(y,t) \doteq \mathcal{M}(t) \langle \hat{O}'_{\mathrm{u}_n}(t,y) \rangle_\rho$. The Monte Carlo estimate of Eq. (2.48) in this case can be easily written as the following

$$\langle \hat{G}(t) \rangle_\rho = I_{d\cdot\{p\}} \left[x'_n(y,t) \right] \doteq \lim_{M\to\infty} \mu \left(\left\{ \bar{x}'_j(t) \right\}_{j\in\{1,\dots,M\}} \right), \tag{2.49}$$

where I also introduced the samples $\bar{x}'_j(t) \doteq \mathcal{M}(t) \langle \hat{\bar{O}}'_{u_j}(t) \rangle_\rho$.

This time, the standard deviation on the Monte Carlo approximation of $\langle \hat{G}(t) \rangle_\rho$ is given by the following identity

$$\sigma_2 \doteq \frac{\sigma_{d\cdot\{p\}} \left[x'_n(y,t) \right]}{\sqrt{M}} \qquad | \qquad \sigma_{d\cdot\{p\}} \left[x'_n(y,t) \right] \doteq \sqrt{I_{d\cdot\{p\}} \left[x'^2_n(y,t) \right] - I_{d\cdot\{p\}} \left[x'_n(y,t) \right]^2}. \tag{2.50}$$

Despite this possibly rather appealing formulation, both of these implementations are plagued by the same exponential curse which was already faced in the previous section, and therefore shares the same strong limitations: this time the expectation value $\langle \hat{G}(t) \rangle_\rho$ is expected to be of order unity, since $\hat{G}(t)$ is a unitary operator, but the terms $\mathcal{N}_n(y,t)$ are instead exponentially increasing in t, which would typically reflect in diverging errors on the Monte Carlo estimates.

To show this behavior more explicitly, I present in the following the results computed for a case analogous to the one studied in the last section, specifying the described algorithm for a two-qubit system where I set $\hat{L} = \hat{Y}_1 + \hat{Y}_2$ in Eq. (2.33), with $\hat{Y}_k$ the $\hat{Y}$ Pauli operator acting on particle k, so that the studied observable is the following unitary operator

$$\hat{G}(t \in \mathbb{R}) = e^{-i\frac{t}{2}\hat{L}^2} = e^{-i\frac{t}{2}\left(\hat{Y}_1 + \hat{Y}_2\right)^2}. \tag{2.51}$$

The expectation value of such an observable on the state $\hat{\rho} = |00\rangle \langle 00|$ is easily found to be the following

$$\langle \hat{G}(t) \rangle_\rho = e^{-it} \cos(t), \tag{2.52}$$

which, being complex, requires separating each quantum measurement performed on the system into real and imaginary parts. To present the two contributions together, Fig. 2.5 shows the absolute value of Eq. (2.52) in green, together with the exact standard deviations computed with the two different proposed methods (see Eqs. (2.46) and (2.50)).

The exponential behavior is clearly evident: both standard deviations soon completely dominate the observable expectation value by several orders of magnitude, making the Monte Carlo approach practically unfeasible for the general case, as it would require an exponential number of samples to give meaningful estimates.

Despite this, the two methods differ substantially in the expected error, with the standard deviation corresponding to the sampling of the probability density

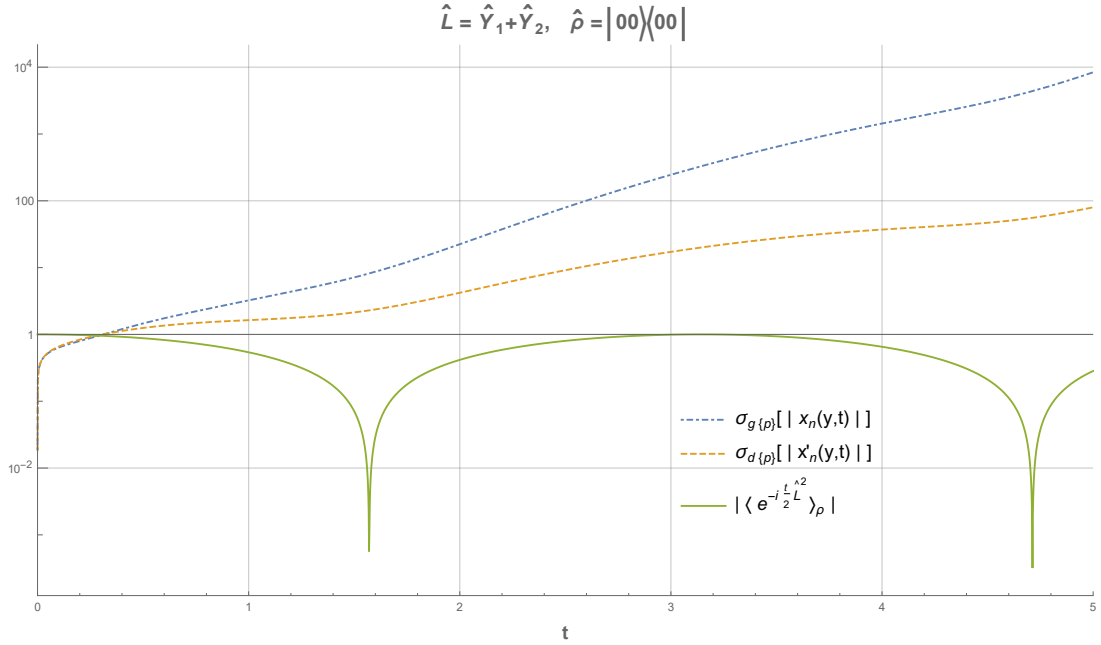


Figure 2.5: Exact values computed for $\left|\langle e^{-i\frac{t}{2}\hat{L}^2}\rangle_\rho\right|$ (solid green, see Eq. (2.52)), $\sigma_{g\cdot\{p\}}\left[\,|x_n(y,t)|\,\right]$ (dot-dashed blue, see Eq. (2.46)) and $\sigma_{d\cdot\{p\}}\left[\,|x'_n(y,t)|\,\right]$ (dashed orange, see Eq. (2.50)), where $\hat{L} = \hat{Y}_1 + \hat{Y}_2$, $\hat{\rho} = |00\rangle\langle 00|$ and the quantities $x_n(y,t)$ and $x'_n(y,t)$ are as defined in Section 2.4.2.

$d(y,t)$ in Eq. (2.47) being evidently smaller than the one obtained through the other method, i.e. by sampling only the Gaussian $g(y)$. To better clarify this point, I recall Eq. (2.26), where the number of needed Monte Carlo samples M was given, depending on the relative error r of the approximation: the same quantity, formulated for the two algorithms proposed in this section, takes the following form

$$M = \left\lceil \left(\frac{\sigma_{g\cdot\{p\}}\left[\,|x_n(y,t)|\,\right]}{\left|\langle \hat{G}(t)\rangle_\rho\right| r} \right)^2 \right\rceil \propto \left(\frac{\sigma_{g\cdot\{p\}}\left[\,|x_n(y,t)|\,\right]}{\left|\langle \hat{G}(t)\rangle_\rho\right|} \right)^2 \doteq R_{g\{p\}}(|x_n(y,t)|), \tag{2.53}$$

$$M = \left\lceil \left(\frac{\sigma_{d\cdot\{p\}}\left[\,|x'_n(y,t)|\,\right]}{\left|\langle \hat{G}(t)\rangle_\rho\right| r} \right)^2 \right\rceil \propto \left(\frac{\sigma_{d\cdot\{p\}}\left[\,|x'_n(y,t)|\,\right]}{\left|\langle \hat{G}(t)\rangle_\rho\right|} \right)^2 \doteq R_{d\{p\}}(|x'_n(y,t)|). \tag{2.54}$$

In Fig. 2.6 the two R ratios given in Eqs. (2.53) and (2.54) are represented, demonstrating both the exponential scaling of M in the t parameter and the greater

efficiency obtained by sampling the probability density $d(y,t)$ rather than $g(y)$.

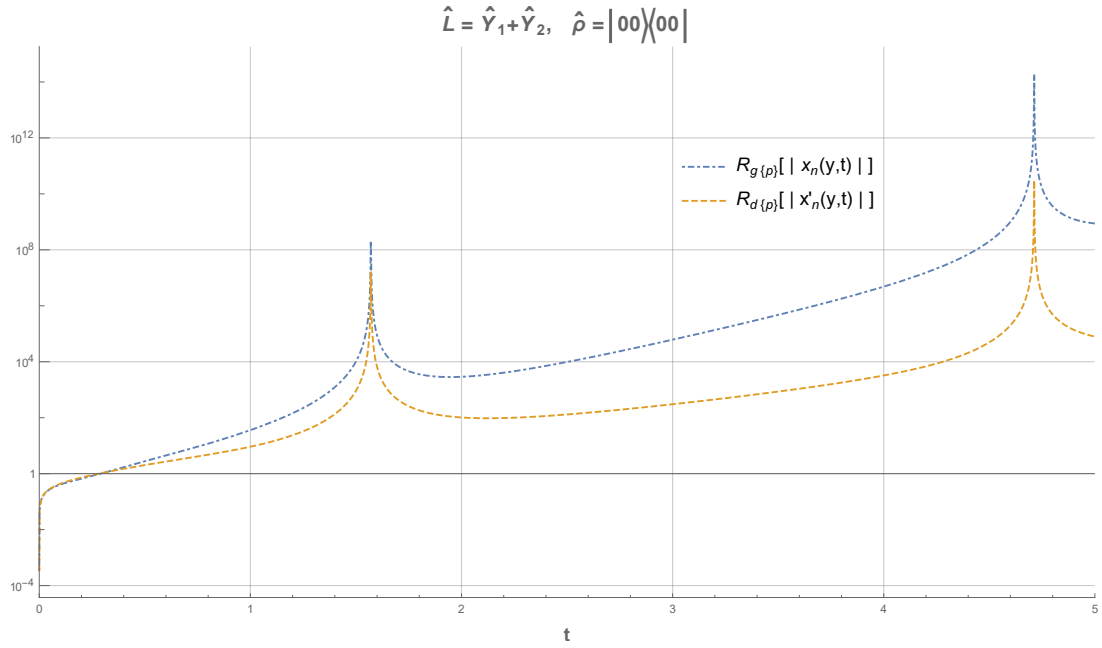


Figure 2.6: Exact values computed for $R_{g\{p\}}\left[\,|x_n(y,t)|\,\right]$ (dot-dashed blue, see Eq. (2.53)) and $R_{d\{p\}}\left[\,|x'_n(y,t)|\,\right]$ (dashed orange, see Eq. (2.54)), where $\hat{L} = \hat{Y}_1 + \hat{Y}_2$, $\hat{\rho} = |00\rangle\langle 00|$ and the quantities $x_n(y,t)$ and $x'_n(y,t)$ are as defined in Section 2.4.2.

2.5 Final remarks

In this chapter, I have explored some of the possible applications of the Hubbard-Stratonovich transformation in the context of quantum computing. This powerful exact identity enables the reformulation of exponential quantum operators as linear combinations of different exponential gates, which may, in some cases, be more easily realizable on quantum hardware. Although direct implementation of quantum gates via this approach is impractical, the transformation holds greater promise for recasting observables and facilitating the estimation of their expectation values through Monte Carlo–based methods. While the proposed implementation is neither optimal nor particularly efficient in the most general case, it remains formally correct and may prove useful in specific scenarios, in particular for smaller systems, selected observables, or tailored quantum states. The main limitation lies in the exponential scaling of the number of Monte Carlo samples required to obtain accurate estimates of the exact expectation values.
Possible further developments to this work include a more focused investigation

of the principal and most practical applications of this approach, as well as quantitatively testing its performance against other well-established deterministic and stochastic methods.

Chapter 3

Thermal quantum simulation of neutrino systems

3.1 Introduction

Neutrinos are among the most elusive and fundamental particles in the Standard Model. Electrically neutral and extremely light, they interact only via the weak force and gravity, making them difficult to detect yet crucial to a wide array of physical processes (see e.g. [42] for a review). They play key roles in beta decay, the dynamics of the early universe, and the energy transport mechanisms in astrophysical events such as core-collapse supernovae and neutron star mergers. In these high-energy environments, neutrinos interactions and their collective behavior can significantly influence the overall evolution of the system [43–49].
A particularly rich feature of neutrino physics is their flavor oscillation, i.e. the phenomenon whereby neutrinos switch between different flavor states (electron, muon, tau) as they propagate. This process arises from the mismatch between flavor and mass eigenstates and is a purely quantum mechanical effect. In dense astrophysical media, such as the neutrino-spheres of supernovae or the early universe plasma, the evolution of neutrino flavors becomes highly nonlinear due to neutrino-neutrino interactions: these interactions give rise to collective flavor oscillations, self-induced coherence, and potentially chaotic dynamics, all of which are encoded in an effective flavor Hamiltonian [18, 50–57].
Importantly, the effective neutrino flavor Hamiltonian in such environments is known to exhibit features of quantum chaos, including level repulsion and broad, bell-shaped energy spectra [57]. As a result, the long-time dynamics of the system can resemble thermalization, even though the underlying evolution is unitary. In such chaotic regimes, a generic initial state, one that overlaps significantly with many energy eigenstates, is expected to evolve in such a way that the long-time

average of few-body observables matches their thermal expectation values at some effective temperature. This is consistent with the predictions of the *Eigenstate Thermalization Hypothesis* (ETH), a central idea in quantum statistical mechanics [58].

This leads to what we refer to as the *neutrino problem*: the challenge of identifying, for a given initial quantum state evolving under the chaotic neutrino Hamiltonian, the effective temperature that characterizes the long-time behavior of few-body observables. More precisely, the problem involves finding the temperature for which the thermal expectation value of an observable matches its long-time average in the unitary evolution. Conceptually, this is justified by the assumption that the initial state is a superposition of "typical" states, and statistically, such a superposition should behave thermally for simple, few-body observables. However, due to the complex structure of the Hamiltonian and the inaccessibility of full eigenspectra in realistic models, this inverse problem is both analytically and numerically challenging.

Further complicating the situation is the fact that thermal states are fundamentally non-unitary. While flavor oscillations in vacuum or coherent media are described by unitary time evolution, thermalization requires imaginary-time evolution or other forms of statistical averaging, which lie outside the standard unitary framework: this makes the computation of thermal expectation values especially difficult in many-body quantum systems such as those describing dense neutrino gases [16, 59, 60].

In this context, quantum algorithms offer new tools for simulating and understanding thermal behavior in strongly correlated quantum systems. One such method is the Quantum METTS (Minimally Entangled Typical Thermal States) algorithm [16] exploited in this work, which provides a way to estimate thermal expectation values by sampling over a sequence of pure states with low entanglement. These states can be generated through repeated measurements and unitary approximation to imaginary-time evolutions, and their ensemble average approximates the thermal state at a specified temperature. Avoiding full density matrix evolution and leveraging typicality, QMETTS is in principle scalable to larger systems.

In this work, the QMETTS algorithm is applied, as a proof of concept, to a small two-flavor neutrino system consisting of four particles, where the thermal evolution of the system is expanded on a basis of *Givens rotations*. Givens rotations constitute in fact a basis for number-conserving operators [17] like the neutrino Hamiltonian, and their local nature combined with the SWAP network (see e.g. [22, 61, 62]) is particularly suitable to the QMETTS approach. The goal is not to exploit the sampling efficiency of QMETTS, but rather to test whether the procedure correctly reproduces thermal expectation values in a well-controlled

setting and for a non-local Hamiltonian. Since the system is small, thermal observables are extracted via full two-body quantum state tomography rather than relying on statistical sampling from the METTS ensemble. Despite the simplicity of the implementation, we believe this serves as a foundational step toward validating quantum thermalization techniques in neutrino contexts.
While the neutrino problem itself is not solved here, the methods explored may contribute to future strategies for addressing it. In larger systems, where classical approaches become infeasible, the QMETTS algorithm and related quantum techniques may offer a practical path for extracting thermal observables and estimating effective temperatures from chaotic flavor dynamics.
This chapter starts with the introduction in Section 3.1.1 of the considered neutrino Hamiltonian in the two-flavor approximation, which main symmetries are presented and explained in detail in Section 3.1.2. In Section 3.2 basic concepts about quantum system thermal evolution are treated, including integrability, thermal density matrices and imaginary-time evolution. The QMETTS algorithm for the quantum computation of observable thermal expectation values is described in Section 3.3, specifically its formulation in terms of Givens rotations. The specific details of the quantum simulation implementation for the neutrino system are given in Section 3.4. Finally, the results of both noiseless and noisy simulations performed with Qiskit software are presented in Section 3.5.

3.1.1 Neutrino Hamiltonian

Much of the structure of this section follows [63].
A complete description of a neutrino system must account for the fact that neutrinos can exist in three distinct flavor states, each corresponding to a different lepton generation: the electron neutrino (ν_e), the muon neutrino (ν_μ), and the tau neutrino (ν_τ). In the environment of a core-collapse supernova, electrons are vastly more abundant than muons or tauons, leading to significantly different behavior for electron neutrinos compared to the other two flavors [64, 65]. In such scenarios, it is often convenient to adopt a simplified two-flavor approximation, where neutrinos are classified as either electron neutrinos (ν_e) or a collective "other" flavor, commonly denoted as ν_x.
This simplification proves particularly advantageous when formulating the problem on a quantum computer, as it enables a direct mapping between the two neutrino flavors and the two states of a qubit. In this framework, the neutrino flavor state can be represented as an $\mathcal{SU}(2)$ spinor, which naturally corresponds to the $\mathfrak{su}(2)$ algebra of qubits. Consequently, each qubit can be utilized to encode the flavor state of a neutrino. Encoding the state of a three-flavor neutrino system on a qubit-based quantum computer is also feasible in principle, at the price of

loosing the correspondence with the simulator algebra: an ideal system suitable for a similar treatment would be a quantum processor based on qutrits rather than qubits. The interested reader can find more information in [66].
Calling a_e and a_x the complex probability amplitudes for a neutrino of having flavor ν_e and ν_x respectively, the spinor describing its state $|\nu\rangle$, written in the flavor basis, will simply be

$$|\nu\rangle = \begin{pmatrix} a_e \\ a_x \end{pmatrix}. \tag{3.1}$$

It's also possible to write the state $|\nu\rangle$ in the different basis made by the 2 eigenstates of the mass (or vacuum energy) operator $\{|\nu_1\rangle, |\nu_2\rangle\}$, where these will differ from the flavor eigenstates $\{|\nu_e\rangle, |\nu_x\rangle\}$ since the two operators do not commute: each eigenstate will be a linear combination of the two eigenstates belonging to the other basis and the relation between them can be expressed by a one-parameter matrix $\hat{U}$:

$$\begin{pmatrix} a_e \\ a_x \end{pmatrix} = \hat{U} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \doteq \begin{pmatrix} \cos\theta_\nu & \sin\theta_\nu \\ -\sin\theta_\nu & \cos\theta_\nu \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}, \tag{3.2}$$

where $\theta_\nu \in \left[0, \frac{\pi}{2}\right]$ is known as the *mixing angle* and a_1, a_2 can be interpreted as the probability amplitudes for the neutrino of having either mass m_1 or m_2, respectively.
The $\hat{U}$ matrix can be generalized to the case of three neutrino flavors, in which case it's known as the Pontecorvo-Maki-Nakagawa-Sakata (PMNS) matrix and it will be characterized by four distinct parameters.
Given that the aim is to describe ultra-relativistic neutrinos emitted from a core-collapse supernova at a distance of the order of 100 km from the surface of the proto-neutron star, where the neutrino density is high enough for neutrino flavor oscillation to be relevant, the relevant contribution to the Hamiltonian will come from three main terms: the one-body vacuum evolution H_{vac}, the one-body matter interaction H_{mat} and the two-body $\nu - \nu$ interaction $H_{\nu\nu}$ originating from the neutrino-neutrino forward scattering.
As a note, I hereby clarify that the role of anti-neutrinos will be neglected in the following derivation. While they are indeed very important for realistic simulations, the current work aims to establish foundational methods, and incorporating them is planned for future extensions. Importantly, adding anti-neutrinos does not introduce any fundamentally new theoretical or computational challenges; it primarily increases the system size and complexity, which can be handled once the basic framework is fully validated.

– Evolution in vacuum, H_{vac}

The first term to consider in the Hamiltonian regulating neutrino flavor evolution is the one-body vacuum term H_{vac}: this is the simplest term giving rise to neutrino

flavor oscillation and it's due to the differences between flavor and mass eigenstates. As already introduced, mass eigenstates $|\nu_1\rangle$ and $|\nu_2\rangle$ are also eigenstates of the vacuum energy, and as such they will have a well-defined value for the vacuum energy: this means that they satisfy the following equation

$$i\frac{\partial}{\partial t}|\nu_j(\vec{x},t)\rangle = \hat{H}_{vac}|\nu_j(\vec{x},t)\rangle = E_j|\nu_j(\vec{x},t)\rangle\,, \tag{3.3}$$

where I'm describing a neutrino in a three-dimensional space parametrized by the variable $\vec{x}$, t is the time, i is the imaginary unit, $j \in \{1,2\}$ is an index for the specific mass eigenstate and E_j is the vacuum energy associated to the eigenvalue $|\nu_j\rangle$. I specify that the same does not hold for the neutrino eigenstates $|\nu_e\rangle$ and $|\nu_x\rangle$, since each of these is a linear combination of states having different vacuum energy eigenvalues.
A solution to Eq. (3.3) is given by plane waves, each of which is characterized by a well-defined value for its momentum: this means that each mass eigenstate at any given position $\vec{x}$ and time t can be expressed in terms of its original state in the origin position in the following way

$$|\nu_j(\vec{x},t)\rangle = e^{-i(E_j t - \vec{p}_j\vec{x})}|\nu_j(0,0)\rangle\,, \tag{3.4}$$

where now $\vec{p}_j$ is the momentum associated to the mass eigenstate $|\nu_j\rangle$.
In general, a possibly more realistic solution to Eq. (3.3) would consist of a wave packet, which would introduce in the description a distribution of momenta for every state rather than a single value: despite being more complete, I do not consider such solution here and limit the analysis by considering mass eigenstates as plane waves, since they do form a complete basis we could use to describe wave-packets. The interested reader can refer to references [67–69] for a deeper exploration of the wave packet formulation.
Having introduced the concept of momentum in the picture, it's now possible to express the relativistic vacuum energy eigenvalues in the following form

$$E_j \doteq \sqrt{p_j^2 + m_j^2} = p_j + \frac{m_j^2}{2p_j} + \mathcal{O}\left(\frac{m_j^4}{p_j^3}\right), \tag{3.5}$$

where $p_j \doteq |\vec{p}_j|$ is the modulus of the momentum of the mass eigenstate $|\nu_j\rangle$ and m_j is its mass value, and where the full formula can be expanded in powers and truncated after the second term since the task is to describe very relativistic neutrinos, i.e. particles such that $m_j/p_j \ll 1$.
At this point one can simplify the picture by assuming all mass eigenstates sharing the same value for the momentum modulus, i.e. $p_j = p\ \forall j \in \{1,2\}$ with p which can be taken to be the momentum modulus of a massless neutrino (therefore also equal to its energy) generated by the considered process: this is in principle wrong

since in general the energy and momentum of a particle produced in any process will both depend on its mass and both are uniquely determined by conservation laws, but from a more in-depth study ([68, 69]) one could show that at this order of approximation this choice would not change the flavor transition probabilities. Under these assumptions and discarding terms of order m_j^4/p_j^3 and greater, the 2-flavor vacuum energy operator written in mass $\hat{H}_{vac}^{(m)}$ basis will take the following form

$$\hat{H}_{vac}^{(m)} \doteq \begin{pmatrix} E_1 & 0 \\ 0 & E_2 \end{pmatrix} = \begin{pmatrix} p + \frac{m_1^2}{2p} & 0 \\ 0 & p + \frac{m_2^2}{2p} \end{pmatrix} = p\hat{\mathbb{I}} + \begin{pmatrix} \frac{m_1^2}{2p} & 0 \\ 0 & \frac{m_2^2}{2p} \end{pmatrix} \cong \begin{pmatrix} \frac{m_1^2}{2p} & 0 \\ 0 & \frac{m_2^2}{2p} \end{pmatrix}, \quad (3.6)$$

where, since Hamiltonian terms which are proportional to the identity operator $\hat{\mathbb{I}}$ only introduce global phases and are not relevant to the time evolution of the system or to the evaluation of observables, I have dropped the term $p\hat{\mathbb{I}}$.
Now, the previous operator can be recast in flavor basis by transforming it with the matrix $\hat{U}$ defined in Eq. (3.2)

$$\begin{aligned} \hat{H}_{vac}^{(f)} &= \hat{U}\hat{H}_{vac}^{(m)}\hat{U}^\dagger = \frac{1}{2p}\begin{pmatrix} c & s \\ -s & c \end{pmatrix}\begin{pmatrix} m_1^2 & 0 \\ 0 & m_2^2 \end{pmatrix}\begin{pmatrix} c & -s \\ s & c \end{pmatrix} = \\ &= \frac{1}{2p}\begin{pmatrix} c^2m_1^2 + s^2m_2^2 & cs(m_2^2 - m_1^2) \\ cs(m_2^2 - m_1^2) & s^2m_1^2 + c^2m_2^2 \end{pmatrix} \cong \frac{\delta m^2}{2p}\begin{pmatrix} s^2 & cs \\ cs & c^2 \end{pmatrix} \cong \\ &\cong \frac{\delta m^2}{4p}\begin{pmatrix} -\cos(2\theta_\nu) & \sin(2\theta_\nu) \\ \sin(2\theta_\nu) & \cos(2\theta_\nu) \end{pmatrix}, \end{aligned} \quad (3.7)$$

where for the sake of conciseness I used the definitions $c \doteq \cos(\theta_\nu)$, $s \doteq \sin(\theta_\nu)$, $\delta m^2 \doteq m_2^2 - m_1^2$ and like before I ignored terms proportional to the identity.
It is customary now to recall the Pauli matrices definitions

$$\hat{X} \equiv \hat{\sigma}_x \doteq \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \hat{Y} \equiv \hat{\sigma}_y \doteq \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \hat{Z} \equiv \hat{\sigma}_z \doteq \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (3.8)$$

With these operators, Eq. (3.7) can be easily recast in the following simple form

$$\hat{H}_{vac}^{(f)} = \Delta\vec{b} \cdot \vec{\hat{\sigma}}, \quad (3.9)$$

where I defined the vacuum potential $\Delta \doteq \frac{\delta m^2}{4p}$, a three-dimensional vector $\vec{b} = (\sin(2\theta_\nu), 0, -\cos(2\theta_\nu))$, lying in the x-z plane, and the Pauli vector $\vec{\hat{\sigma}} \doteq (\hat{\sigma}_x, \hat{\sigma}_y, \hat{\sigma}_z)$, already introduced in Eq. (1.13).
In this form the Hamiltonian is equivalent to that of a 2-level quantum system in an external static magnetic field: one can identify $\vec{b}$ as the direction of the magnetic field and interpret the flavor as a spin, which will evolve on the Bloch sphere

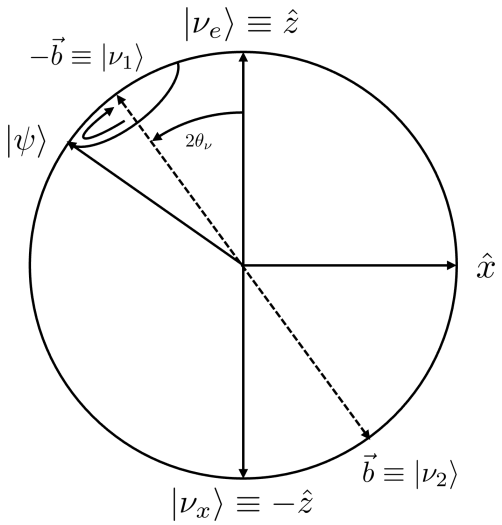


Figure 3.1: x-z plane cut of the Bloch sphere for the flavor state of a neutrino. The generic pure state $|\psi\rangle$ evolves under the action of the vacuum term of the Hamiltonian by precessing with Larmor frequency $\omega = \frac{\delta m^2}{2p}$ around a unit vector $\vec{b}$ oriented as the mass eigenstate $|\nu_2\rangle$, at an angle $2\theta_\nu$ with respect to the $-\hat{z}$ direction.

by undergoing a precession around $\vec{b}$ (which points at an angle $2\theta_\nu$ with respect to the $-\hat{z}$ axis and on the x-z plane) with Larmor frequency $\omega = 2\Delta$ (see Fig. 3.1).

This derivation was performed considering just one neutrino, but since the vacuum evolution of any particle is completely independent of the others, it can be easily extended to the case of a system of N neutrinos simply by summing the different contributions: the complete vacuum energy operator for a system of N neutrinos is a one-body operator with different terms acting each on a different particle and on an N-dimensional Hilbert space.

– Evolution in matter, H_{mat}

Despite neutrino interaction with matter being very feeble, being mediated by the weak force, it can be particularly relevant when considering collective interactions, i.e. interactions involving all particles and adding up in a coherent fashion: this holds when considering elastic scattering with leptons. The main contribution to this process arises from coherent forward scattering, which preserves particle momentum and can have two different origins: it may be due to charged-current interaction, mediated by boson $W^\pm$, or to neutral-current interaction, mediated by boson Z^0.

The neutral-current interaction being identical for all neutrino flavors means that it affects the Hamiltonian with a term which is proportional to the identity and as such can be neglected, as already discussed. On the other hand, charged-

current interaction differs depending on the considered neutrino flavor: this is true because neutrinos of a given flavor interact with the leptons of the same flavor, but the environment inside a core-collapse supernova is characterized by a great abundance of electrons with respect to the other leptons. The thermal energies in a collapsing supernova core are in fact of the order of tens of MeV, much lower than the hundreds and thousands needed for the production of mu and tau leptons. This introduces an imbalance between the behavior of the electronic neutrino with respect to the other flavors and is part of the reason why it's possible to effectively treat such particle as having only two different flavors in this specific environment. The interaction among neutrinos and charged particles has been deeply studied and is responsible for the well-known MSW effect [70–72],
The forward scattering effective potential V between an electron neutrino and an electron must be proportional to the Fermi constant G_F and to the electron density n_e: more specifically, the Hamiltonian $\hat{H}_{mat}$ describing the interaction between a neutrino and a lepton in the two flavor approximation can be written in the flavor basis in the following way

$$\hat{H}^{(f)}_{mat} = \sqrt{2}G_F n_e \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \doteq V \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \cong \frac{V}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \frac{V}{2}\hat{\sigma}_z, \tag{3.10}$$

where I defined $V \doteq \sqrt{2}G_F n_e$ and as usual terms proportional to the identity were neglected.
It is particularly interesting to notice that it's possible to express the sum of the vacuum and matter terms of the neutrino Hamiltonian in the same form as the vacuum term alone by conveniently redefining the mass difference and the mixing angle thanks to the introduction of a rescaling factor α. By recalling equation Eq. (3.7) it's in fact possible to recast such sum in the following form

$$\begin{aligned} \hat{H}^{(f)}_{vac} + \hat{H}^{(f)}_{mat} &= \frac{\delta m^2}{4p} \begin{pmatrix} -\cos(2\theta_\nu) & \sin(2\theta_\nu) \\ \sin(2\theta_\nu) & \cos(2\theta_\nu) \end{pmatrix} + \frac{V}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \\ &= \frac{\delta m^2}{4p} \begin{pmatrix} -(\cos(2\theta_\nu) - \frac{V}{2}\frac{4p}{\delta m^2}) & \sin(2\theta_\nu) \\ \sin(2\theta_\nu) & \cos(2\theta_\nu) - \frac{V}{2}\frac{4p}{\delta m^2} \end{pmatrix} = \\ &= \frac{\delta m^2}{4p}\alpha \begin{pmatrix} -\frac{1}{\alpha}\left(\cos(2\theta_\nu) - \frac{V}{2}\frac{4p}{\delta m^2}\right) & \frac{1}{\alpha}\sin(2\theta_\nu) \\ \frac{1}{\alpha}\sin(2\theta_\nu) & \frac{1}{\alpha}\left(\cos(2\theta_\nu) - \frac{V}{2}\frac{4p}{\delta m^2}\right) \end{pmatrix}. \end{aligned} \tag{3.11}$$

By choosing now the rescaling factor $\alpha = \sqrt{\sin^2(2\theta_\nu) + \left(\cos(2\theta_\nu) - \frac{V}{2}\frac{4p}{\delta m^2}\right)^2}$ it's possible to define the new matter parameters $\delta m'^2$ and θ'_ν through the following identities

$$\delta m'^2 \doteq \delta m^2 \alpha, \quad \cos(2\theta'_\nu) \doteq \frac{1}{\alpha}\left(\cos(2\theta_\nu) - \frac{V}{2}\frac{4p}{\delta m^2}\right), \quad \sin(2\theta'_\nu) \doteq \frac{1}{\alpha}\sin(2\theta_\nu). \tag{3.12}$$

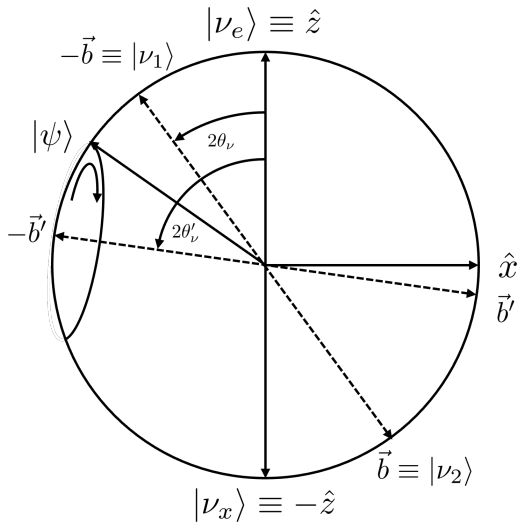


Figure 3.2: x-z plane cut of the Bloch sphere for the flavor state of a neutrino. The generic pure state $|\psi\rangle$ evolves under the joint action of the vacuum and matter terms of the Hamiltonian by preceeding with Larmor frequency $\omega' = \alpha\frac{\delta m^2}{2p}$ around a unit vector $\vec{b}'$ oriented at an angle $2\theta'_\nu$ with respect to the $-\hat{z}$ direction. Angle θ'_ν is identified by imposing relations in Eq. 3.12 and is always closer to the $\hat{z}$ direction than θ_ν.

By identifying the terms of Eq. (3.12) in Eq. (3.11), this last can be further simplified and explicitly expressed in the same form of Eq. (3.9)

$$\hat{H}^{(f)}_{vac} + \hat{H}^{(f)}_{mat} = \frac{\delta m'^2}{4p}\begin{pmatrix} -\cos(2\theta'_\nu) & \sin(2\theta'_\nu) \\ \sin(2\theta'_\nu) & \cos(2\theta'_\nu) \end{pmatrix} = \Delta'\vec{b}' \cdot \vec{\hat{\sigma}}, \tag{3.13}$$

where now $\Delta' \doteq \frac{\delta m'^2}{4p}$ and $\vec{b}' \doteq (\sin(2\theta'_\nu), 0, -\cos(2\theta'_\nu))$.
Recalling the analogy with magnetic systems, it's possible to say that the matter effect on the vacuum Hamiltonian is that of both shifting upwards the axis around which the flavor state of the considered neutrino processes, adding to $\vec{b}$ a contribution directed as $\hat{z}$ (with respect to the orientation in Fig. 3.1) and to modify the Larmor frequency of such precession to the new value $\omega' \doteq \alpha\omega$ (see Fig. 3.2).

Once again the previous derivation only considered one neutrino: the vacuum plus matter Hamiltonian for a system of many neutrinos will be a one-body operator with one term for each particle, each of which will have the form of Eq. 3.13. It is important to stress the fact that the terms Δ' and $\vec{b}'$ in Eq. 3.13 depend on the neutrino momentum p, so they will potentially be different for each considered particle.

– Evolution in neutrino dense environment, $H_{\nu\nu}$

Neutrinos interact weakly also among themselves via the neutral current, i.e. by exchanging Z_0 bosons. At the range of $10-100$ km from the center of the collapsing supernova the neutrino density is so high that also this interaction becomes relevant and even dominates over the other two contributions treated before. This term comes from the forward scattering among neutrinos and introduces the much relevant feature of non-linearity in the evolution, as well as giving rise to various phenomena like collective oscillations, synchronization, bipolar oscillations and spectral swaps [73, 74]: all of these effects are associated to a coherent evolution of the states of a great number of neutrinos.
A peculiarity of this term, compared with the matter forward scattering contribution, is that it has a non-diagonal structure, allowing for flavor mixing among neutrinos.
In order to treat this term it is common to assume that the energies into play are far smaller than the Z_0 boson mass: the value of the latter is in fact approximately 90 GeV, while we already stated that supernovae neutrinos have energies of the order of tens of MeV, which justifies the assumption. This allows for the use of the four-fermion contact interaction of Fermi's effective theory, according to which the neutrino-neutrino term $\hat{H}_{\nu\nu}$ in the Hamiltonian, considering now a system of N interacting particles, takes the following form [18, 75]

$$\hat{H}_{\nu\nu} = \frac{G_F n_\nu}{\sqrt{2}N} \sum_{i<j=1}^{N} \left(1 - \frac{\vec{p}_i \cdot \vec{p}_j}{\|\vec{p}_i\|\|\vec{p}_j\|}\right) \vec{\hat{\sigma}}_i \cdot \vec{\hat{\sigma}}_j, \tag{3.14}$$

where $\|\cdot\|$ is the Euclidean norm.
As expected, once again this term is proportional to the Fermi constant G_F and to the density of interacting particles, in this case the neutrino density $n_\nu = \frac{N}{V}$, where V is the volume of the system. Furthermore, in Eq. (3.14) $\vec{p}_i$ is the i-th neutrino momentum vector and $\vec{\hat{\sigma}}_i$ the Pauli vector acting on such particle

$$\vec{\hat{\sigma}}_i \doteq \left(\hat{X}_i, \hat{Y}_i, \hat{Z}_i\right), \tag{3.15}$$

$$\hat{\alpha}_i \doteq \bigotimes_{j=1}^{i-1} \hat{\mathbb{I}}_2 \otimes \hat{\alpha} \otimes \bigotimes_{j=i+1}^{N} \hat{\mathbb{I}}_2 \qquad \forall \alpha \in \{X, Y, Z\}, \tag{3.16}$$

where $\hat{\mathbb{I}}_2$ is the two-dimensional identity operator.
The operatorial form of this term corresponds to a spin-exchange interaction, coherently with the fact that under this interaction either the neutrinos exchange their own momenta or they do not. The angular factor J_{ij}, defined as the following

$$J_{ij} \doteq 1 - \frac{\vec{p}_i \cdot \vec{p}_j}{\|\vec{p}_i\|\|\vec{p}_j\|} = 1 - \cos(\phi_{ij}), \tag{3.17}$$

encodes the dependence of the interaction strength from the neutrino directions of propagation, where ϕ_{ij} is the angle among these directions: the more the neutrinos are counter-propagating, the more the interaction is intense, while when they move in parallel, the interaction becomes null.
It is important to note that this term describes an all-to-all interaction, which is in general exponentially difficult to treat classically and much better suited for a quantum computing approach. The all-to-all nature descends from both the choice of treating the neutrinos as plane waves and the approximation of contact effective interaction.
By defining the neutrino-neutrino interaction energy $\mu \doteq \sqrt{2}G_F n_\nu \doteq N\eta$ it is possible to rewrite Eq. (3.14) in a somewhat more elegant form

$$\hat{H}_{\nu\nu} = \frac{\mu}{2N} \sum_{i<j=1}^{N} J_{ij} \vec{\hat{\sigma}}_i \cdot \vec{\hat{\sigma}}_j. \tag{3.18}$$

Thanks to the global $\mathcal{SU}(2)$ invariance which characterizes this term I could remove from the notation the information about the basis used: the operatorial form of $\hat{H}_{\nu\nu}$ will in fact be the same regardless of the specific basis chosen, as long as the different bases are connected by an $\mathcal{SU}(2)$ rotation, which is indeed the case for the mass and flavor bases (see Eq. (3.2)).

– Full many-body Hamiltonian, H_N

Let's now combine the single terms just introduced into the full many-body neutrino Hamiltonian that was used for this work. As already stated, this is done by applying the single-particle term in Eq. (3.13) to each of the N particles in the system and summing to this the all-to-all interaction of Eq. (3.18). The result is the following expression

$$\hat{H}_N^{(f)} = \sum_{i=1}^{N} \Delta_i' \vec{b}_i' \cdot \vec{\hat{\sigma}}_i + \frac{\mu}{2N} \sum_{i<j=1}^{N} J_{ij} \vec{\hat{\sigma}}_i \cdot \vec{\hat{\sigma}}_j, \tag{3.19}$$

with the definitions of the various quantities given previously in this chapter.
More specifically, I decide to focus on a monochromatic beam of neutrinos propagating outwards from the supernova core in a narrow cone: this means that all the particles share the same momentum modulus (i.e. $p_i = p \; \forall i \in \{1, \ldots, N\}$), which implies all matter potentials Δ_i' and vectors $\vec{b}_i'$ also being identical, so that their i index can be dropped and Δ' collected in front of the first sum in Eq. (3.19). This, in turn, allows to express the energies in units of Δ by redefining the Hamiltonian

as $\hat{\tilde{H}}_N^{(f)} \doteq \frac{\hat{H}_N^{(f)}}{\Delta'}$

$$\hat{H}_N^{(f)} = \Delta' \sum_{i=1}^{N} \vec{b'} \cdot \vec{\tilde{\sigma}}_i + \frac{\mu}{2N} \sum_{i<j=1}^{N} J_{ij} \vec{\tilde{\sigma}}_i \cdot \vec{\tilde{\sigma}}_j, \tag{3.20}$$

$$\hat{H}_N^{(f)} \to \hat{\tilde{H}}_N^{(f)} \doteq \frac{\hat{H}_N^{(f)}}{\Delta'} = \sum_{i=1}^{N} \vec{b'} \cdot \vec{\tilde{\sigma}}_i + \gamma \sum_{i<j=1}^{N} J_{ij} \vec{\tilde{\sigma}}_i \cdot \vec{\tilde{\sigma}}_j, \tag{3.21}$$

where I defined $\gamma \doteq \frac{\mu}{2N\Delta'}$.
Since in this work I'll use a Hamiltonian written in these terms, I remove the tilde and single quote marks from the notation, to reduce its complexity

$$\hat{H}_N^{(f)} = \sum_{i=1}^{N} \vec{b} \cdot \vec{\tilde{\sigma}}_i + \gamma \sum_{i<j=1}^{N} J_{ij} \vec{\tilde{\sigma}}_i \cdot \vec{\tilde{\sigma}}_j. \tag{3.22}$$

As already done in previous works [61–63], I consider for the computations an angular distribution of the momenta consisting of a grid of equally spaced angles on a plane, i.e. I choose the angles ϕ_{ij} in Eq. (3.17) in accordance with the following relation

$$\phi_{ij} = \arccos(0.9) \frac{|i-j|}{N-1}. \tag{3.23}$$

This is a bit different from the popular choice of using the geometrical average for such angle (single-angle approximation): the reason is that, thanks to this simple not-isotropic assumption, one does not introduce any spurious symmetries into the system description.

3.1.2 Neutrino Hamiltonian properties and symmetries

The neutrino Hamiltonian defined in Eq. (3.22) possesses a lot of useful symmetries that can be conveniently exploited in different ways, for example to reduce the number of operations that a quantum computer has to execute when performing the simulation of the system. In the following chapter I will address the most relevant among these symmetries, which will be particularly important to my work.

– $\hat{H}_{\nu\nu}$ Global $\mathcal{SU}(2)$ invariance

As already stated, it's good to recall that the two-body term introduced in Eq. (3.18) is invariant under a global $\mathcal{SU}(2)$ transformation, meaning that the following iden-

tity holds

$$\begin{aligned} \hat{H}_{\nu\nu} &= \hat{U}(\alpha,\vec{n})\hat{H}_{\nu\nu}\hat{U}^{\dagger}(\alpha,\vec{n}) && \forall\ \hat{U}(\alpha,\vec{n}) \in \mathcal{SU}(2) \\ &= e^{-i\alpha\vec{n}\cdot\vec{\hat{\sigma}}_{tot}}\hat{H}_{\nu\nu}e^{i\alpha\vec{n}\cdot\vec{\hat{\sigma}}_{tot}} && \forall\ \alpha \in \mathbb{R},\ \vec{n} \in \mathbb{R}^3 \mid \|\vec{n}\| = 1, \end{aligned} \tag{3.24}$$

with $\vec{n} \doteq \{n_x, n_y, n_z\}$ a real tridimensional unit vector and the total Pauli vector operator $\vec{\hat{\sigma}}_{tot}$ of the system defined as the following

$$\vec{\hat{\sigma}}_{tot} \doteq \left\{\hat{X}_{tot}, \hat{Y}_{tot}, \hat{Z}_{tot}\right\} \doteq \left\{\sum_{i=1}^{N}\hat{X}_i, \sum_{i=1}^{N}\hat{Y}_i, \sum_{i=1}^{N}\hat{Z}_i\right\} = \sum_{i=1}^{N}\vec{\hat{\sigma}}_i. \tag{3.25}$$

A consequence of this is that such term will always have the same form in every basis, simplifying for example the task of moving between the mass and flavor representation.
It is straightforward to prove this property: just check that $\hat{H}_{\nu\nu}$ commutes with $\hat{G}_{\mathcal{SU}(2)}(\vec{n})$, the generators of the $\mathcal{SU}(2)$ group

$$\hat{G}_{\mathcal{SU}(2)}(\vec{n}) \doteq \vec{n}\cdot\vec{\hat{\sigma}}_{tot} = n_x\hat{X}_{tot} + n_y\hat{Y}_{tot} + n_z\hat{Z}_{tot}. \tag{3.26}$$

With this definition, and by recalling the $\hat{H}_{\nu\nu}$ definition given in Eq. (3.18), the aforementioned commutator satisfies the following relation

$$\left[\hat{H}_{\nu\nu}, \hat{G}_{\mathcal{SU}(2)}(\vec{n})\right] \propto \left[\sum_{j<k=1}^{N} J_{jk}\vec{\hat{\sigma}}_j\cdot\vec{\hat{\sigma}}_k, \vec{n}\cdot\vec{\hat{\sigma}}_{tot}\right], \tag{3.27}$$

where, exploiting Eq. (3.25), the RHS may be re-written in the following way

$$\begin{aligned} \left[\sum_{j<k=1}^{N} J_{jk}\vec{\hat{\sigma}}_j\cdot\vec{\hat{\sigma}}_k, \vec{n}\cdot\vec{\hat{\sigma}}_{tot}\right] &= \sum_{j<k=1}^{N} J_{jk}\sum_{i=1}^{N}\left[\vec{\hat{\sigma}}_j\cdot\vec{\hat{\sigma}}_k, \vec{n}\cdot\vec{\hat{\sigma}}_i\right] = \\ &= \sum_{j<k=1}^{N} J_{jk}\left(\sum_{i=j,k}\left[\vec{\hat{\sigma}}_j\cdot\vec{\hat{\sigma}}_k, \vec{n}\cdot\vec{\hat{\sigma}}_i\right] + \sum_{\substack{i=1\\ i\neq j,k}}^{N}\left[\vec{\hat{\sigma}}_j\cdot\vec{\hat{\sigma}}_k, \vec{n}\cdot\vec{\hat{\sigma}}_i\right]\right), \end{aligned} \tag{3.28}$$

having used the commutator linearity with respect to its arguments. All terms in Eq. (3.28) with $i \neq j, k$ vanish, since the commutator between operators acting on

different particles is always null. I can then continue with the following

$$\begin{aligned}\left[\sum_{j<k=1}^{N} J_{jk}\vec{\hat{\sigma}}_j\cdot\vec{\hat{\sigma}}_k, \vec{n}\cdot\vec{\hat{\sigma}}_{tot}\right] &= \sum_{j<k=1}^{N} J_{jk}\sum_{i=j,k}\left[\vec{\hat{\sigma}}_j\cdot\vec{\hat{\sigma}}_k, \vec{n}\cdot\vec{\hat{\sigma}}_i\right] = \\ &= \sum_{j<k=1}^{N} J_{jk}\sum_{i=j,k}\left[\hat{X}_j\hat{X}_k+\hat{Y}_j\hat{Y}_k+\hat{Z}_j\hat{Z}_k, n_x\hat{X}_i+n_y\hat{Y}_i+n_z\hat{Z}_i\right] = \\ &= \sum_{j<k=1}^{N} J_{jk}\sum_{i=j,k}\left[\hat{X}_j\hat{X}_k, n_x\hat{X}_i\right]+\left[\hat{X}_j\hat{X}_k, n_y\hat{Y}_i+n_z\hat{Z}_i\right]+ \\ &\quad +\left[\hat{Y}_j\hat{Y}_k, n_y\hat{Y}_i\right]+\left[\hat{Y}_j\hat{Y}_k, n_x\hat{X}_i+n_z\hat{Z}_i\right]+ \\ &\quad +\left[\hat{Z}_j\hat{Z}_k, n_z\hat{Z}_i\right]+\left[\hat{Z}_j\hat{Z}_k, n_x\hat{X}_i+n_y\hat{Y}_i\right].\end{aligned} \tag{3.29}$$

Given that any commutator trivially commutes with itself, all terms involving only one single kind of Pauli operator can be discarded, and by recalling the following commutation relations between Pauli operators (i here is the imaginary unit)

$$\left[\hat{X},\hat{Y}\right]=2i\hat{Z},\quad \left[\hat{Y},\hat{Z}\right]=2i\hat{X},\quad \left[\hat{Z},\hat{X}\right]=2i\hat{Y}, \tag{3.30}$$

Eq. (3.29) can be further reduced to the following

$$\begin{aligned}&\left[\sum_{j<k=1}^{N} J_{jk}\vec{\hat{\sigma}}_j\cdot\vec{\hat{\sigma}}_k, \vec{n}\cdot\vec{\hat{\sigma}}_{tot}\right] = \\ &\quad = \sum_{j<k=1}^{N} 2iJ_{jk}\Big(n_y\left(\hat{Z}_j\hat{X}_k+\hat{X}_j\hat{Z}_k\right)-n_z\left(\hat{Y}_j\hat{X}_k+\hat{X}_j\hat{Y}_k\right)+ \\ &\qquad -n_x\left(\hat{Z}_j\hat{Y}_k+\hat{Y}_j\hat{Z}_k\right)+n_z\left(\hat{X}_j\hat{Y}_k+\hat{Y}_j\hat{X}_k\right)+ \\ &\qquad +n_x\left(\hat{Y}_j\hat{Z}_k+\hat{Z}_j\hat{Y}_k\right)-n_y\left(\hat{X}_j\hat{Z}_k+\hat{Z}_j\hat{X}_k\right)\Big),\end{aligned} \tag{3.31}$$

where also here the letter i now stands for the imaginary unit.
Since all terms in the last equation cancel two-by-two, finally the expected result is recovered from Eq. (3.27)

$$\left[\hat{H}_{\nu\nu}, \hat{G}_{\mathcal{SU}(2)}(\vec{n})\right]=0. \tag{3.32}$$

– One-body and two-body terms commutation

Let's define $\hat{H}_{1B}$ the one-body term of the full many-body neutrino Hamiltonian (i.e. the first summation in Eq. (3.22), the contribution from vacuum and matter)

$$\hat{H}_{1B}^{(f)} \doteq \sum_{i=1}^{N} \vec{b} \cdot \vec{\hat{\sigma}}_i, \tag{3.33}$$

and $\hat{H}_{2B}$ the two-body term (i.e. the second summation, the contribution from interaction among neutrinos)

$$\hat{H}_{2B} \doteq \gamma \sum_{i<j=1}^{N} J_{ij} \vec{\hat{\sigma}}_i \cdot \vec{\hat{\sigma}}_j, \tag{3.34}$$

where I wrote $\hat{H}_{1B}$ in the flavor representation, $\hat{H}_{1B}^{(f)}$, while $\hat{H}_{2B}$ does not depend on the chosen basis thanks to the global $\mathcal{SU}(2)$ invariance.
I'll now show that these two terms commute one with the other, which is a very useful property since it allows to separate the evolution of the system under the two contributions (see Section 3.4.1). This will come in particularly handy when dealing with the estimation of observables, as will become clearer in the following chapters.
Let us start by noticing that by applying a unitary transformation $\hat{U}$ to two commuting operators $\hat{A}$ and $\hat{B}$, the two new operators $\hat{A}' \doteq \hat{U}\hat{A}\hat{U}^\dagger$ and $\hat{B}' \doteq \hat{U}\hat{B}\hat{U}^\dagger$ still commute. This is evident from Eq. (3.35), where setting $\left[\hat{A}, \hat{B}\right] = 0$ implies $\left[\hat{A}', \hat{B}'\right] = 0$.

$$\begin{aligned} \left[\hat{A}', \hat{B}'\right] &= \hat{A}'\hat{B}' - \hat{B}'\hat{A}' = \hat{U}\hat{A}\hat{U}^\dagger\hat{U}\hat{B}\hat{U}^\dagger - \hat{U}\hat{B}\hat{U}^\dagger\hat{U}\hat{A}\hat{U}^\dagger = \\ &= \hat{U}\hat{A}\hat{B}\hat{U}^\dagger - \hat{U}\hat{B}\hat{A}\hat{U}^\dagger = \hat{U}\left[\hat{A}, \hat{B}\right]\hat{U}^\dagger, \end{aligned} \tag{3.35}$$

where I used the unitarity property $\hat{U}\hat{U}^\dagger = \hat{U}^\dagger\hat{U} = \hat{\mathbb{I}}$.
This simplifies the problem, since I'm allowed to represent $\hat{H}_{1B}$ in the mass basis, where it is diagonal. Neglecting terms proportional to identity as they do not

influence neutrino flavor oscillation, I can write the following

$$\begin{aligned}\left[\hat{H}_{1B},\hat{H}_{2B}\right] &= \left[\hat{H}_{1B}^{(m)},\hat{H}_{2B}\right] = -\gamma\left[\sum_{k=1}^{N}\hat{Z}_k,\sum_{i<j=1}^{N}J_{ij}\vec{\hat{\sigma}}_i\cdot\vec{\hat{\sigma}}_j\right] = \\ &= -\gamma\sum_{i<j=1}^{N}J_{ij}\left[\sum_{k=1}^{N}\hat{Z}_k,\vec{\hat{\sigma}}_i\cdot\vec{\hat{\sigma}}_j\right] = \\ &= -\gamma\sum_{i<j=1}^{N}J_{ij}\left(\left[\hat{Z}_i+\hat{Z}_j,\vec{\hat{\sigma}}_i\cdot\vec{\hat{\sigma}}_j\right]+\sum_{\substack{k=1\\k\neq i,j}}^{N}\left[\hat{Z}_k,\vec{\hat{\sigma}}_i\cdot\vec{\hat{\sigma}}_j\right]\right),\end{aligned} \tag{3.36}$$

but all the terms in the last sum are always 0, since they are commutators among operators acting on different particles. Continuing Eq. (3.36) by using the definition of Pauli vector $\vec{\hat{\sigma}}_i$ given in Eqs. (3.15), (3.16)

$$\begin{aligned}\left[\hat{H}_{1B},\hat{H}_{2B}\right] &= -\gamma\sum_{i<j=1}^{N}J_{ij}\left[\hat{Z}_i+\hat{Z}_j,\vec{\hat{\sigma}}_i\cdot\vec{\hat{\sigma}}_j\right] = \\ &= -\gamma\sum_{i<j=1}^{N}J_{ij}\left[\hat{Z}_i+\hat{Z}_j,\hat{X}_i\hat{X}_j+\hat{Y}_i\hat{Y}_j+\hat{Z}_i\hat{Z}_j\right] = \\ &= -\gamma\sum_{i<j=1}^{N}J_{ij}\left(\left[\hat{Z}_i+\hat{Z}_j,\hat{X}_i\hat{X}_j+\hat{Y}_i\hat{Y}_j\right]+\left[\hat{Z}_i+\hat{Z}_j,\hat{Z}_i\hat{Z}_j\right]\right).\end{aligned} \tag{3.37}$$

Once again, the last commutator can be discarded: this time the reason is that any operator trivially commutes with itself.
It is also straightforward to show that the first commutator must vanish as well

$$\begin{aligned}&\left[\hat{Z}_i+\hat{Z}_j,\hat{X}_i\hat{X}_j+\hat{Y}_i\hat{Y}_j\right] = \\ &\qquad= \left[\hat{Z}_i,\hat{X}_i\hat{X}_j\right]+\left[\hat{Z}_i,\hat{Y}_i\hat{Y}_j\right]+\left[\hat{Z}_j,\hat{X}_i\hat{X}_j\right]+\left[\hat{Z}_j,\hat{Y}_i\hat{Y}_j\right] = \\ &\qquad= \left[\hat{Z}_i,\hat{X}_i\right]\hat{X}_j+\left[\hat{Z}_i,\hat{Y}_i\right]\hat{Y}_j+\left[\hat{Z}_j,\hat{X}_j\right]\hat{X}_i+\left[\hat{Z}_j,\hat{Y}_j\right]\hat{Y}_i = \\ &\qquad= 2i\left(\hat{Y}_i\hat{X}_j-\hat{X}_i\hat{Y}_j+\hat{X}_i\hat{Y}_j-\hat{Y}_i\hat{X}_j\right)=0,\end{aligned} \tag{3.38}$$

where I used the Pauli operators commutation relations given in Eq. (3.30).
With this result I concluded that all terms in Eq. (3.37) vanish, which in turn implies that indeed the one-body and two-body terms of the Hamiltonian commute

$$\left[\hat{H}_{1B},\hat{H}_{2B}\right]=0. \tag{3.39}$$

– $\hat{H}_{\nu\nu}$ $\mathcal{U}(1)$ invariance

The neutrino-neutrino interaction $\hat{H}_{\nu\nu}$ in Eq. (3.18) is also $\mathcal{U}(1)$ invariant under the action of transformations generated by the total Pauli Z operator $\hat{Z}_{tot} \doteq \sum_{i=1}^{N} \hat{Z}_i$, meaning that

$$\hat{H}_{\nu\nu} = e^{-i\alpha\hat{Z}_{tot}} \hat{H}_{\nu\nu} e^{i\alpha\hat{Z}_{tot}} \qquad \forall \alpha \in \mathbb{R}. \tag{3.40}$$

This is simply evident from the already discussed $\hat{H}_{\nu\nu}$ $\mathcal{SU}(2)$ invariance: by setting $\vec{n} = (0, 0, 1)$ in fact, Eq. (3.24) reduces exactly to Eq. (3.40).
This is also a property of the computational basis states, which is the usual basis used to represent the state of a set of N qubits, defined as the following set

$$\left\{ \bigotimes_{i=1}^{N} |x_i\rangle \;\middle|\; x_i \in \{0,1\} \;\forall\; i \in \{1,\ldots,N\} \right\} \quad \text{with } |0\rangle \doteq \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ and } |1\rangle \doteq \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{3.41}$$

Together, these two facts imply that, if the system is initially in a computational basis state and is evolved in complex time under the action of $\hat{H}_{\nu\nu}$, at any moment during evolution the full density matrix of the system will always remain invariant under the said symmetry. Since symmetries of the global state are inherited by its reduced states under the corresponding local action of the symmetry group, in such case it's always possible to represent at any time τ the two-body reduced density matrix $\hat{\rho}_{ij}$ in the following form

$$\begin{aligned} \hat{\rho}_{ij}(\tau) &\doteq \mathrm{Tr}_{k\notin\{i,j\}}\left[\hat{\rho}(\tau)\right] \\ &= \frac{1}{4}\Bigg(\hat{\mathbb{I}}_4 + \hat{Z}_i \mathrm{Tr}[\hat{Z}_i\hat{\rho}_{ij}(\tau)] + \hat{Z}_j \mathrm{Tr}[\hat{Z}_j\hat{\rho}_{ij}(\tau)] \\ &\qquad + \hat{Z}_i\hat{Z}_j \mathrm{Tr}[\hat{Z}_i\hat{Z}_j\hat{\rho}_{ij}(\tau)] \\ &\qquad + (\hat{X}_i\hat{X}_j + \hat{Y}_i\hat{Y}_j)\mathrm{Tr}[\hat{X}_i\hat{X}_j\hat{\rho}_{ij}(\tau)] \\ &\qquad + (\hat{X}_i\hat{Y}_j - \hat{Y}_i\hat{X}_j)\mathrm{Tr}[\hat{X}_i\hat{Y}_j\hat{\rho}_{ij}(\tau)] \Bigg), \end{aligned} \tag{3.42}$$

where $\hat{\rho}(\tau)$ is the full system density matrix at complex time $\tau \in \mathbb{C}$, defined as the following

$$\hat{\rho}(\tau) \doteq \frac{e^{-i\tau\hat{H}_{\nu\nu}}\hat{\rho}_0 e^{i\tau^*\hat{H}_{\nu\nu}}}{\mathrm{Tr}\left[e^{-i\tau\hat{H}_{\nu\nu}}\hat{\rho}_0 e^{i\tau^*\hat{H}_{\nu\nu}}\right]}, \tag{3.43}$$

with $\hat{\rho}_0$ the initial full density matrix at $\tau = 0$ that, as stated above, must correspond to a computational basis state (or at least must be itself $\mathcal{U}(1)$ invariant under the action of $\hat{Z}_{tot}$).
Eq. (3.42) is simply obtained by considering the two-body reduced density matrix $\hat{\rho}_{ij}(\tau)$ as a general two-body operator and expanding it on the Pauli operatorial

basis, neglecting in the decomposition all not $\mathcal{U}(1)$ invariant terms. I also exploited the fact that the expectation values of $\hat{X}_i\hat{X}_j$ and $\hat{X}_i\hat{Y}_j$ are equal to those of $\hat{Y}_i\hat{Y}_j$ and $-\hat{Y}_i\hat{X}_j$ respectively: this happens because in each couple the operators are connected by transformations belonging to the considered $\mathcal{U}(1)$ symmetry group. Furthermore, since operators $\hat{\mathbb{I}}_4$, $\hat{Z}_i$, $\hat{Z}_j$ and $\hat{Z}_i\hat{Z}_j$ all commute among themselves, Eq. (3.42) implies that only three independent measures on the system are needed to fully reconstruct the reduced two-body reduced density matrix. This can be further simplified when the evolution is performed in fully imaginary (real) time: in this case the off-diagonal term of the evolved two-body reduced density matrix will necessarily be completely real (imaginary), which in turn means that the expectation value of $\hat{X}_i\hat{Y}_j$ ($\hat{X}_i\hat{X}_j$) must be zero and therefore that two independent measurements are enough for determining completely the reduced density matrix. More explicitly, the two operators to be measured can be taken to be $\hat{X}_i\hat{X}_j$ ($\hat{X}_i\hat{Y}_j$) and $\hat{Z}_i\hat{Z}_j$, from which all the non-zero terms in Eq. (3.42) can be deduced.
This will be central for the estimation of two-body observables on thermal states, described in Section 3.4.2

– $\hat{H}_{\nu\nu}$ conservation of particle number

Let's consider the particle number operator $\hat{N}$, which can be defined as the following

$$\hat{N} \doteq \sum_{i=1}^{N} \frac{\hat{\mathbb{I}} - \hat{Z}_i}{2} = \frac{1}{2}(N\hat{\mathbb{I}} + \hat{Z}_{tot}). \tag{3.44}$$

Pay attention to the fact that $\hat{N}$ is the particle number operator, while N is the number of neutrinos in the system, which is not connected to the eigenvalues of the former operator: the name "particle number operator" is misleading in this context, since it is borrowed from a framework where a qubit in state $|1\rangle$ is interpreted as if a particle was present at a specific site, opposite to the state $|0\rangle$ that would represent the absence of such particle. In our context instead, the number of neutrinos is fixed for every initial state and it's equal to the number of considered qubits: operator $\hat{N}$ in this case measures the number of neutrinos in the non-electronic flavor $|\nu_x\rangle$, represented by a qubit in state $|1\rangle$.
From its definition it is clear that the $\hat{N}$ operator is diagonal in the computational basis and such that the eigenvalue corresponding to each basis state is its Hamming weight, i.e. the number of single-qubit states $|1\rangle$ in its representation according to Eq. (3.41).
It is very easy to realize that $\hat{N}$ must commute with $\hat{H}_{\nu\nu}$: writing down explicitly the commutator in fact, it reads

$$\left[\hat{H}_{\nu\nu}, \hat{N}\right] = \frac{1}{2}\left(N\left[\hat{H}_{\nu\nu}, \hat{\mathbb{I}}\right] + \left[\hat{H}_{\nu\nu}, \hat{Z}_{tot}\right]\right) \tag{3.45}$$

but since every operator always commutes with the identity $\hat{\mathbb{I}}$ and it was just discussed in the previous sections that $\left[\hat{H}_{\nu\nu}, \hat{Z}_{tot}\right] = 0$, necessarily the desired result follows

$$\left[\hat{H}_{\nu\nu}, \hat{N}\right] = 0. \tag{3.46}$$

This means that $\hat{H}_{\nu\nu}$ is a number-conserving Hamiltonian, which will have many interesting implications in what will be presented in the following sections. Clearly, this also means that if the initial state of the system has a well-defined Hamming weight, this will be conserved during the evolution.

– Invariance under particle ordering inversion

With the parameter choices made at the end of previous Section (see Eqs. (3.22) and (3.23)), it can be shown that both the one-body ($\hat{H}_{1B}$ defined in Eq. (3.33)) and the two-body ($\hat{H}_{2B}$ defined in Eq. (3.34)) terms must be invariant under the order reversal of particles, i.e. that the following relations hold:

$$\sum_{i=1}^{N} \vec{b} \cdot \vec{\hat{\sigma}}_i = \sum_{i=1}^{N} \vec{b} \cdot \vec{\hat{\sigma}}_{N-i+1}, \tag{3.47}$$

$$\sum_{i<j=1}^{N} J_{ij} \vec{\hat{\sigma}}_i \cdot \vec{\hat{\sigma}}_j = \sum_{i<j=1}^{N} J_{ij} \vec{\hat{\sigma}}_{N-i+1} \cdot \vec{\hat{\sigma}}_{N-j+1}, \tag{3.48}$$

where the first equation is for $\hat{H}_{1B}$ and the second for $\hat{H}_{2B}$.
It must be noted that this invariance is not intrinsic to the general neutrino interaction in Eq. (3.22), rather it is due to having chosen an angle distribution which is itself invariant for the reversal of the particle order (check Eq. (3.23)): such choice inserts this spurious symmetry in the problem, allowing on one hand for some simplification of the observable measurement process (as it is presented in Section 3.4.2), while still not generating relevant modifications to the overall system physics.
The proof for the first identity in Eq. (3.47) is pretty trivial: since the coupling with the external field is the same for all particles, any permutation of these ones won't have any effect. It is enough to change the summation index in the RHS to $i' \doteq N - i + 1$ to recover $\hat{H}_{1B}$.
In order to show the validity of Eq. (3.48), one starts again by redefining in the RHS the summation indexes to $i' \doteq N - j + 1$ and $j' \doteq N - i + 1$ (notice that i' is defined in terms of j and j' in terms of i) and then continues by exploiting the facts that operators acting on different particles commute and that the parameter J_{ij} is proportional to $|i - j|$ (and therefore also symmetric under the exchange

$i \leftrightarrow j$). More explicitly, the steps are as follows:

$$\begin{aligned}\sum_{i<j=1}^{N} J_{ij}\vec{\hat{\sigma}}_{N-i+1}\cdot\vec{\hat{\sigma}}_{N-j+1} &= \sum_{i'<j'=1}^{N} J_{(N-j'+1)(N-i'+1)}\vec{\hat{\sigma}}_{j'}\cdot\vec{\hat{\sigma}}_{i'} = \\ &= \sum_{i'<j'=1}^{N} J_{(N-i'+1)(N-j'+1)}\vec{\hat{\sigma}}_{i'}\cdot\vec{\hat{\sigma}}_{j'} = \\ &= \sum_{i'<j'=1}^{N} J_{i'j'}\vec{\hat{\sigma}}_{i'}\cdot\vec{\hat{\sigma}}_{j'},\end{aligned} \tag{3.49}$$

and result in Eq. (3.48) simply follows by arbitrarily renaming $\{i', j'\}$ as $\{i, j\}$. This result implies that if two initial states differ only for the reversal of the order of the particle, this property will be maintained also during the flavor evolution: one can then rely on such property to reduce the number of needed measurements to be performed on the system in order to extract specific information, as we will see in the context of estimating two-body observables in Section 3.4.2

– $\hat{H}_{\nu\nu}$ global Pauli-X invariance

The neutrino-neutrino interaction term $\hat{H}_{\nu\nu}$ in Eq. (3.18) possesses yet another useful symmetry: it is invariant under the transformation which flips all the neutrino flavor states, this transformation corresponding to the Pauli-X operator acting on all the particles, $\hat{U}_X \doteq \prod_{i=1}^{N} \hat{X}_i = \hat{U}_X^\dagger$

$$\hat{H}_{\nu\nu} = \hat{U}_X \hat{H}_{\nu\nu} \hat{U}_X. \tag{3.50}$$

This result comes straightforwardly when considering the action of an $\hat{X}$ transformation on each Pauli operator

$$\hat{X}\hat{X}\hat{X} = \hat{X}, \qquad \hat{X}\hat{Y}\hat{X} = -\hat{Y}, \qquad \hat{X}\hat{Z}\hat{X} = -\hat{Z}. \tag{3.51}$$

Expanding the RHS of Eq. (3.50) by using the definitions of $\hat{U}_X$ and $\hat{H}_{\nu\nu}$ and exploiting identities in Eq. (3.51), the desired result is soon obtained

$$\begin{aligned}\hat{U}_X \hat{H}_{\nu\nu} \hat{U}_X &= \left(\prod_{i=1}^{N} \hat{X}_i\right)\left(\frac{\mu}{2N}\sum_{j<k=1}^{N} J_{jk}\vec{\hat{\sigma}}_j\vec{\hat{\sigma}}_k\right)\left(\prod_{l=1}^{N} \hat{X}_l\right) = \\ &= \frac{\mu}{2N}\sum_{j<k=1}^{N} J_{jk}\hat{X}_j\hat{X}_k\left(\hat{X}_j\hat{X}_k + \hat{Y}_j\hat{Y}_k + \hat{Z}_j\hat{Z}_k\right)\hat{X}_j\hat{X}_k = \\ &= \frac{\mu}{2N}\sum_{j<k=1}^{N} J_{jk}\left(\hat{X}_j\hat{X}_k + \hat{Y}_j\hat{Y}_k + \hat{Z}_j\hat{Z}_k\right) = \hat{H}_{\nu\nu}.\end{aligned} \tag{3.52}$$

Like the previous result, also this symmetry helps in reducing the number of needed measurements for the estimation of expectation values of observables (see Section 3.4.2).

3.2 Finite temperature simulation and neutrino thermalization

There is general consensus in considering the general neutrino flavor Hamiltonian described in Section 3.1.1 (see Eq. (3.19)) as being non-integrable, where this statement has several implications about its spectrum and the expectation value of local observables.
The topic of Hamiltonian integrability is rather ample and complex (see for example [76, 77]) and in general many different non-equivalent definitions may be found in literature for such a property: here, by not-integrable, I mean that the differences Δ between sequential eigenstate energies in the spectrum do not follow a Poisson distribution $e^{-\Delta}$, rather they show repulsion between states (see Fig. 3.3).
This behavior is strongly connected with the symmetry properties of the system, which may contribute to increasing the number of degenerate states in the spectrum: Ref. [57] shows how the neutrino-neutrino interaction term of Eq. (3.18) indeed possesses a non-integrable character in the most general formulation, but this property is soon lost when a symmetric momenta distribution is considered rather than a more random one.
Non-integrable systems are also known to thermalize for the vast majority of possible initial states (possible exceptions are treated in [78]), meaning that the long-time average (i.e. the time average performed after a characteristic relaxation time) of the expectation value of few-body observables becomes insensitive to the specific initial state and can be estimated by considering the appropriate thermal ensemble (the interested reader may look for "Eigenstate Thermalization Hypothesis", ETH [58]).
Given these premises, I decide to describe the system as being in thermal equilibrium, using a density matrix $\hat{\rho}$ computed from the canonical ensemble, which therefore can be written in the following form

$$\hat{\rho}(\beta) = \frac{e^{-\beta \hat{H}}}{\mathcal{Z}}, \tag{3.53}$$

where $\beta \in \mathbb{R}$ is the generalized inverse temperature (which specific value can be chosen by fixing an observable expectation value, for example the system energy), $\hat{H}$ the Hamiltonian describing the system (see Eq. (3.22)) and $\mathcal{Z}$ the canonical partition function, defined as the following

$$\mathcal{Z} \doteq \mathrm{Tr}\left[\hat{\rho}\right]. \tag{3.54}$$

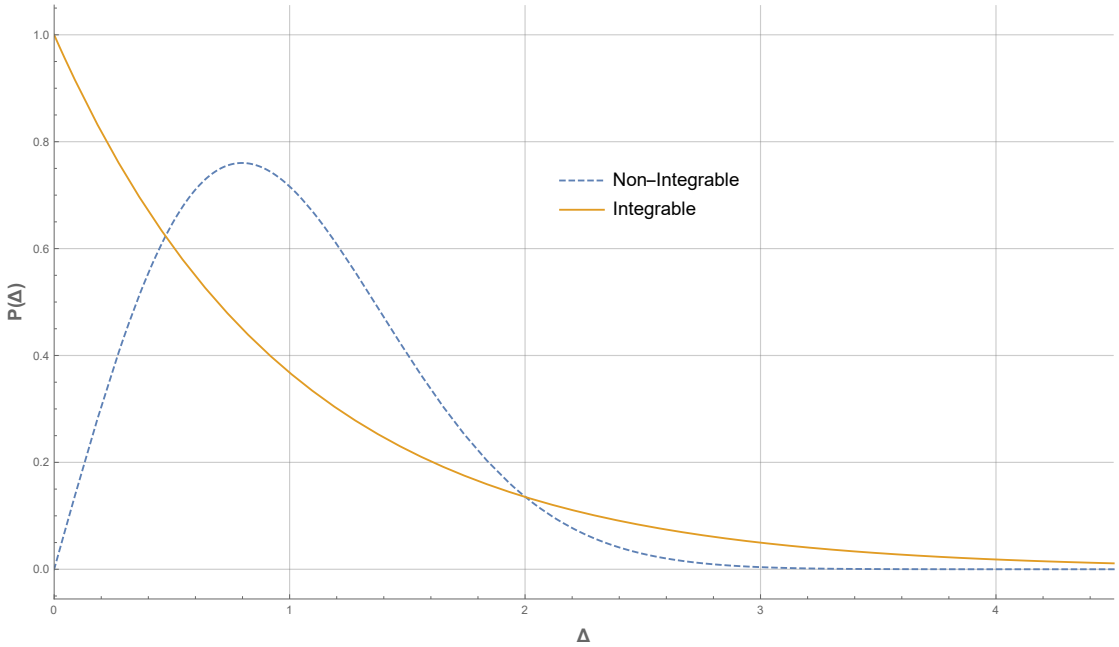


Figure 3.3: Comparison between the typical distribution of Δ between integrable (solid line) and non-integrable (dashed line) systems, with Δ being the difference between sequential energy eigenvalues. In integrable systems Δ is Poisson distributed, meaning that its probability is $P_{\mathrm{I}}(\Delta) = e^{-\Delta}$, while non-integrable systems show repulsion between consecutive energy eigenvalues, making the distribution more similar to $P_{\mathrm{NI}}(\Delta) = \frac{\pi}{2}\Delta e^{-\frac{\pi\Delta^2}{4}}$.

Here and in what follows we choose units such that $\hat{H}$ is dimensionless, which implies that also the inverse temperature β is.

3.2.1 Thermal expectation values

As usual in quantum mechanics, the expectation value $\langle\hat{O}\rangle_{\hat{\rho}}$ of an observable $\hat{O}$ computed on a system being in a state described by the density matrix $\hat{\rho}$ is obtained by taking the trace of the product of the two operators

$$\langle\hat{O}\rangle_{\hat{\rho}} \doteq \mathrm{Tr}\left[\hat{O}\hat{\rho}\right]. \tag{3.55}$$

If the system is divided into two distinct parts A and B, so that its Hilbert space $\mathcal{H}$ decomposes as the tensor product of the two correspondent subspaces, $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$, and the observable $\hat{O}$ only acts on one of such subsystems, say A, then Eq. (3.55) can be simplified by using the reduced density matrix $\hat{\rho}_A$ in place of $\hat{\rho}$, where $\hat{\rho}_A$ is obtained by taking the partial trace of $\hat{\rho}$ over subsystem B.
Eq. (3.55) clearly also holds when the system is in a thermal state, in which

case the density matrix takes the form of Eq. (3.53) and a dependence on the inverse temperature β is expected. By calling $\hat{\rho}_\beta$ such density matrix and $\langle \hat{O} \rangle_\beta$ the correspondent expectation value of operator $\hat{O}$, this latter will take the following form

$$\begin{aligned} \langle \hat{O} \rangle_\beta &= \frac{1}{\mathcal{Z}} \operatorname{Tr}\left[\hat{O} e^{-\beta \hat{H}}\right] = \frac{1}{\mathcal{Z}} \operatorname{Tr}\left[e^{-\frac{\beta}{2}\hat{H}} \hat{O} e^{-\frac{\beta}{2}\hat{H}}\right] = \\ &= \frac{1}{\mathcal{Z}} \sum_{j=1}^{d} \langle j | \, e^{-\frac{\beta}{2}\hat{H}} \hat{O} e^{-\frac{\beta}{2}\hat{H}} \, | j \rangle \,, \end{aligned} \tag{3.56}$$

where I exploited the cyclic property of the trace and introduced an arbitrary orthonormal basis $\{|j\rangle \mid j \in \{1, \dots, d\}\}$, d being the dimension of the system Hilbert space. In this basis, the canonical partition function $\mathcal{Z}$ defined in Eq. (3.54) can be expressed as follows

$$\mathcal{Z} = \sum_{j=1}^{d} \langle j | \, e^{-\beta \hat{H}} \, | j \rangle \,. \tag{3.57}$$

3.2.2 Imaginary-time evolution connection with thermalization

Eq. (3.56) at the end of last section seems to suggest a procedure for evaluating observable thermal expectation values: if it would be possible to efficiently implement in a quantum computer the operator $e^{-\beta \hat{H}}$, one could in principle prepare the states $|j\rangle$, apply $e^{-\frac{\beta}{2}\hat{H}}$ to them and then measure the observable $\hat{O}$, reconstructing the thermal expectation by computing $\mathcal{Z}$ and combining all the results.
There is a major problem with this: since β is a real number and $\hat{H}$ is Hermitian, it follows that the operator $e^{-\frac{\beta}{2}\hat{H}}$ is not unitary

$$e^{-\frac{\beta}{2}\hat{H}} \left(e^{-\frac{\beta}{2}\hat{H}}\right)^\dagger = e^{-\frac{\beta}{2}\hat{H}} e^{-\frac{\beta}{2}\hat{H}} = e^{-\beta \hat{H}} \neq \hat{\mathbb{I}}, \tag{3.58}$$

which means that it will modify the norm of the states onto which it is applied and that cannot be decomposed as a product of unitary operators.
Operators like this one are called Imaginary-Time Evolution (ITE) operators, since they clearly have the form of a time evolution (TE) operator with the time parameter being imaginary rather than generically complex

$$\textbf{TE} \quad e^{-i\mathfrak{t}\hat{H}} \mid \mathfrak{t} \in \mathbb{C},\ \hat{H} = \hat{H}^\dagger \quad \longrightarrow \quad e^{-\beta \hat{H}} \mid \beta \in \mathbb{R},\ \hat{H} = \hat{H}^\dagger \quad \textbf{ITE}. \tag{3.59}$$

Usually, operations are realized on actual hardware via switching on a controlled potential between the involved qubits and by letting them evolve for a specific time under such interaction: this exactly corresponds to a real-time evolution,

though, and as such, cannot allow by itself for the implementation of non-unitary operators, which are intrinsically more complex to be dealt with.
There are several alternatives which can lead to an effective implementation of non-unitary operators on a quantum machine, e.g. by exploiting additional qubits known as *ancillas* or also performing measurements on the systems [34]. In the following sections I'll show how to use an algorithm known as Quantum Imaginary-Time Evolution (QITE) to estimate the value of $\langle \hat{O} \rangle_\beta$, which only requires to execute unitary operations.

3.3 Unitary quantum estimation of thermal observables

As anticipated in Section 3.2.2, computing finite-temperature expectation values of observables is closely related to an evolution in imaginary time of the system, this last being quite a sophisticated operation to perform on a quantum computer given its non-unitary nature.
Despite this, it is still possible to write the expression for the thermal expectation value of an observable $\hat{O}$, i.e. $\langle \hat{O} \rangle_\beta$ (see Eq. (3.56)), in a form which makes it explicit how it is enough to only act with unitary operators in order to get to the result: the core of the issue consists in interpreting the non-unitary objects generated by the action of the imaginary-time evolution on the states $|j\rangle$, as unit-norm states $|\phi_j\rangle_\beta$ generated by acting with a certain unitary operator $\hat{U}_{j,\beta}$ (yet to be determined) again on states $|j\rangle$, times a weight $\mathcal{N}_{j,\beta}$ taking care of the norm difference. More explicitly, the steps are as follows

$$\begin{aligned}
\langle \hat{O} \rangle_\beta &= \frac{1}{\mathcal{Z}} \sum_{j=1}^{d} \langle i | \, e^{-\frac{\beta}{2}\hat{H}} \hat{O} e^{-\frac{\beta}{2}\hat{H}} \, |j\rangle = \\
&= \frac{1}{\mathcal{Z}} \sum_{j=1}^{d} \mathcal{N}_{j,\beta}^2 \, \langle j | \, \frac{e^{-\frac{\beta}{2}\hat{H}}}{\mathcal{N}_{j,\beta}} \hat{O} \frac{e^{-\frac{\beta}{2}\hat{H}}}{\mathcal{N}_{j,\beta}} \, |j\rangle = \\
&= \sum_{j=1}^{d} \mathbb{P}_{j,\beta} \, \langle \phi_j |_\beta \, \hat{O} \, |\phi_j\rangle_\beta \, ,
\end{aligned} \tag{3.60}$$

where the following quantities were introduced

$$\mathcal{N}_{j,\beta} \doteq \left\| e^{-\frac{\beta}{2}\hat{H}} \, |j\rangle \right\| , \qquad \mathbb{P}_{j,\beta} \doteq \frac{\mathcal{N}_{j,\beta}^2}{\mathcal{Z}}, \qquad |\phi_j\rangle_\beta \doteq \hat{U}_{j,\beta} \, |j\rangle \doteq \frac{e^{-\frac{\beta}{2}\hat{H}} \, |j\rangle}{\mathcal{N}_{j,\beta}}, \tag{3.61}$$

the norm here being the L^2-norm, defined as $\||\psi\rangle\| \doteq \sqrt{\langle\psi|\psi\rangle}$. With these definitions, by recalling Eq. (3.57) one also realizes the following relation

$$\mathcal{Z} = \sum_{j=1}^{d} \mathcal{N}_{j,\beta}^2, \tag{3.62}$$

which makes evident that the quantity $\mathbb{P}_{j,\beta}$ can be interpreted as a probability.
If it is possible to determine and implement the unitary operator $\hat{U}_{j,\beta}$ and to estimate the probabilities $\mathbb{P}_{j,\beta}$, then Eq. (3.60) is a valid prescription about how to get thermal expectation values of observables on a quantum computer.
Quantum Imaginary-Time Evolution (QITE) algorithm [16] exactly describes how to obtain good estimates for these two quantities: the starting point consists in choosing a general parametrized form for $\hat{U}_{j,\beta}$, call it $\hat{U}_\beta(\vec{\theta})$, where $\vec{\theta} = \{\theta_1, \ldots, \theta_g\}$ is a set of g real parameters to be determined by equating the action of $\hat{U}_\beta(\vec{\theta})$ on the $|j\rangle$ state to that of the normalized imaginary-time evolution.
Considering an example that will prove useful later, imagine representing $\hat{U}_\beta(\vec{\theta})$ as a product of g unitary operators, each of which I write as the imaginary exponential of an Hermitian operator $\hat{G}_k$

$$\hat{U}_\beta(\vec{\theta}) = \prod_{k=1}^{g} e^{-i\frac{\beta}{2}\theta_k \hat{G}_k}, \tag{3.63}$$

where β is the dimensionless inverse temperature and i the imaginary unit.
Let's now apply Eq. (3.63) on the state $|j\rangle$ and expand the result $|\psi_j(\vec{\theta})\rangle_\beta$ in powers of β up to the first order

$$|\psi_j(\vec{\theta})\rangle_\beta \doteq \hat{U}_\beta(\vec{\theta})\,|j\rangle = |j\rangle - i\frac{\beta}{2}\sum_{k=1}^{g}\theta_k \hat{G}_k\,|j\rangle + O(\beta^2). \tag{3.64}$$

Compare now this with $|\phi_j\rangle_\beta$, i.e. with the action of the normalized imaginary-time evolution on the same state $|j\rangle$, always expanded up to the first order in β

$$|\phi_j\rangle_\beta \doteq \frac{e^{-\frac{\beta}{2}\hat{H}}\,|j\rangle}{\sqrt{\langle j|\,e^{-\beta\hat{H}}\,|j\rangle}} = |j\rangle - \frac{\beta}{2}\hat{H}\,|j\rangle + \frac{\beta}{2}\langle j|\,\hat{H}\,|j\rangle\,|j\rangle + O(\beta^2). \tag{3.65}$$

The correct values $\vec{\tilde{\theta}}_j$ for $\hat{U}_\beta(\vec{\theta})$ are chosen so that they minimize the norm of the difference (divided by $\frac{\beta}{2}$) between these two objects

$$\begin{aligned}\vec{\tilde{\theta}}_j &\doteq \arg\min_{\vec{\theta}} \left[\left\| \frac{|\psi_j(\vec{\theta})\rangle_\beta - |\phi_j\rangle_\beta}{\frac{\beta}{2}} \right\|\right] = \\ &= \arg\min_{\vec{\theta}} \left[f_j + \left(\vec{b}_j\right)_k \theta_k + \theta_k\,(\mathbf{S}_j)_{kl}\,\theta_l + O(\beta)\right],\end{aligned} \tag{3.66}$$

where summation over repeated symbols is implied and with the following definitions

$$f_j \doteq \langle j|\, \hat{H}^2\, |j\rangle - \langle j|\, \hat{H}\, |j\rangle^2\,, \quad \left(\vec{b}_j\right)_k \doteq i\, \langle j|\, \left[\hat{G}_k, \hat{H}\right]\, |j\rangle\,, \quad (\mathbf{S}_j)_{kl} \doteq \langle j|\, \hat{G}_k \hat{G}_l\, |j\rangle\,, \tag{3.67}$$

i being here the imaginary identity.
Neglecting $O(\beta)$ terms, Eq. (3.66) corresponds to solving the following system

$$(\mathbf{A}_j)_{kl} \left(\vec{\tilde{\theta}}_j\right)_l = -\left(\vec{b}_j\right)_k, \tag{3.68}$$

where again, summation over repeated symbols is implied and matrix $\mathbf{A}_j$ is defined as follows

$$(\mathbf{A}_j)_{kl} \doteq \langle j|\, \{\hat{G}_k, \hat{G}_l\}\, |j\rangle\,, \tag{3.69}$$

with $\{\hat{A}, \hat{B}\} \doteq \hat{A}\hat{B} + \hat{B}\hat{A}$ the anticommutators between two operators $\hat{A}$ and $\hat{B}$.
By solving Eq. (3.68), one finds up to $O(\beta)$ the parameters with which to build a good approximation for the unitary operator $\hat{U}_{j,\beta}$ in Eq. (3.61).
At this point, parameters $\mathcal{N}_{i,\beta}$ in Eq. (3.60) are easily estimated up to order $O(\beta)$ by measuring the energy of the system on the state before the evolution:

$$\mathcal{N}_{j,\beta} \doteq \left\| e^{-\frac{\beta}{2}\hat{H}}\, |j\rangle \right\| = \sqrt{\langle j|\, e^{-\beta\hat{H}}\, |j\rangle} = \sqrt{1 - \beta\, \langle j|\, \hat{H}\, |j\rangle + O(\beta^2)}. \tag{3.70}$$

This also allows for the estimation of $\mathcal{Z}$, obtained by summing the $\mathcal{N}_{j,\beta}$ estimates over all $j \in [1, d]$ and therefore one is finally able to estimate $\mathbb{P}_{j,\beta}$ by exploiting its definition in Eq. (3.61).
To increase the accuracy of these estimates, one could proceed similarly to the Trotter expansion and divide the evolution in smaller steps, implementing the product of unitary approximations of the individual shorter operators rather than the full evolution: this would reduce the error of the estimates by a factor equal to the number of steps taken

$$\hat{U}_\beta(\vec{\tilde{\theta}}_j) \to \prod_{m=1}^{M} \hat{U}_{\Delta\beta}(\vec{\tilde{\theta}}_{j,m}), \tag{3.71}$$

with $\Delta\beta \doteq \beta/M$. More details on this procedure are given in Section 3.4.4.

3.3.1 Minimally entangled typical thermal states (METTS)

Let us reconsider states $|\phi_j\rangle_\beta$ and their associated probabilities $\mathbb{P}_{j,\beta}$, defined in Eq. (3.61)

$$|\phi_j\rangle_\beta \doteq \frac{e^{-\frac{\beta}{2}\hat{H}}\, |j\rangle}{\sqrt{\langle j|\, e^{-\beta\hat{H}}\, |j\rangle}}, \qquad \mathbb{P}_{j,\beta} \doteq \frac{\langle j|\, e^{-\beta\hat{H}}\, |j\rangle}{\sum_{k=1}^{d} \langle k|\, e^{-\beta\hat{H}}\, |k\rangle} \tag{3.72}$$

These states are quite special when the basis $\{|j\rangle\}_{j\in\{1,\dots,d\}}$ is chosen so that each state belonging to it is a separable state, i.e. a classical product state (CPS), which are states that can be written as a Kronecker product of single particle states: for a system consisting of N particles, any of these states can therefore be written in the following way

$$|j\rangle = \bigotimes_{k=1}^{N} |j_k\rangle \,, \tag{3.73}$$

Where each $|j_k\rangle$ is a single particle state. A basis of this kind is the computational basis, (see Eq. (3.41)).
With this choice, one could think of a process allowing the sampling of states from $\{|\phi_j\rangle_\beta\}_{j\in[1,d]}$, each taken with probability $\mathbb{P}_{j,\beta}$. The states sampled in this way are called Minimally Entangled Typical Thermal States (METTS) [59, 60] and possess lots of nice properties:

- at high temperatures METTSs reduce to CPSs, which are easy to generate in quantum computers and to treat, given that they don't have entanglement,
- if a many-body system is divided into non-interacting parts, any METTS describing it will be separable as a product of METTSs for the different parts,
- at any temperature, this states should possess the minimum entanglement and entropy possible, compared to states generated by choosing any basis $\{|j\rangle\}$ different from Eq. (3.73),
- observables expectation values on them do not differ much from the thermal average, computed with Eq. (3.56),
- when the system shows long-range order, these states naturally break the Hamiltonian symmetry.

Ref. [60] discusses different alternatives to efficiently implement the sampling procedure, known as "pure state algorithm" which would allow for a reliable estimate of thermal expectation values on bigger system. Given that in this work I'll focus on a smaller system though, these techniques are not strictly needed (they become advantageous typically for systems with 8 or more particles) and I'll just expand the system thermal density matrix in the basis constituted by all the METTS, exploiting symmetries to reduce the number of needed measurements (see Section 3.4.2).

3.3.2 Number-conserving unitaries and Givens rotations

In Section 3.1.2 I showed how the neutrino-neutrino interaction term $\hat{H}_{\nu\nu}$ introduced in Eq. (3.18) conserves the Hamming weight of the initial state (when this is well defined), this meaning that $\hat{H}_{\nu\nu}$ is a number-conserving Hamiltonian (see Eq. (3.46)). This can be efficiently exploited to find a good parametrization of the unitaries $\{\hat{U}_{j,\beta}\}_{j\in\{1,\dots,d\}}$ approximating the normalized imaginary-time evolution of the system under $\hat{H}_{\nu\nu}$.
In the literature, it is known in fact that there is a set of operators forming a basis for all particle-conserving unitary operators; this basis is composed of controlled single-excitation gates [17], which are particle-conserving generic unitary transformations acting only in the subspace of the system spanned by two computational basis states (see Eq. (3.41)) sharing the same Hamming weight and having Hamming distance equal to 2, which means that they are represented by bitstrings differing in only two bits.
Single-excitation gates are Givens rotations acting on the two differing qubits, mixing only the components $|01\rangle$ and $|10\rangle$ of their state: considering for simplicity real gates, the general Givens rotation $\hat{G}$ of this kind acting on two qubits can be written in the following way

$$\begin{aligned}\hat{G}(\theta) &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= e^{-i\frac{\theta}{2}\left(\hat{X}\otimes\hat{Y}-\hat{Y}\otimes\hat{X}\right)},\end{aligned} \tag{3.74}$$

where parameter $\theta \in [0, 2\pi)$ regulates how much this operator mixes the two states. When the aim is to build a parametrized circuit for the generation of a state, it is also known [17] that the controls are not needed, since removing them allows for the exploration of a wider portion of the Hilbert space: when looking for a parametrized operator $\hat{U}_\beta(\vec{\theta})$ to approximate the unitaries $\{\hat{U}_{j,\beta}\}_{j\in\{1,\dots,d\}}$ in the QITE algorithm (see Section 3.3) it is therefore possible to choose it as the product of Givens rotations acting on all possible couples of different qubits in the system

$$\hat{U}_\beta(\vec{\theta}) = \prod_{j=1}^{N(N-1)/2} \hat{G}_j(\beta\theta_j) = \prod_{j=1}^{N(N-1)/2} e^{-i\frac{\beta\theta_j}{2}\hat{O}_j}, \tag{3.75}$$

where N is the number of neutrinos in the system, $\frac{N(N-1)}{2}$ the number of different couples of neutrinos and the index j addresses each couple. Introducing the indices k_j and l_j, addressing the specific qubits belonging to the j-th couple, operators $\hat{O}_j$

can be defined with the following identity

$$\hat{O}_j \doteq \hat{X}_{k_j}\hat{Y}_{l_j} - \hat{Y}_{k_j}\hat{X}_{l_j}. \tag{3.76}$$

It is important to notice that the neutrino-neutrino Hamiltonian term $\hat{H}_{\nu\nu}$ is completely real: this means that also the imaginary-time evolution generated by it will be completely real and this in turn implies that the real Givens rotations given in Eq. (3.74) are indeed sufficient to effectively approximate the unitaries $\{\hat{U}_{j,\beta}\}_{j\in\{1,\dots,d\}}$.
A nice consequence of implementing $\hat{U}_\beta(\vec{\theta})$ with Givens rotations is that the system is still $\mathcal{U}(1)$ invariant (conserves flavor number): this means that one can still rely on such property to mitigate possible hardware errors happening during the simulation. In particular, it also means that the two-body reduced density matrix of the system initially in a computational basis state is always in the form of Eq. (3.42), even when the evolution is implemented with Givens rotations.
In order to realize Givens rotations on a quantum machine, one can decompose them in terms of basic single-qubit operators and CNOTs: a possible realization is given by the following circuit

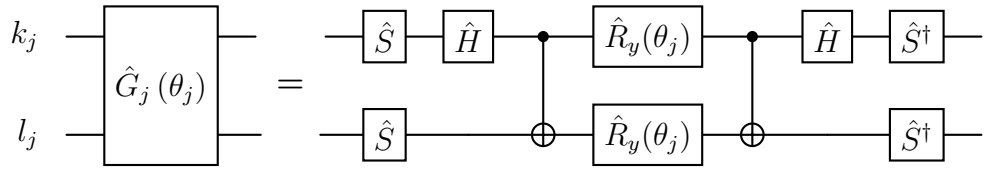


Figure 3.4: Quantum circuit implementing a real Givens $\hat{G}$ rotation of an angle θ_j on the j-th couple of qubits, more precisely qubit k_j and l_j. $\hat{S}$ is the phase gate, $\hat{H}$ the Hadamard gate and $\hat{R}_y(\theta)$ a rotation about the y-axis of an angle θ.

where $\hat{H}$ is here the Hadamard gate, $\hat{S}$ the phase gate and $\hat{R}_y$ the rotation generated by $\hat{Y}$:

$$\begin{aligned} \hat{H} &\doteq \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad \hat{S} \doteq \sqrt{\hat{Z}} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \\ \hat{R}_y(\theta) &\doteq e^{-i\frac{\theta}{2}\hat{Y}} = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}. \end{aligned} \tag{3.77}$$

The full parametrized circuit implementing the unitary operator $\hat{U}_\beta(\vec{\theta})$ for a system of four particles ($N = 4$) can then be represented in the following way

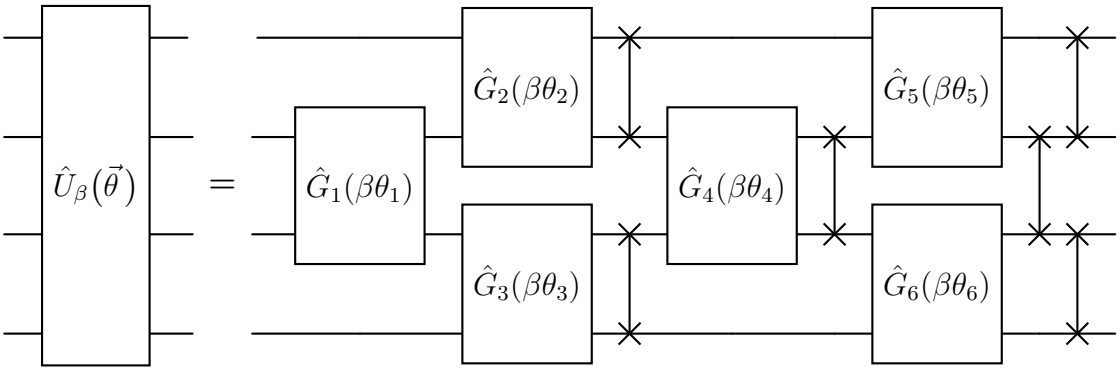


Figure 3.5: Quantum circuit implementing the QITE operator $\hat{U}_\beta(\vec{\theta})$ in Eq. (3.75). $\hat{G}_j(\theta_j)$ is a real Givens rotation of angle θ_j acting on the j-couple of qubits, implemented with the circuit in Fig. 3.4

where I recall the following SWAP operator introduced in Eq. (1.33), which effect is that of exchanging the states of the qubits onto which it acts

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} .$$

I point out that the SWAP gates are not needed in general, but only in case the topology of the quantum hardware chosen for the simulation only allows for two-qubit gates acting on neighboring qubits. Furthermore, the last three SWAP gates in the two rightmost layers of the circuit are only needed to restore the original associations between neutrinos and qubits: these three gates could always be avoided in principle by an adequate remap of the particle states to the physical qubits, regardless of the specific hardware connectivity topology.

3.4 QMETTS algorithm implementation

Having introduced in the previous sections all the topics relevant to the performed study, I'll explain here more thoroughly the details of the QMETTS algorithm employed to study the thermal expectation values of a system of neutrinos relying on the QITE algorithm and, more specifically, its Givens rotations implementation. Since the purpose of this work is that of presenting and qualitatively study the performances of the QMETTS algorithm, rather than gaining novel insights about a complex physical system, I chose to apply it on a very simple toy model consisting

of just four neutrinos, with their interaction modeled as described in Section 3.1.1

$$\hat{H}_4^{(f)} = \hat{H}_{1B}^{(f)} + \hat{H}_{2B} = \sum_{i=1}^{4} \vec{b} \cdot \vec{\hat{\sigma}}_i + \gamma \sum_{i<j=1}^{4} J_{ij} \vec{\hat{\sigma}}_i \cdot \vec{\hat{\sigma}}_j, \tag{3.78}$$

where $\hat{H}_{1B}^{(f)}$ and $\hat{H}_{2B}$ are the one-body and two-body terms respectively, defined in Eqs. (3.33) and (3.34), and with the following already discussed, choices for the parameters

$$\begin{aligned} J_{ij} =& 1 - \cos(\phi_{ij}), \;\; \phi_{ij} = \arccos(0.9)\frac{|i-j|}{3} \\ \vec{b} \doteq& \left(\sin(2\theta'_\nu), 0, -\cos(2\theta'_\nu)\right). \end{aligned} \tag{3.79}$$

The ratio γ between the strength of the two-body and one-body terms varies a lot along the evolution of a usual supernova neutrino: I will therefore consider a range of values for this parameter rather than fixing a single number, more specifically all the integer values comprised between 0 and 10.

3.4.1 Effective $\hat{H}_{2B}$-only evolution

Rather than considering Eq. (3.78) as is, I add some slight formal modifications to it, so to exploit some of the symmetries introduced in Section 3.1.2.
The first thing to consider is that, as it was demonstrated in Section 3.1.2, the one-body and two-body terms commute: this turns out to be very useful for separating the respective effects in the computation of thermal expectation values of observables. Imposing $\left[\hat{H}_{1B}^{(f)}, \hat{H}_{1B}\right] = 0$ in Eq. (3.56) for the evaluation of these expectation values, one gets the following relation

$$\begin{aligned} \langle \hat{O} \rangle_\beta &= \sum_{j=1}^{2^4} \frac{\langle j | \, e^{-\frac{\beta}{2}\hat{H}_4^{(f)}} \hat{O} e^{-\frac{\beta}{2}\hat{H}_4^{(f)}} \, | j \rangle}{\sum_{k=1}^{2^4} \langle k | \, e^{-\beta \hat{H}_4^{(f)}} \, | k \rangle} = \\ &= \sum_{j=1}^{16} \frac{\langle j | \, e^{-\frac{\beta}{2}\left(\hat{H}_{1B}^{(f)} + \hat{H}_{2B}\right)} \hat{O} e^{-\frac{\beta}{2}\left(\hat{H}_{1B}^{(f)} + \hat{H}_{2B}\right)} \, | j \rangle}{\sum_{k=1}^{16} \langle k | \, e^{-\beta\left(\hat{H}_{1B}^{(f)} + \hat{H}_{2B}\right)} \, | k \rangle} = \\ &= \sum_{j=1}^{16} \frac{\langle j | \, e^{-\frac{\beta}{2}\hat{H}_{1B}^{(f)}} e^{-\frac{\beta}{2}\hat{H}_{2B}} \hat{O} e^{-\frac{\beta}{2}\hat{H}_{1B}^{(f)}} e^{-\frac{\beta}{2}\hat{H}_{2B}} \, | j \rangle}{\sum_{k=1}^{16} \langle k | \, e^{-\beta\hat{H}_{1B}^{(f)}} e^{-\beta\hat{H}_{2B}} \, | k \rangle}. \end{aligned} \tag{3.80}$$

This form would be very convenient if one could chose the basis $\{|j\rangle\}$ so that it diagonalizes $\hat{H}_{1B}^{(f)}$: in this case the one-body evolution could be extracted from the bracket and would contribute to each term in the sum just as a weight, while the only actual evolution remaining would be due to the two-body term, i.e. to the

neutrino-neutrino interaction term.
Actually, I will proceed in a slightly different way: since $\hat{H}_{2B}$ is $\mathcal{SU}(2)$ invariant (see discussion in Section 3.1.2), one could also apply a global $\mathcal{SU}(2)$ transformation $\hat{U}$ to $\hat{H}_4^{(f)}$ and move to a reference where the diagonal basis for $\hat{H}_{1B}$ is known. More specifically, I choose the transformation so that the new representation of the one-body term is exactly the operator $\hat{Z}_{tot}$

$$\hat{H}_{1B}^{(f)} \to \hat{U}\hat{H}_{1B}^{(f)}\hat{U}^\dagger \doteq \hat{H}_{1B}^{(d)} = \hat{Z}_{tot} = \sum_{i=1}^{4} \hat{Z}_i. \tag{3.81}$$

This representation is analogous to the mass representation, and like this last is diagonal in the computational basis defined in Eq. (3.41), which we also saw possesses many useful properties relative to the evolution under $\hat{H}_{2B}$.
With this substitution and by choosing $\{|j\rangle\}$ as the computational basis, Eq. (3.80) becomes

$$\begin{aligned}
\langle\hat{O}\rangle_\beta &= \sum_{j=1}^{16} \frac{\langle j|\, e^{-\frac{\beta}{2}\hat{H}_{1B}^{(d)}} e^{-\frac{\beta}{2}\hat{H}_{2B}} \hat{O} e^{-\frac{\beta}{2}\hat{H}_{1B}^{(d)}} e^{-\frac{\beta}{2}\hat{H}_{2B}} \,|j\rangle}{\sum_{k=1}^{16} \langle k|\, e^{-\beta\hat{H}_{1B}^{(d)}} e^{-\beta\hat{H}_{2B}} \,|k\rangle} = \\
&= \sum_{j=1}^{16} \frac{e^{-\beta E_j^1} \langle j|\, e^{-\frac{\beta}{2}\hat{H}_{2B}} \hat{O} e^{-\frac{\beta}{2}\hat{H}_{2B}} \,|j\rangle}{\sum_{k=1}^{16} e^{-\beta E_k^1} \langle k|\, e^{-\beta\hat{H}_{2B}} \,|k\rangle} = \\
&= \sum_{j=1}^{16} \frac{e^{-\beta E_j^1} \langle j|\, e^{-\beta\hat{H}_{2B}} \,|j\rangle}{\sum_{k=1}^{16} e^{-\beta E_k^1} \langle k|\, e^{-\beta\hat{H}_{2B}} \,|k\rangle} \langle\phi_j|_\beta \hat{O} \,|\phi_j\rangle_\beta \approx \\
&\approx \sum_{j=1}^{16} \frac{e^{-\beta E_j^1} \left(1 - \beta \langle j|\, \hat{H}_{2B} \,|j\rangle\right)}{\sum_{k=1}^{16} e^{-\beta E_k^1} \left(1 - \beta \langle k|\, \hat{H}_{2B} \,|k\rangle\right)} \langle\psi_j(\vec{\tilde{\theta}}_j)|_\beta \hat{O} \,|\psi_j(\vec{\tilde{\theta}}_j)\rangle_\beta ,
\end{aligned} \tag{3.82}$$

with E_j^1 being $\hat{H}_{1B}^{(d)}$ eigenvalue on state $|j\rangle$, $|\phi_j\rangle_\beta$ being the normalized state evolved in imaginary time from state $|j\rangle$ under the action of $\hat{H}_{2B}$ alone, rather than all $\hat{H}_4^{(d)}$, and $|\psi_j(\vec{\tilde{\theta}}_j)\rangle_\beta$ is the correspondent QITE approximated state, where the angles $\vec{\tilde{\theta}}_j$ are found by solving Eqs. (3.67) with $\hat{H}_{2B}$ in place of $\hat{H}$

$$E_j^1 \doteq \langle j|\, \hat{H}_{1B}^{(d)} \,|j\rangle \qquad |\phi_j\rangle_\beta \doteq \frac{e^{-\frac{\beta}{2}\hat{H}_{2B}} \,|j\rangle}{\sqrt{\langle j|\, e^{-\beta\hat{H}_{2B}} \,|j\rangle}}. \tag{3.83}$$

As explained at the end of Section 3.3, in order to reduce the error of the QITE estimates for the computation of thermal expectation values, I separate the thermal evolution in several steps, starting from $\beta = 0$ and reaching step by step the desired

value. This implies that I'll have to generate multiple QITE states, one for every state in the computational basis at every step, and evaluate observable expectation values at each step. Thanks to Eq. (3.82) though, I can consider explicitly only the evolution due to $\hat{H}_{2B}$ and compute instead the contribution due to $\hat{H}_{1B}^{(d)}$ just once (for every basis state) at the beginning of the simulation, since this will remain constant throughout all the evolution.

3.4.2 Observable expectation values on QITE states

Now I move on to explain how to actually compute each term appearing in the sum of Eq. (3.82). Having already treated the estimation of the weights $\mathbb{P}_{j,\beta}$ defined as

$$\mathbb{P}_{j,\beta} = \frac{e^{-\frac{\beta}{2}E_j^1} \langle j| e^{-\beta \hat{H}_{2B}} |j\rangle}{\sum_{k=1}^{16} e^{-\frac{\beta}{2}E_k^1} \langle k| e^{-\beta \hat{H}_{2B}} |k\rangle} \tag{3.84}$$

in Section 3.3, here I'll focus on the evaluation of the terms $\langle \hat{O} \rangle_{j,\beta} \doteq \langle \phi_j|_\beta \hat{O} |\phi_j\rangle_\beta$, with β the inverse temperature of an arbitrary step of the total evolution and $\hat{O}$ either a one-body or two-body observable.

The procedure consists in generating all the approximating states $\{|\psi_j(\vec{\tilde{\theta}}_j)\rangle_\beta\}_{j\in\{1,\dots,16\}}$ with the QITE algorithm and then measuring on each of them all the strings $\vec{\hat{P}}$ of Pauli operators (including the identity) needed to be able to reconstruct every possible one-body and two-body reduced density matrices $\hat{\rho}_{j,\beta}^k$ and $\hat{\rho}_{j,\beta}^{kl}$ (indices k and l identify the specific particles included in the reduced density matrix). These matrices are then generated as solutions of the following convex optimization that uses the measurements

$$\hat{\rho}_{j,\beta}^k = \arg\min_{\hat{\rho}'} \left[\left\| \vec{a}_j^{\,k}(\hat{\rho}') - \vec{p}_{j,\beta}^{\,k} \right\| \right]. \tag{3.85}$$

Here, $\hat{\rho}_{j,\beta}^k$ represents either a one-body or two-body reduced density matrix, $\vec{p}_{j,\beta}^{\,k}$ being the vector of measurements on state $|\psi_j\rangle_\beta$ of all the possible Pauli strings acting on the qubits included in $\hat{\rho}_{j,\beta}^k$ and $\vec{a}_j^{\,k}(\hat{\rho}')$ being the vector which elements are the expectation values of these Pauli strings on the density matrix $\hat{\rho}'$

$$\vec{p}_{j,\beta}^{\,k} \doteq \left\{ \left(\vec{p}_{j,\beta}^{\,k}\right)_m = \langle \psi_j|_\beta \hat{P}_m |\psi_j\rangle_\beta \right\}_m, \tag{3.86}$$

$$\vec{a}_j^{\,k}(\hat{\rho}') \doteq \left\{ \left(\vec{a}_j^{\,k}(\hat{\rho}')\right)_m = \mathrm{Tr}\left[\hat{P}_m \hat{\rho}'\right] \right\}_m. \tag{3.87}$$

The solutions to the optimization problem in Eq. (3.85) are constrained to both have unit trace and being positive semi-definite. This is implemented numerically using the CVXPY Python script [79, 80] with MOSEK solver [81]. The reduced

density matrices obtained in this way are then used to evaluate the expectation values of any one-body or two-body observable of interest by taking the trace of their product with the desired observable

$$\langle \hat{O} \rangle_{j,\beta} \approx \langle \psi_j(\vec{\tilde{\theta}}_j)|_\beta \, \hat{O} \, |\psi_j(\vec{\tilde{\theta}}_j)\rangle_\beta = \mathrm{Tr}\left[\hat{\rho}^k_{j,\beta}\hat{O}\right]. \tag{3.88}$$

Due to the finite number of shots taken for every measurement, each $\vec{p}^{\,k}_{j,\beta}$ element $\left(\vec{p}^{\,k}_{j,\beta}\right)_m$ carries an error $\sigma\left(\vec{p}^{\,k}_{j,\beta}\right)_m$, estimated as the standard error computed from the measurement outcomes (see Section 1.2.3 and Eq. (1.44)). In order to transfer this uncertainty to $\langle \hat{O} \rangle_{j,\beta}$, the following procedure is followed: M samples of $\vec{p}^{\,k}_{j,\beta}$ are taken and for each of them Eq. (3.85) is solved, giving a set of M reduced density matrices. Each of these matrices is used to compute the expectation value of operator $\hat{O}$ through Eq. (3.88) and the final value for $\langle \hat{O} \rangle_{j,\beta}$ is obtained by taking the sample mean $\bar{O}_{j,\beta}$ of these values.

$$\langle \hat{O} \rangle_{j,\beta} \cong \bar{O}_{j,\beta} \doteq \sum_{m=1}^{M} \frac{O^{(j,\beta)}_m}{M}, \tag{3.89}$$

where $\{O^{(j,\beta)}_m\}_{m\in\{1,\dots,M\}}$ is a set of samples taken at fixed j and β. The error $\delta\bar{O}_{j,\beta}$ on the $\langle \hat{O} \rangle_{j,\beta}$ estimation $\bar{O}_{j,\beta}$ is computed as the standard deviation of the samples

$$\delta\bar{O}_{j,\beta} \doteq \sqrt{\frac{\sum_{m=1}^{M}\left(O^{(j,\beta)}_m - \bar{O}_{j,\beta}\right)^2}{M}}, \tag{3.90}$$

so that in the end we estimate the observable as $\langle \hat{O} \rangle_{j,\beta} \cong \bar{O}_{j,\beta} \pm \delta\bar{O}_{j,\beta}$.
How many Pauli strings measurements are needed to achieve such result? In general in fact, any N-body reduced density matrix can always be fully determined by measuring every possible Pauli string of length N, i.e. by measuring in principle 4^N operators consisting of all the possible combinations of Pauli operators (and identity) acting on the N qubits.
In this neutrino case though, I recall that in Section 3.1.2 it was shown how any two-body reduced thermal density matrix of the system evolved under $\hat{H}_{2B}$ starting from a computational basis state can always be written in the following form (see Eq. (3.42)), thanks to the $\mathcal{U}(1)$ symmetry and the thermal evolution being a real operator

$$\begin{aligned}
\hat{\rho}_{ij}(\tau) &\doteq \mathrm{Tr}_{k\notin\{i,j\}}\left[\hat{\rho}(\tau)\right] \\
&= \frac{1}{4}\Big(\hat{\mathbb{I}}_4 + \hat{Z}_i\mathrm{Tr}[\hat{Z}_i\hat{\rho}_{ij}(\tau)] + \hat{Z}_j\mathrm{Tr}[\hat{Z}_j\hat{\rho}_{ij}(\tau)] \\
&\qquad + \hat{Z}_i\hat{Z}_j\mathrm{Tr}[\hat{Z}_i\hat{Z}_j\hat{\rho}_{ij}(\tau)] \\
&\qquad + (\hat{X}_i\hat{X}_j + \hat{Y}_i\hat{Y}_j)\mathrm{Tr}[\hat{X}_i\hat{X}_j\hat{\rho}_{ij}(\tau)]\Big),
\end{aligned} \tag{3.91}$$

which only needs the measurements of two distinct independent Pauli strings to be determined.
This means that on any state $|\phi_j\rangle_\beta$ it is sufficient to measure all the Pauli strings having an even number of $\hat{X}$ operators and $\hat{Z}$ on all the remaining positions to be able to fully reconstruct any desired one-body or two-body reduced density matrix: more explicitly, the needed strings are the following 8 ones

$$\begin{aligned}
&\hat{P}_1 = \hat{Z}\otimes\hat{Z}\otimes\hat{Z}\otimes\hat{Z}, \quad \hat{P}_2 = \hat{Z}\otimes\hat{Z}\otimes\hat{X}\otimes\hat{X}, \quad \hat{P}_3 = \hat{Z}\otimes\hat{X}\otimes\hat{Z}\otimes\hat{X},\\
&\hat{P}_4 = \hat{Z}\otimes\hat{X}\otimes\hat{X}\otimes\hat{Z}, \quad \hat{P}_5 = \hat{X}\otimes\hat{Z}\otimes\hat{Z}\otimes\hat{X}, \quad \hat{P}_6 = \hat{X}\otimes\hat{Z}\otimes\hat{X}\otimes\hat{Z}, \quad (3.92)\\
&\hat{P}_7 = \hat{X}\otimes\hat{X}\otimes\hat{Z}\otimes\hat{Z}, \quad \hat{P}_8 = \hat{X}\otimes\hat{X}\otimes\hat{X}\otimes\hat{X}.
\end{aligned}$$

All of the other Pauli expectation values are either zero or can be directly inferred from these. For instance

$$\mathrm{Tr}\Big\{\hat{Y}\otimes\hat{Z}\otimes\hat{Y}\otimes\hat{Z}\hat{\rho}\Big\} = \mathrm{Tr}\Big\{\hat{X}\otimes\hat{Z}\otimes\hat{X}\otimes\hat{Z}\hat{\rho}\Big\} = \mathrm{Tr}\Big\{\hat{P}_6\hat{\rho}\Big\}\,, \tag{3.93}$$

due to the rotational symmetry. Furthermore, for estimating only one and two-qubit observables only 6 measurement setups are required since we do not need $\hat{P}_1$ nor $\hat{P}_8$.
At this point let's check how many of the states $\{|\phi_j\rangle_\beta\}_{j\in\{1,\dots,16\}}$ at any given β are actually needed to be physically measured in order to be still able to reconstruct the needed one-body and two-body reduced density matrices fully.
I start by recalling that at the end of Section 3.1.2 it was shown that the neutrino-neutrino interaction term $\hat{H}_{\nu\nu}$ (and therefore also the 2-body interaction $\hat{H}_{2B}$) is invariant under two transformations: the flip of all particle flavors $\hat{U}_X$ and the inversion of the particle order $\hat{U}_{\mathrm{inv}}$

$$\hat{U}_X \doteq \prod_{j=1}^{4}\hat{X}_j, \tag{3.94}$$

$$\hat{U}_{\mathrm{inv}} \mid \hat{U}_{\mathrm{inv}}\bigotimes_{j=1}^{4}|\phi_j\rangle = \bigotimes_{j=1}^{4}|\phi_{4-j+1}\rangle\,. \tag{3.95}$$

A very nice feature of these two transformations is that both belong to the Clifford group [82], and, as such, they normalize the Pauli group, which means that they map it to itself under conjugation, up to a phase. In other words, $\hat{U}_X$ and $\hat{U}_{\mathrm{inv}}$ both transform Pauli operators into other Pauli operators times a constant $c\in[\pm1,\pm i]$, i being the imaginary unit.

$$\begin{aligned}
&\hat{U}\hat{P}\hat{U}^\dagger = c_{U,P}\hat{P}' \mid c_{U,P}\in\{\pm1,\pm i\}\,,\ \hat{P}^{(\prime)} \doteq \bigotimes_j \hat{P}_j^{(\prime)}\\
&\forall\ \hat{U}\in\Big\{\hat{U}_X,\hat{U}_{\mathrm{inv}},\hat{U}_X\hat{U}_{\mathrm{inv}}\Big\}\,,\ \hat{P}_j^{(\prime)}\in\Big\{\hat{\mathbb{I}},\hat{X},\hat{Y},\hat{Z}\Big\}\,.
\end{aligned} \tag{3.96}$$

This is very convenient for our case because, if $|j\rangle$ and $|k\rangle$ are two computational states such that $|j\rangle = \hat{U}|k\rangle$ with $\hat{U} \in \left\{\hat{U}_X, \hat{U}_{\text{inv}}, \hat{U}_X\hat{U}_{\text{inv}}\right\}$, then at any inverse temperature β all the possible Pauli expectation values evaluated on the state $|\phi_j\rangle_\beta$ evolved from $|j\rangle$ can be automatically obtained from those measured on $|\phi_k\rangle_\beta$, i.e. the state evolved from $|k\rangle$

$$\langle\phi_j|_\beta \hat{P}|\phi_j\rangle_\beta = \langle\phi_k|_\beta \hat{U}^\dagger\hat{P}\hat{U}|\phi_k\rangle_\beta = c_{U,P}\langle\phi_k|_\beta \hat{P}'|\phi_k\rangle_\beta\,, \tag{3.97}$$

with $\hat{P}$ and $\hat{P}'$ as in Eq. (3.96).
The way I proceed is therefore to select some states from $\{|\phi_j\rangle_\beta\}_{j\in\{1,\dots,d\}}$ that I call **independent** onto which I measure the expectation values of operators in (3.92), while the Pauli expectation values on the remaining **dependent** states are computed from the performed measurements by exploiting Eq. (3.97).
The specific independent states are somewhat arbitrary: I take those generated starting from the following computational basis states

$$\{|0000\rangle, |0001\rangle, |0010\rangle, |0011\rangle, |0101\rangle, |0110\rangle\}, \tag{3.98}$$

while all the others can be obtained from these through the following symmetries:

$$\begin{aligned}
&|0100\rangle = \hat{U}_{\text{inv}}|0010\rangle && |0111\rangle = \hat{U}_X\hat{U}_{\text{inv}}|0001\rangle && |1000\rangle = \hat{U}_{\text{inv}}|0001\rangle\\
&|1001\rangle = \hat{U}_X|0110\rangle && |1010\rangle = \hat{U}_{\text{inv}}|0101\rangle && |1011\rangle = \hat{U}_X\hat{U}_{\text{inv}}|0010\rangle\\
&|1100\rangle = \hat{U}_{\text{inv}}|0011\rangle && |1101\rangle = \hat{U}_X|0010\rangle && |1110\rangle = \hat{U}_X|0001\rangle\\
& && |1111\rangle = \hat{U}_X|0000\rangle\,. &&
\end{aligned} \tag{3.99}$$

Actually, states $|0000\rangle$ and $|1111\rangle$ are eigenstates of $\hat{H}_{2B}$, which means that there is no need to measure Paulis on states evolved from them, rather all the needed values can be computed once at the start of the computation and used throughout all the evolution.
Summing up, in order to obtain the estimate $\langle\hat{O}\rangle_\beta$ given in Eq. (3.82) for the thermal expectation value of some generic observable $\hat{O}$, the execution of 40 circuits per evolution step are needed: 8 Pauli strings measures times 5 independent QITE evolved states. If $\hat{O}$ is further only a one or two-qubit operator, then only 6 different Pauli strings must be measured, reducing the total number of required circuits to 30.

3.4.3 QITE implementation: Givens parameter computation

As already stated in Section 3.3, the QITE algorithm approximates states $|\phi_j\rangle_\beta$ defined in Eq. (3.61) by finding the correct parameters $\vec{\theta}_j$ for a given unitary

ansatz $\hat{U}_\beta(\vec{\theta})$, which in this case consists of the product of all the possible real Givens rotation on the system (see Eq. (3.75)). These parameters are found by solving Eq. (3.68), where the elements of matrix $\mathbf{A}_j$ and vector $\vec{b}_j$ are computed as expectation values of commutators and anticommutators involving the ansatz generators and the Hamiltonian (see Eqs. (3.67), (3.69)), evaluated on the state before the QITE evolution step.
The procedure followed to obtain these expectation values is a bit different than what was explained in previous Section 3.4.2, as it is not based on the reconstruction of many reduced density matrices, since the needed computational time would be too demanding, instead it directly relies on the measures $\vec{p}^{\,k}_{j,\beta}$ of the Pauli strings performed on the system (see Eq. (3.86)), where β value here is that of the inverted temperature before performing the QITE step evolution.
Each operator appearing in the elements of $\mathbf{A}_j$ and $\vec{b}_j$ is therefore decomposed as a combination of Pauli operators, and the expectation value of the full operator is obtained as the same combination of the measured expectation values of the Pauli terms, following a procedure identical to the one explained at the end of Section 1.2.3.
At this point, to further propagate the error from matrix $\mathbf{A}_j$ and vector $\vec{b}_j$ to the estimated Givens parameters $\vec{\tilde{\theta}}_j$, as well as to obtain a quite stable and reliable estimate of these angles, I proceeded in the following way: N_θ different samples of both matrix $\mathbf{A}_j$ and vector $\vec{b}_j$ are taken, sampling each individual element from a normal distribution centered on the measured value and having the same variance as the element itself. Then, for each set of samples Eq. (3.68) is solved, producing N_θ vectors of Givens parameters $\{\vec{\theta}_{j_n}\}_{n\in\{1,\dots,N_\theta\}}$: the final estimation $\vec{\tilde{\theta}}_j$ is then retrieved as the circular mean of the N_θ samples, with period $4\pi/\beta$. More explicitly, if $\left(\vec{\tilde{\theta}}_j\right)_l$ is the l-th element of $\vec{\tilde{\theta}}_j$ and $\left(\vec{\theta}_{j_n}\right)_l$ the same element of its n-th sample, than the following identity is used

$$\left(\vec{\tilde{\theta}}_j\right)_l = \frac{2}{\beta}\mathrm{Arg}\left[\frac{1}{N_\theta}\sum_{n=1}^{N_\theta} e^{i\frac{\beta}{2}\left(\vec{\theta}_{j_n}\right)_l}\right]. \tag{3.100}$$

The error $\sigma\left(\vec{\tilde{\theta}}_j\right)_l$ on each element of $\vec{\tilde{\theta}}_j$ is instead estimated as the circular standard deviation of the samples

$$\sigma\left(\vec{\tilde{\theta}}_j\right)_l = \frac{2}{\beta}\sqrt{-2\ln\left(\left|\frac{1}{N_\theta}\sum_{n=1}^{N_\theta} e^{i\frac{\beta}{2}\left(\vec{\theta}_{j_n}\right)_l}\right|\right)}. \tag{3.101}$$

The rescaling factor $\frac{2}{\beta}$ introduced both in Eq. (3.100) and (3.101) comes from the QITE angles definitions given in Eq. (3.75).

3.4.4 QITE implementation: step merge procedure

Let's discuss in more depth the QITE algorithm implementation via the division of the full evolution into several steps. In this scenario, if every step evolves a state for an inverse temperature $\Delta\beta$, at step M the state $|j\rangle$ will have evolved to the state $|\phi_j\rangle_{M\Delta\beta}$

$$|\phi_j\rangle_{M\Delta\beta} = \hat{U}_{j,M\Delta\beta}\,|j\rangle \doteq \hat{U}'_{j,M}(\Delta\beta)\cdot\hat{U}'_{j,M-1}(\Delta\beta)\cdot\ldots\cdot\hat{U}'_{j,1}(\Delta\beta)\,|j\rangle\,, \tag{3.102}$$

where $|\phi_j\rangle_\beta$ and $\hat{U}_{j,\beta}$ were introduced in Eq. (3.61), while $\hat{U}'_{j,m}(\Delta\beta)$ is the unitary operator responsible for the m-th evolution step of size $\Delta\beta$

$$\hat{U}'_{j,m}(\Delta\beta)\,|\phi_j\rangle_{(m-1)\Delta\beta} \doteq |\phi_j\rangle_{m\Delta\beta}\,. \tag{3.103}$$

As already discussed (see Section 3.3), for each m-th step of size $\Delta\beta$ QITE gives an approximation $\hat{U}_{\Delta\beta}(\vec{\tilde{\theta}}_{j,m})$ to $\hat{U}'_{j,m}(\Delta\beta)$, which corresponds to the circuit in Fig. 3.5 implemented with parameters determined by the measurement of some operators on the state just before the evolution step (see Eq. (3.67), where here $|\psi_j(\vec{\tilde{\theta}}_{j,m-1})\rangle_{(m-1)\Delta\beta}$ would substitute $|j\rangle$).
The state $|\phi_j\rangle_{M\Delta\beta}$ is then approximated with $|\psi_j(\vec{\tilde{\theta}}_{j,M})\rangle_{M\Delta\beta}$, obtained on the quantum computer by applying M copies of the circuit in Fig. 3.5 (each with the appropriate choice for the parameters) to the circuit generating the state $|j\rangle$. This process soon becomes unsustainable, since it requires too many resources: each copy of the circuit in Fig. 3.5 requires 30 CNOTs to be implemented (counting 3 CNOTs for every SWAP gate and 2 for each Givens rotations), which means that very few steps can be performed before the accuracy of the computation is lost.
To overcome this issue, I rely on the **step merge** procedure, a heuristic approach which consists of the substitution of the products of the M circuits $\{\hat{U}_{\Delta\beta}(\vec{\tilde{\theta}}_{j,m})\}_{m\in\{1,\ldots,M\}}$ with a single circuit which parameters are each one the sum of the different values that such a parameter would take in the different M circuits

$$\hat{U}_{\Delta\beta}(\vec{\tilde{\theta}}_{j,M})\cdot\hat{U}_{\Delta\beta}(\vec{\tilde{\theta}}_{j,M-1})\cdot\ldots\cdot\hat{U}_{\Delta\beta}(\vec{\tilde{\theta}}_{j,1}) \simeq \hat{U}_{\Delta\beta}\left(\sum_{m=1}^{M}\vec{\tilde{\theta}}_{j,m}\right) \tag{3.104}$$

This approach keeps the circuit depth fixed at any β value and is correct for $\Delta\beta$ being sufficiently small, but I do not further investigate the specific extent of its accuracy. Part of the present work also concerns the testing of this procedure, heuristically checking whether it can allow for reliable computations or not.

3.5 Results

In this section I present the results we got from the different simulations.
First of all I introduce the analyzed observables and their properties, then I move to the presentations of the actual computed expectation values. I show three different simulation types:

- First of all, I compare the exact evolution, computed by means of an exact diagonalization of the small 4-neutrino problem, with an ideal implementation of the QMETTS algorithm, simulated by computing up to machine precision the true system state vector after each thermal evolution step;

- A second simulation type is performed by means of a noiseless QASM simulation of the QMETTS algorithm, which simulates the actual hardware measurement procedure by providing a measured bitstring for each run and therefore allows for an estimation of the error coming from this source (see Section 1.2.3);

- A last kind of simulation is always performed thanks to a QASM simulator, this time mimicking a real IBM quantum device, considering also features like connectivity, error rates, noise parameters and native gates. For this type of simulation we equipped the implemented QMETTS algorithm with two error mitigation techniques: dynamical decoupling and readout error mitigation (see Section 3.5.4)

3.5.1 Observables

The analysis is performed by mainly focusing on the estimation of some key few-body observables and comparing the obtained values with the ideal ones, computed by means of linear algebra. In the following I introduce the specific addressed observables, which are the energy, the qubit purity and the mutual information between qubit pairs: all together, these observables should give a rather informative description of the system evolution, quantifying the level of convergence and the entanglement and correlation spread.

System energy

The first and principal observable is the system energy U, which corresponds to the expectation value on the system thermal density matrix of the following operator

$$\hat{H} = \hat{H}_{1B}^{(d)} + \hat{H}_{2B} \implies U(\hat{\rho}) \doteq \mathrm{Tr}\left[\hat{H}\hat{\rho}\right] \tag{3.105}$$

where the one-body term $\hat{H}_{1B}^{(d)}$ was introduced in Eq. (3.81) while the two-body term $\hat{H}_{2B}$ in Eq. (3.34).
The thermal evolution effect on the system state is effectively that of filtering out all contributions coming from excited states, leaving the system in the ground state after a proper imaginary time, or inverse temperature, has been reached.
Checking the energy behavior as a function of β allows to understand whether or not the system converges at all to a stable state, and comparing with the expected evolution, the quality of the algorithm may be estimated.
The ground state of the neutrino system under analysis depends on the γ parameter, the relative strength between the two-body and one-body contributions: we will therefore perform different simulations changing such parameter, considering all integers in the range $[1, 10]$. We expect to get results closer to the ideal value for smaller γ values, since the effective quantum evolution is only due to the two-body contribution, which is proportional to γ.
It is relevant to notice that in the explored range of γ values, the ground state of the system changes between two different eigenstates, which eigenvalues are expected to cross for $\gamma = 3.81$: this suggests that two different system states should appear at system thermalization, depending on the value of γ.

Purity

The purity (P) is a quantity characterizing a quantum state and in particular it is computed from the density matrix $\hat{\rho}$ associated to its state: it is defined as the trace of the density matrix squared and it is constrained to be 1 for pure states, while it decreases for mixed states, the smallest possible value being $1/d$ with d the dimension of the Hilbert space of the state

$$P(\hat{\rho}) \doteq \frac{1}{d} \leq \mathrm{Tr}\left[\hat{\rho}^2\right] \leq 1. \tag{3.106}$$

In particular, one can compute the purity of the reduced density matrices associated to single qubits: in this case, the purity is sensitive both to the 'mixedness' of the full quantum state, but also to the entanglement in the system. If the full system state is pure (as should be for ground states, extracted by the ideal thermal evolution) but characterized by entanglement among some qubits, the reduced density matrix of the single qubits participating in the entanglement will be mixed, so that its purity will be smaller than 1. In particular, the smallest possible value for single-qubit purity is $P = 1/2$, corresponding to the maximally mixed single-qubit state.

Mutual information

The mutual information (I) is another few-body observable sensitive to correlations in the system, both quantum and classical. Its definition is based on the von Neumann entropy S, which is the generalization to quantum systems of the classical Shannon entropy and can be computed as follows from the system density matrix $\hat{\rho}$

$$S(\hat{\rho}) \doteq -\operatorname{Tr}\left[\hat{\rho}\log_2\left(\hat{\rho}\right)\right]. \tag{3.107}$$

Exploiting this definition of the von Neumann entropy and considering a quantum system which can be divided into two parts A and B, mutual information can be further defined as the relative entropy between the original state and the one consisting of the two reduced states of the systems A and B

$$I_{A:B} \doteq S(\hat{\rho}_A) + S(\hat{\rho}_B) - S(\hat{\rho}_{AB}), \tag{3.108}$$

where $\hat{\rho}_{AB}$ is the density matrix that describes the state of the entire system and $\hat{\rho}_A$ and $\hat{\rho}_B$ are the reduced density matrices that correspond to the subsystems A and B, respectively.
From these definitions, it is quite evident that the mutual information on a system of two qubits must be bounded between 0, if the two qubits are not correlated, and 2, if the two qubits are maximally correlated.

3.5.2 Ideal simulations

Here I propose a comparison between the exact simulation of the system and the ideal implementation of the QMETTS algorithm, obtained by generating the true system state vectors at each evolution step.
The thermal evolution in inverse temperature β is studied by setting the initial density matrix of the system as the identity, corresponding to $\beta = 0$ ($e^{-0\hat{H}} \equiv \hat{\mathbb{I}}$, $\hat{H}$ as in Eq. (3.105)), and then performing an evolution until $\beta = 10$, dividing it in 20 equal steps of duration $\Delta\beta = 0.5$.
In Fig. 3.6 the computed energy $U(\beta)$ expectation values for 10 different values of the γ parameter, more specifically the integers in the range $[1, 10]$, are reported, as computed in three different ways:

- Dashed lines represent exact values, computed by diagonalizing the Hamiltonian $\hat{H}$ given in Eq. (3.105) and using the obtained eigenvalues $\{U_i\}_{i\in\{1,\dots,16\}}$ to compute thermal energies at a given γ with the following formula

$$U_{\mathbf{ex}}(\gamma,\beta) = \sum_{i=1}^{16} \frac{e^{-\beta U_i}}{\sum_{j=1}^{16} e^{-\beta U_j}} U_i. \tag{3.109}$$

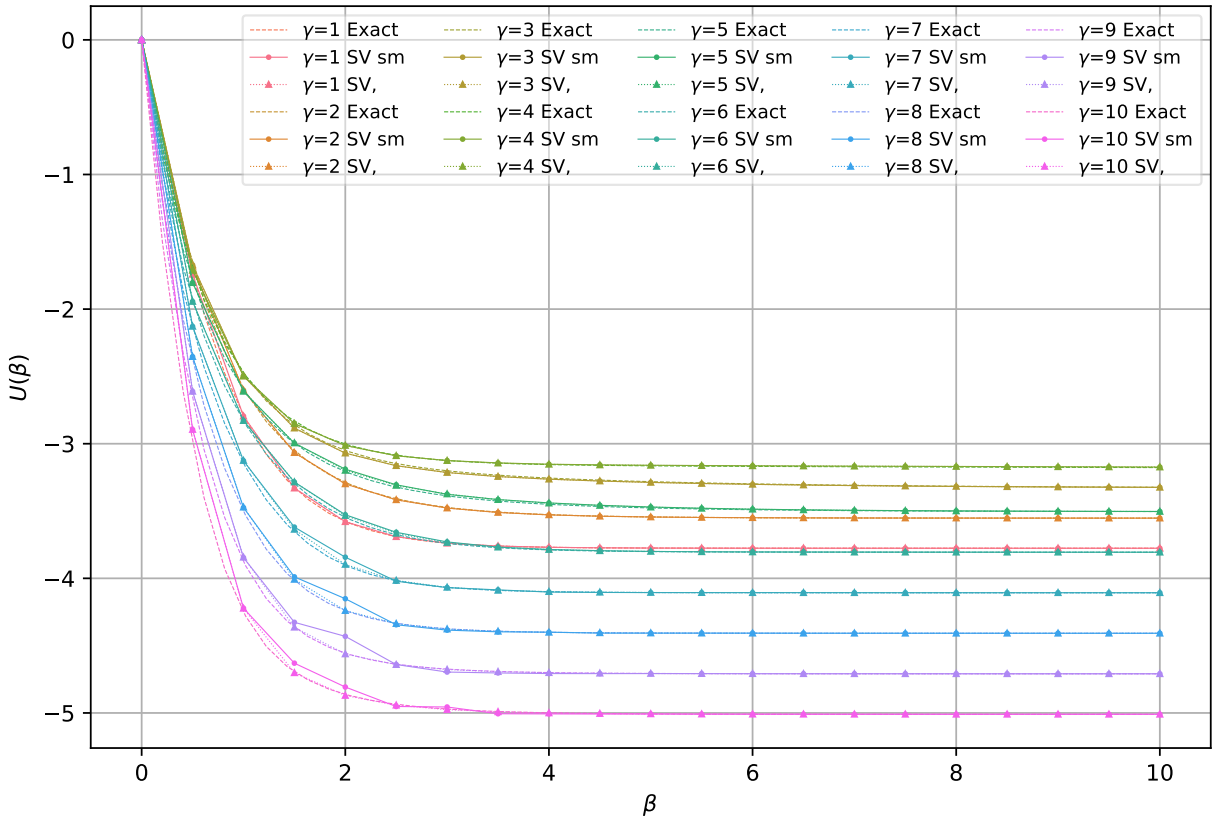


Figure 3.6: Colored plot showing the system energy expectation values, as a function of the inverse temperature β. Different colors correspond to different $\gamma \in [1, 10]$ values. Dashed lines are the exact values. Dotted lines with triangular markers are the values computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the complete QMETTS algorithm, omitting the step merge approximation. Solid lines with dot markers are the values computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the approximated QMETTS algorithm, including the step merge approximation.

This line is represented considering β as a continuous real parameter rather than taking only the discrete set of values described before;

- Dotted lines with triangular markers are the values obtained by implementing the QMETTS algorithm described in the previous sections, omitting the step merge approximating procedure (see Section 3.4.4) and accessing the

actual state vectors generated at each evolution step thanks to the Qiskit Aer *StatevectorSimulator*: this is the ideal realization of the complete QMETTS algorithm, obtained by dividing the full evolution from $\beta = 0$ to $\beta = 10$ in 20 identical steps;

- Solid lines with dot markers represent the values obtained by the implementation of the QMETTS algorithm including the step merge procedure (see Section 3.4.4) and accessing the actual state vectors generated at each evolution step thanks to the Qiskit Aer *StatevectorSimulator*: this is the ideal realization of the approximated QMETTS algorithm, always obtained by dividing the full evolution from $\beta = 0$ to $\beta = 10$ in 20 identical steps.

Fig. 3.6 clearly shows that both the full and (step merge) approximated ideal realizations of the QMETTS algorithm would lead to the system convergence in energy after a few iterations. Furthermore, the level of agreement of both these two simulation implementations with the exact system behavior is particularly satisfying, meaning that the system thermal energy reached at the end of the simulations is indeed the correct one, for all the considered values of γ and regardless of whether the step merge procedure is employed or not.
A clear description of the compatibility between the three results cannot be performed though, as these expectation values are not associated with any error, since they are not obtained through the usual quantum measurement procedure described in Section 1.2.3. In order to give a clearer representation of the distance between the results coming from the QMETTS algorithm and the ideal values, the relative absolute distance between the two values are visually represented in Fig. 3.7 for all the considered values of γ, where this relative distance $\Delta U_{\text{ex-sv}}(\beta)$ is defined as the following

$$\Delta U_{\text{ex-sv}}(\beta) \doteq \left| \frac{U_{\text{ex}}(\beta) - U_{\text{sv}}(\beta)}{U_{\text{ex}}(\beta)} \right|, \tag{3.110}$$

with $U_{\text{ex}}(\beta)$ the exact energy expectation value, and $U_{\text{ev}}(\beta)$ the QMETTS estimation obtained with the Qiskit Aer *StatevectorSimulator*, either with or without the step merge approximation. As expected, lower γ values show a lesser relevance of the step merge approximation, since this impacts only the two-body evolution, which is more suppressed at small gammas: as γ grows, the difference between lines referring to the step merge implementation and those referring to the complete QMETTS implementation begin to differ more and more, with the step merge simulations struggling in achieving the same level of accuracy as the complete QMETTS implementation.
Furthermore, for nearly every value of γ, the complete QMETTS implementation

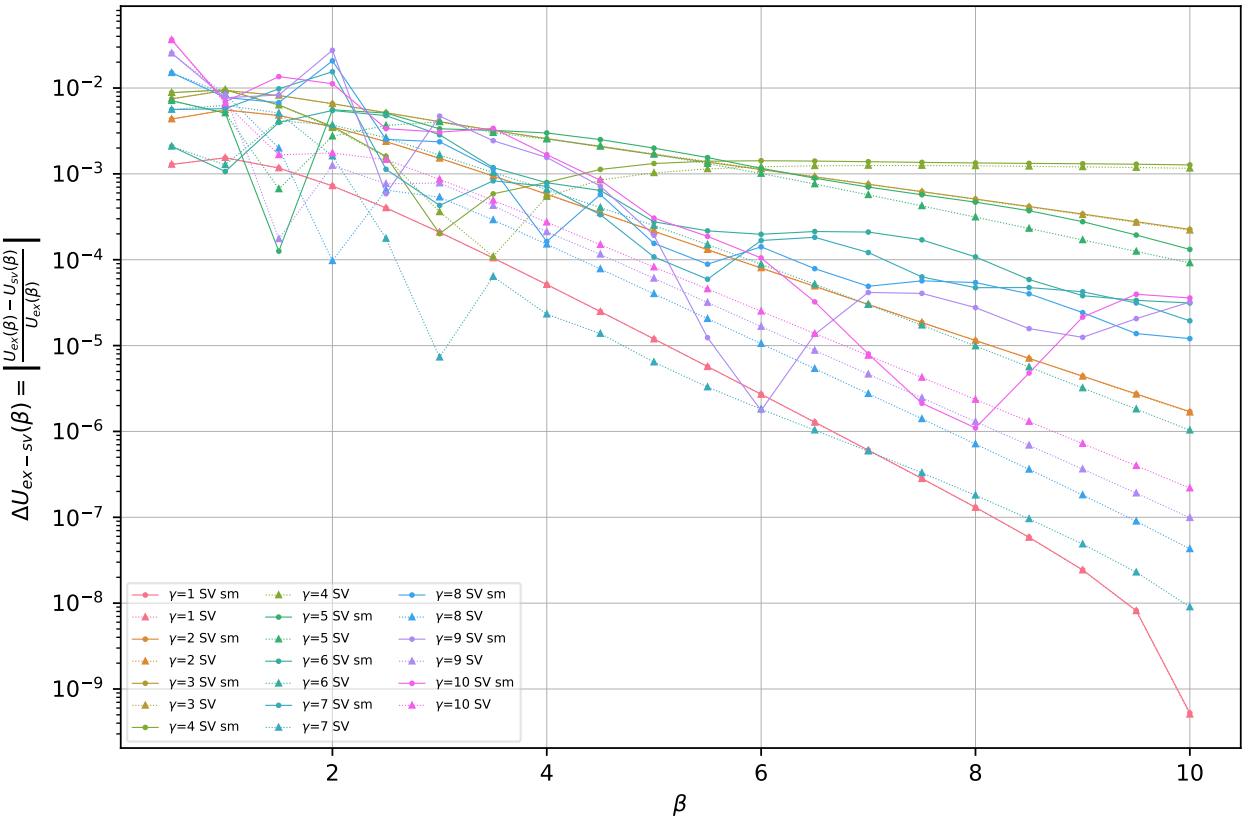


Figure 3.7: Colored plot showing the relative distance between the exact values of the energy and those obtained from the Qiskit Aer *StatevectorSimulator* of the QMETTS algorithm (see Eq. (3.110)), as a function of the inverse temperature β. Different colors correspond to different $\gamma \in [1, 10]$ values. Dotted lines with triangular markers are the values computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the complete QMETTS algorithm, omitting the step merge approximation. Solid lines with dot markers are the values computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the approximated QMETTS algorithm, including the step merge approximation.

approximates better the exact thermal evolution the more β increases: we hypothesize that this is due to the good level of accuracy with which the algorithm is able to construct the system ground state, to which the system state is always closer the more the simulation progresses. The only exception to this behavior is represented by the $\gamma = 4$ curve, this possibly being related to the fact that, as

briefly introduced in Section 3.5.1, the system ground state changes when the γ parameter crosses the 3.81 value: $\gamma = 4$ may be close enough to this critical value that the system struggles more to select the ground state, reducing the level of accuracy that the QMETTS algorithm can achieve at a given β.
This "2-ground states" structure of the system in the explored γ parameter range of values is evident also in the other examined observables, as it can be seen in Figs. 3.8 and 3.9 where the purity and mutual information are represented, respectively: the system ground state for $\gamma < 3.81$ is the completely separable state $|\psi\rangle_{g1}$

$$|\psi\rangle_{g1} \doteq |1111\rangle\,, \tag{3.111}$$

which is correctly reflected by the convergence of purity to value 1 on all four qubits and by the mutual information between any couple of qubits converging to the value 0, as it must be for completely separable states.
For $\gamma > 3.8$ the system ground state is instead $|\psi\rangle_{g2}$

$$\begin{aligned} |\psi\rangle_{g2} &\approx 0.695(|1110\rangle - |0111\rangle) + 0.127(|1101\rangle - |1011\rangle) \\ &\sim \frac{1}{\sqrt{2}}(|1110\rangle - |0111\rangle) \equiv |\Phi_-\rangle_{14} \otimes |11\rangle_{23}\ , \end{aligned} \tag{3.112}$$

where I defined $|\Phi_-\rangle \doteq \frac{1}{\sqrt{2}}(|10\rangle - |01\rangle)$ one of the maximally two-qubit entangled state, known as Bell state, and where I added subscripts to states expressing the qubits described by them.
According to Eq. (3.112), the ground state of the system at high γ values is quite close to a state where the inner qubits 2 and 3 are in a separable state, while the external qubits 1 and 4 are maximally entangled. This feature is also well described by the purities and mutual information reported in Figs. 3.8 and 3.9: the purity of the internal qubits is still pretty close to the maximum value 1, slightly decreasing as γ increases, while the purity of the external qubits rapidly reaches the value 1/2, reflecting the fact that they are in a maximally entangled state. Similarly, at convergence mutual information between any pair of qubits is pretty low, just a little higher than the low γ case, with the exception of the qubits 1 and 4 couple, for which it has a dramatic increase and approaches very closely the maximum possible value of 2.
This behavior characterizes at convergence all the simulations but the one obtained by setting $\gamma = 4$, which is slower in reaching convergence: as already stated, this is probably due to $|\psi\rangle_{1g}$ and $|\psi\rangle_{2g}$ being too close in energy at this γ, so that a longer evolution is needed to observe the same properties.
At the start of each computation, for $\beta = 0$, the observables correctly do not depend on γ: the system is always in the maximally mixed state $\hat{\rho}_0 \doteq \hat{\rho}(\beta = 0) = \frac{1}{16}\hat{\mathbb{I}}$ and this is consistent with what is observed in the figures. The system internal

energy at $\beta = 0$ must be proportional to $\mathrm{Tr}\left[\hat{H}\right]$, with $\hat{H}$ given in Eq. (3.105), and by recalling that Pauli matrices are traceless and that $\mathrm{Tr}\left[\hat{A} \otimes \hat{B}\right] = \mathrm{Tr}\left[\hat{A}\right] \mathrm{Tr}\left[\hat{B}\right]$ for any two quantum operators $\hat{A}$ and $\hat{B}$, it follows that $U(\beta = 0) = 0$ for every γ. Similarly, all the reduced density matrices at $\beta = 0$ are also maximally mixed and proportional to the identity: this means that all single-qubit purities must start from the minimum possible value $P = 1/2$, while the mutual information at the start of each simulation is null, reflecting the absence of correlation among qubits at high temperatures.

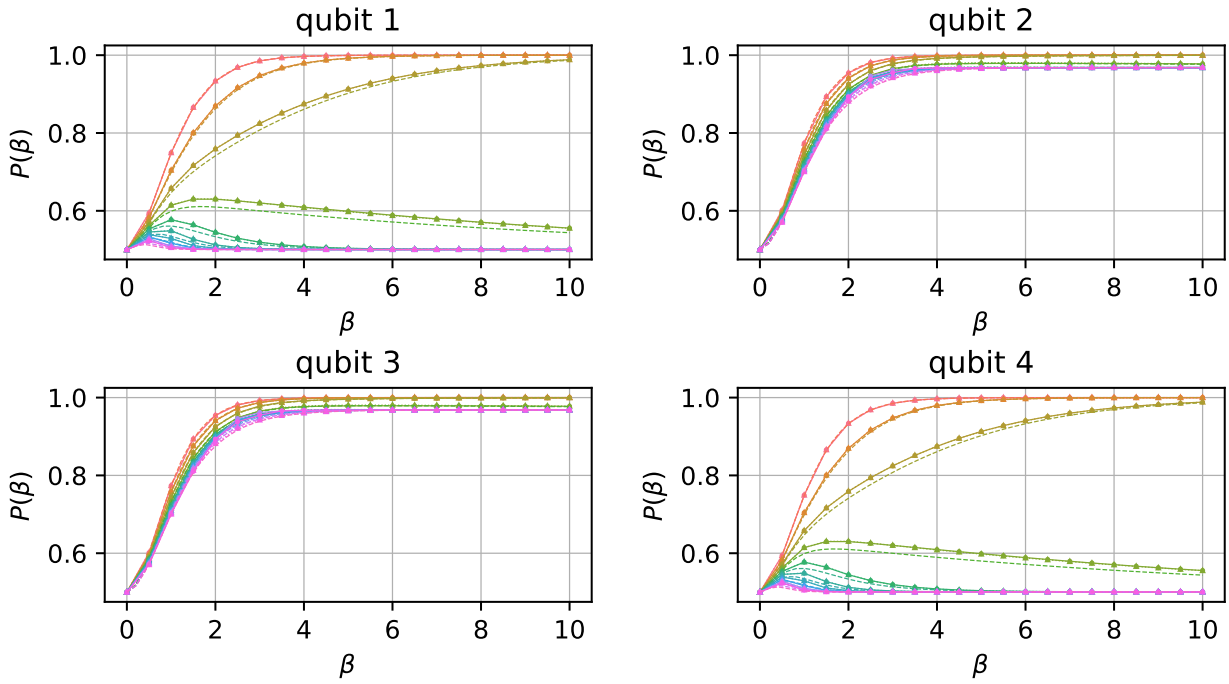


Figure 3.8: Colored plots showing the purities expectation values of each qubit in the system, as a function of the inverse temperature β, qubits 1 and 4 being the external ones while qubits 2 and 3 being the internal ones. Different colors correspond to different $\gamma \in [1, 10]$ values. Dashed lines are the exact values. Dotted lines with triangular markers are the values computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the complete QMETTS algorithm, omitting the step merge approximation. Solid lines with dot markers are the values computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the approximated QMETTS algorithm, including the step merge approximation. Legend is the same as in Fig. 3.6, omitted here to favor readability.

While the correct expectation values are retrieved with both the QMETTS sim-

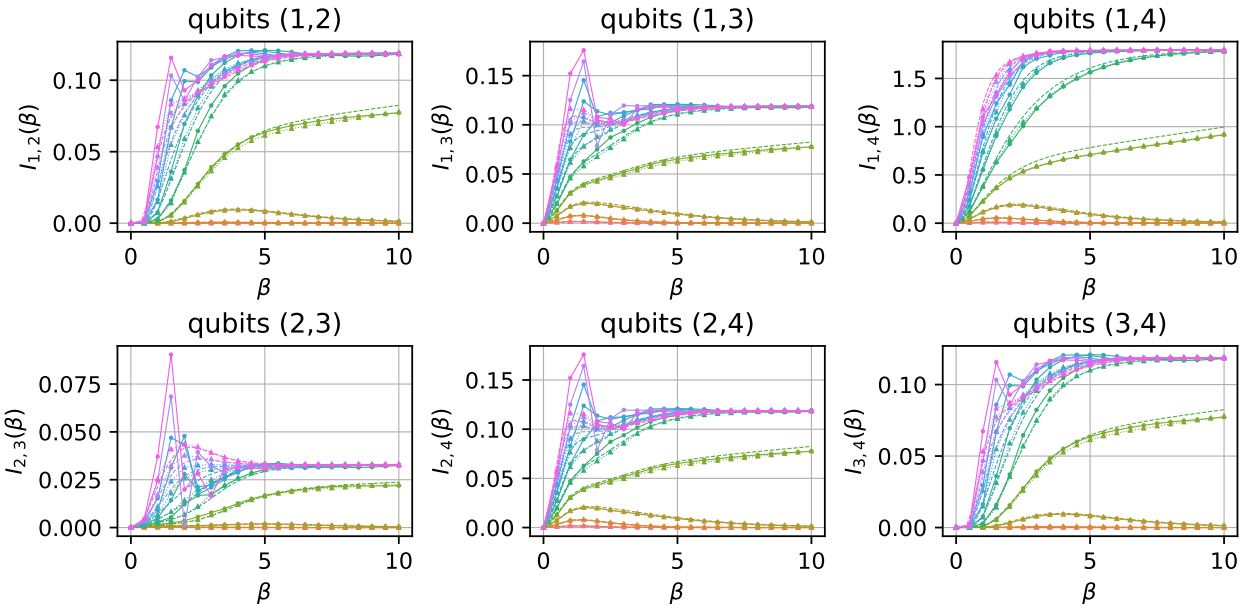


Figure 3.9: Colored plots showing the mutual information expectation values of each qubit pair in the system, as a function of the inverse temperature β, qubits 1 and 4 being the external ones while qubits 2 and 3 being the internal ones. Different colors correspond to different $\gamma \in [1, 10]$ values. Dashed lines are the exact values. Dotted lines with triangular markers are the values computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the complete QMETTS algorithm, omitting the step merge approximation. Solid lines with dot markers are the values computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the approximated QMETTS algorithm, including the step merge approximation. Legend is the same as in Fig. 3.6, omitted here to favor readability.

ulations for $\beta \gtrsim 5$, the predictions at higher temperatures (smaller β) are quite distant from the exact values, suggesting that the implemented algorithm is particularly suitable for the extraction of ground state properties, and that it requires more resources to be applied for finite temperature estimations.

3.5.3 Noiseless simulations

A more realistic simulation of the results obtained with the proposed algorithm is performed by exploiting the OpenQASM programming language through the Qiskit Aer *QasmSimulator*, thanks to which we built a noiseless realization of the QMETTS procedure: this simulator runs ideal gate operations but realistic quantum measurements over perfectly generated, yet inaccessible, state vectors,

so that observables are computed by relying on measurements of Pauli bitstrings performed as explained in Section 1.2.3.
More specifically, each needed Pauli bitstring measure in Eq. (3.86) is obtained by running the corresponding circuit $N_S = 10^5$ times in order to collect enough statistics (see Eqs. (1.43) and (1.44)). From these measurements, observables are estimated by proceeding as described in Section 3.4.2 (see Eqs. (3.89) and (3.90)), generating for each needed observable measure $M = 10$ samples. As we did for the Qiskit Aer *StatevectorSimulator* simulations, we performed different implementations, exploring the integer values of γ in the range $[1, 10]$ and dividing the thermal evolution from $\beta = 0$ to $\beta = 10$ in 20 steps. The QITE algorithm is implemented by exploiting the step merge procedure (see Section 3.4.4) and the needed Givens angles are computed by sampling them $N_\theta = 10^4$ times (see Eqs. (3.100) and (3.101)).
Furthermore, in order to show plots of observable estimates, the following procedure is followed: each run producing the desired observables is performed $N_R = 15$ times, varying for each of them the simulator seed; then 1000 samples are taken from each of these 15 estimates originating from different runs and the values corresponding to the 15, 50 and 85 percentiles of all the distribution are collected, with the caveat that the samples for strictly positive quantities (i.e. purity and mutual information) are kept only if they are positive, otherwise they are resampled.
In Figs. 3.10, 3.11, 3.12 and 3.13 I report the internal energy $U_{\mathbf{qa}}(\beta)$, its relative error $\Delta U_{\text{sv-qa}}(\beta)$ with respect to the values obtained from Qiskit Aer *StatevectorSimulator* with the step merge approximation, the single-qubit purities and the mutual information for all the qubit couples, respectively. To favor readability, I decided to report only $\gamma \in \{2, 4, 7, 10\}$ values: images with all the computed γ values are reported in Appendix A in Figs. A.1, A.2, A.3 and A.4. In Fig. 3.14 I reported as an example all the 15 runs obtained by changing the Qiskit Aer *QasmSimulator* seed for the simulation of the system with $\gamma = 10$.

These results show a good level of agreement with the expectations described in previous Section 3.5.2: the two different behaviors arising from the distinct γ-dependent system ground states $|\psi\rangle_{g1}$ and $|\psi\rangle_{g2}$ (see Eqs. (3.111), (3.112)) are clearly evident in the purity and mutual information plots (Figs. 3.12 and 3.13), well recognizable despite the predicted uncertainty. Energy levels are also correctly extracted for about all the explored γ values, where the relative error with respect to the Qiskit Aer *StatevectorSimulator* predictions settles at about 10^{-2} for the higher γ values, while maintaining a decreasing behavior for the smaller ones.

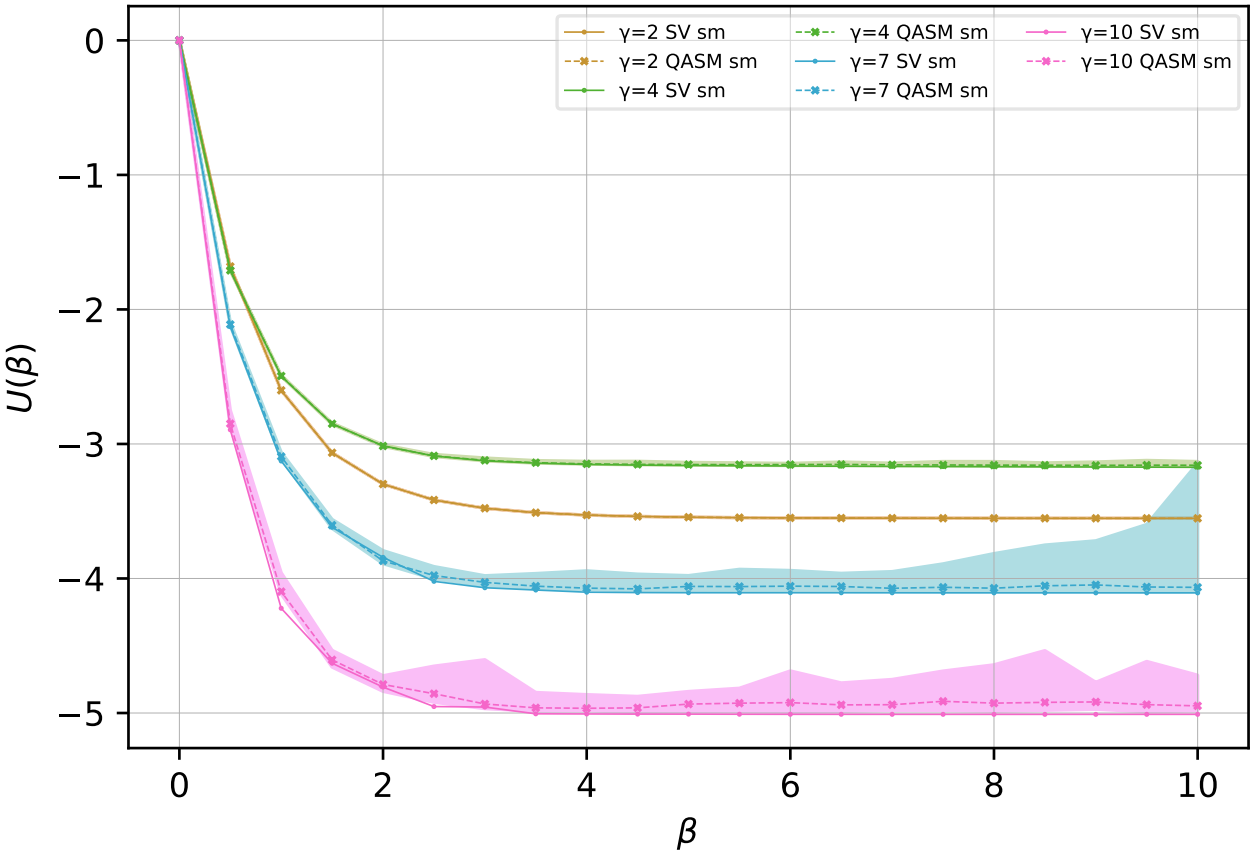


Figure 3.10: Colored plot showing the estimated system energy expectation values, as a function of the inverse temperature β. Different colors correspond to different $\gamma \in \{2, 4, 7, 10\}$ values. Solid lines with dot markers are the values computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the QMETTS algorithm, including the step merge approximation. Dashed lines with 'X' markers are the 50th percentile values (see Section 3.5.3) computed with Qiskit Aer *QasmSimulator*, corresponding to the noiseless implementation of the QMETTS algorithm, including the step merge approximation. Shaded regions correspond to the values between the 85th and 15th percentiles.

3.5.4 Noisy simulations

In order to get a more representative estimate of the algorithm behavior on real hardware and observe the expected influence of noise, we run the simulation on a Qiskit *fake backend*, this consisting in a simulator built so that it mimics the

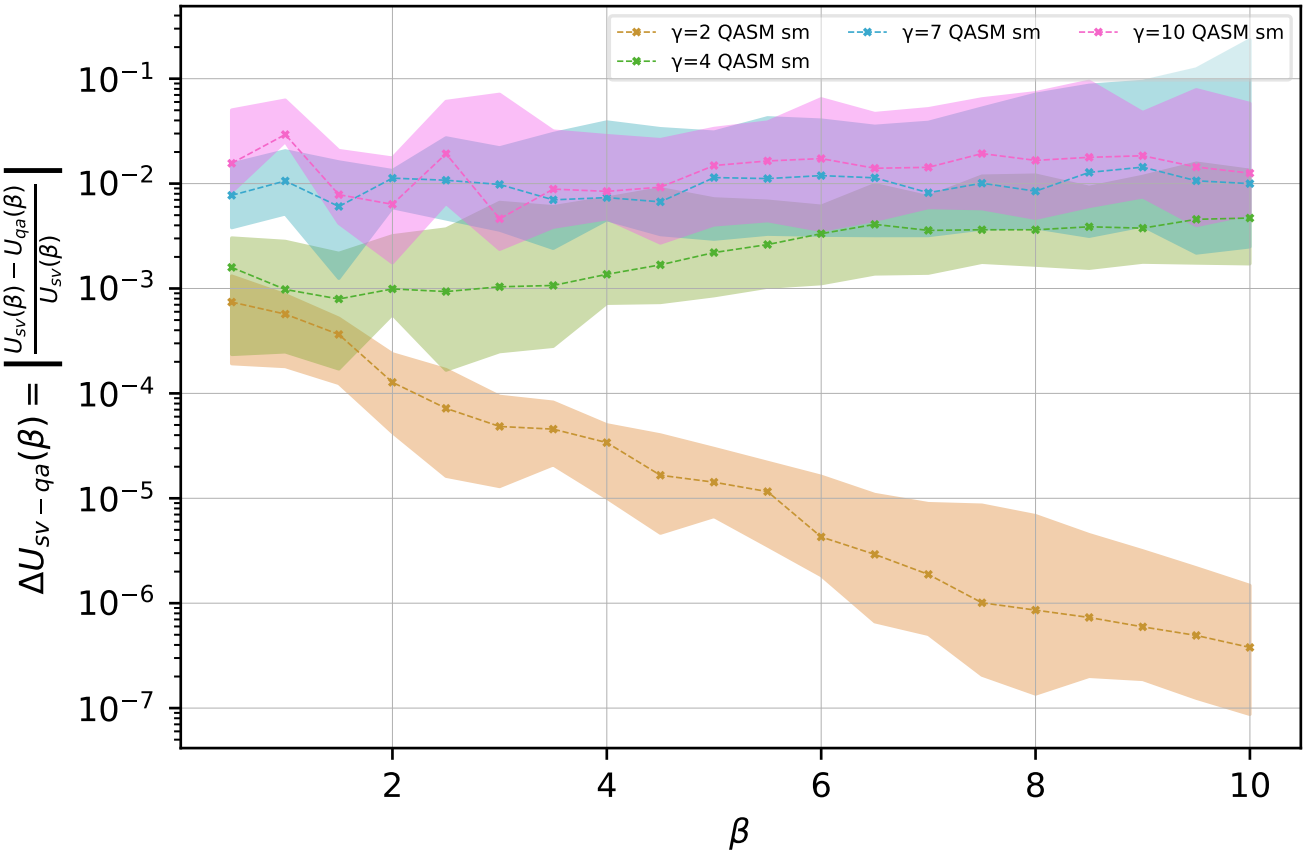


Figure 3.11: Colored plot showing the relative distance between the estimated system energy obtained from the Qiskit Aer *StatevectorSimulator* and *QasmSimulator*, as a function of the inverse temperature β. Different colors correspond to different $\gamma \in \{2, 4, 7, 10\}$ values. The reference values are those computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the QMETTS algorithm, including the step merge approximation. Dashed lines with 'X' markers are the relative errors of the 50th percentile energy values (see Section 3.5.3) computed with Qiskit Aer *QasmSimulator*, corresponding to the noiseless implementation of the QMETTS algorithm, including the step merge approximation. Shaded regions correspond to the values between the 85th and 15th percentiles of the energy relative error.

snapshot of a specific hardware: such kind of simulators are equipped with lots of information, including noise parameters, qubit coupling maps and native gates, which are included in the QASM simulation and effectively modify the ideal instructions so that the implemented operations resemble more the not-ideal ones

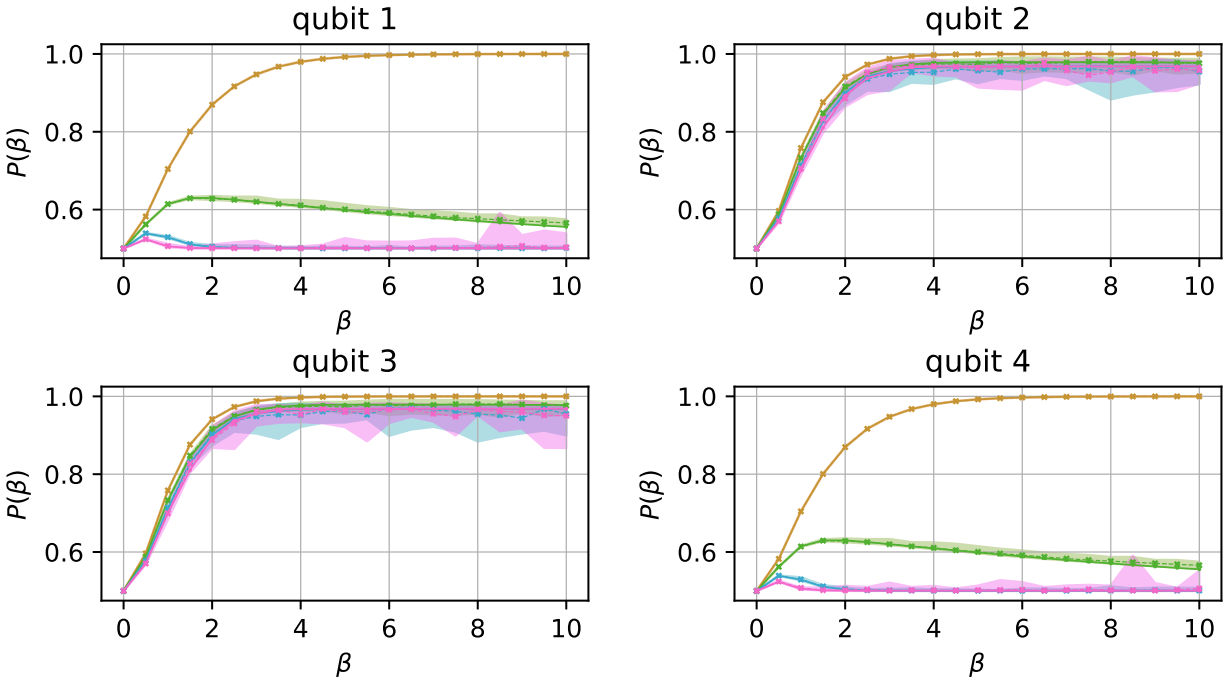


Figure 3.12: Colored plot showing the purities expectation values of each qubit in the system, as a function of the inverse temperature β, qubits 1 and 4 being the external ones while qubits 2 and 3 being the internal ones. Different colors correspond to different $\gamma \in \{2, 4, 7, 10\}$ values. Solid lines with dot markers are the values computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the QMETTS algorithm, including the step merge approximation. Dashed lines with 'X' markers are the 50th percentile values (see Section 3.5.3) computed with Qiskit Aer *QasmSimulator*, corresponding to the noiseless implementation of the QMETTS algorithm, including the step merge approximation. Shaded regions correspond to the values between the 85th and 15th percentiles. Legend is the same as in Fig. 3.10, omitted here to favor readability.

usually performed on real devices. More specifically, we run the simulations on a Qiskit Aer simulator built from the 7 qubit *FakeJakartaV2* Qiskit fake backend. To suppress the effects of the introduced noise, we added two error mitigation steps to the algorithm: first of all we patched all the executed quantum circuits with dynamical decoupling sequences, then we also performed read-out mitigation after the measurement step, in order to correct for possible errors.

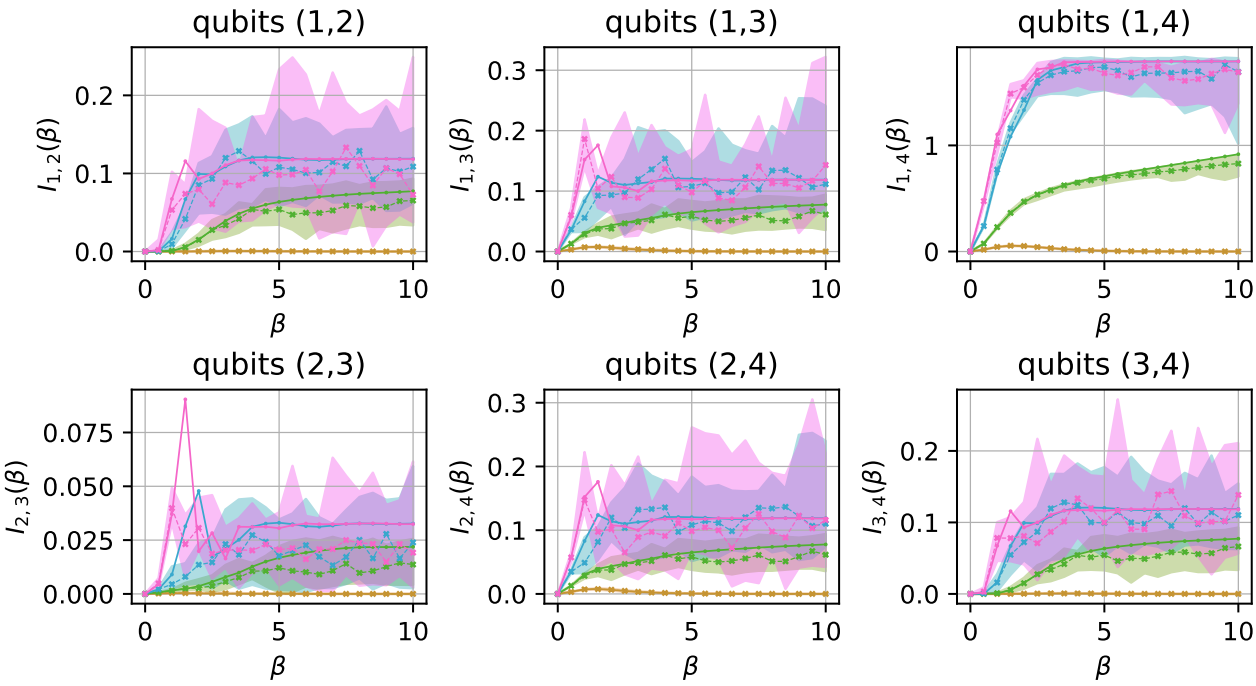


Figure 3.13: Colored plot showing the mutual information expectation values of each qubit pair in the system, as a function of the inverse temperature β, qubits 1 and 4 being the external ones while qubits 2 and 3 being the internal ones. Different colors correspond to different $\gamma \in \{2, 4, 7, 10\}$ values. Solid lines with dot markers are the values computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the QMETTS algorithm, including the step merge approximation. Dashed lines with 'X' markers are the 50th percentile values (see Section 3.5.3) computed with Qiskit Aer *QasmSimulator*, corresponding to the noiseless implementation of the QMETTS algorithm, including the step merge approximation. Shaded regions correspond to the values between the 85th and 15th percentiles. Legend is the same as in Fig. 3.10, omitted here to favor readability.

Dynamical decoupling Dynamical decoupling [26, 27] is a very common error mitigation technique that aims at reducing the decoherence effects due to the coupling with the environment. The idea is to act with a sequence of rapid qubit state flips on idling particles, with the intent of averaging out the slow-noise components.
Several different sequences of operations may be chosen, depending on the characteristics of the noise to suppress, some of the most common being the Carr-Purcell, the Carr-Purcell-Meiboom-Gill (CPMG) or the Uhrig schemes. Since we do not intend to perform a detailed analysis of the effects of such a technique, we decided

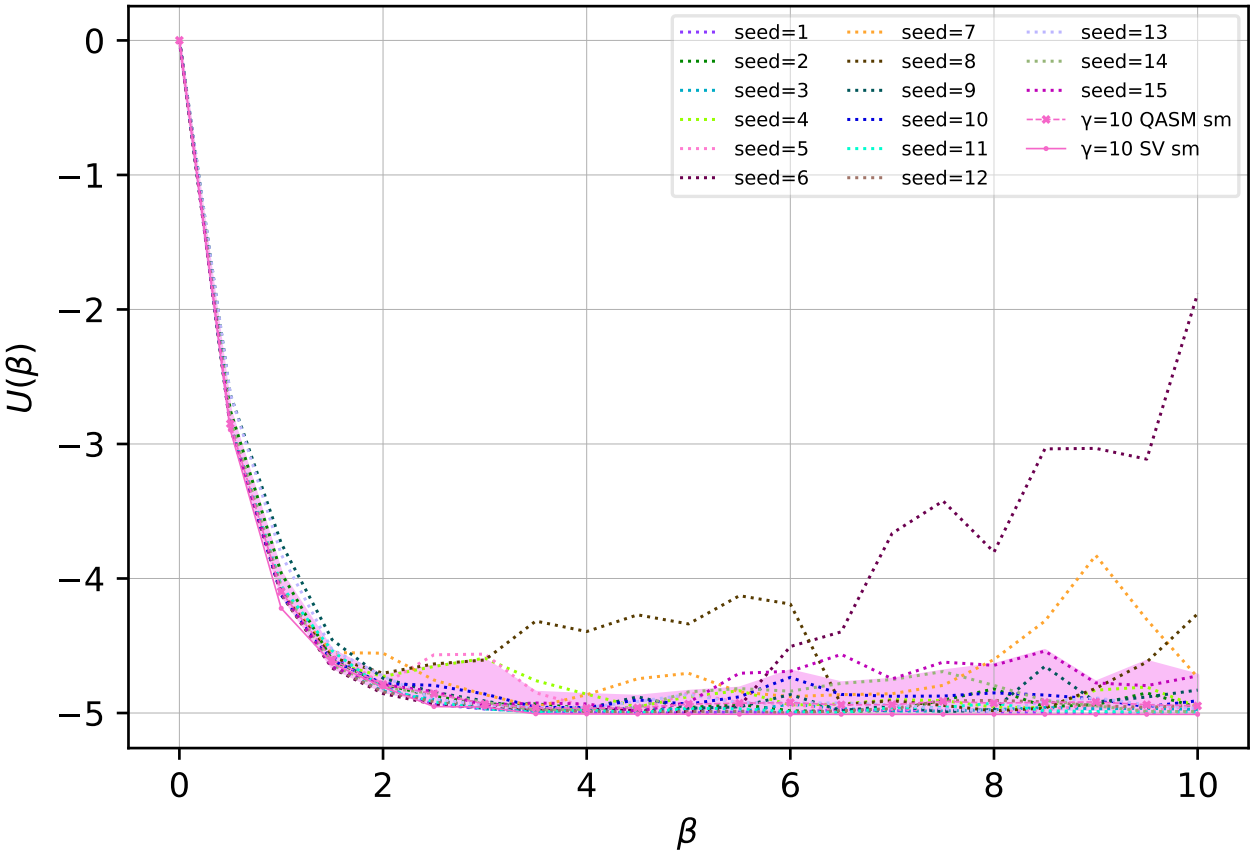


Figure 3.14: Colored plot showing the estimated system energy expectation values for the $\gamma = 10$ simulation, as a function of the inverse temperature β. Dotted lines in different colors are different realizations obtained by changing the Qiskit Aer *QasmSimulator* seed. The solid line with dot markers is the reference simulation computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the QMETTS algorithm, including the step merge approximation. The dashed line with 'X' markers represents the 50th percentile values (see Section 3.5.3) computed with Qiskit Aer *QasmSimulator*, corresponding to the noiseless implementation of the QMETTS algorithm, including the step merge approximation. The shaded region corresponds to the values between the 85th and 15th percentiles.

to simply divide every idle qubit time in four blocks, adding an $\hat{X}$ gate at the center of every block. A visual representation of this implementation on a generic circuit is given in Figs. 3.15 and 3.16.

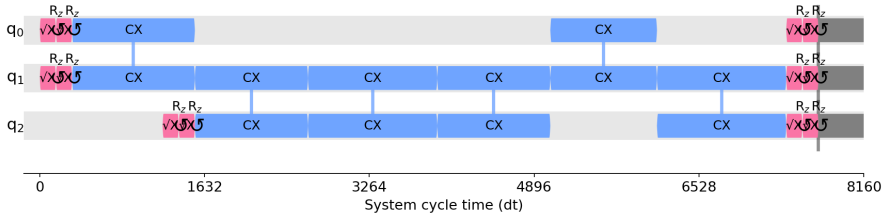


Figure 3.15: Generic quantum circuits of three qubits, where the sequence of operations on every qubit is represented as a function of time expressed in system cycles. Different colors correspond to different gates. Top and bottom qubits remain idle for some periods, during which they may be affected by decoherence due to coupling with the environment.

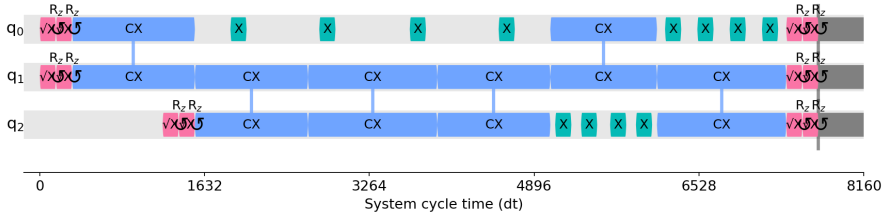


Figure 3.16: Same circuit of Fig. 3.15, but now with dynamical decoupling: idle times are divided into four identical intervals, at the center of which an $\hat{X}$ pulse is added (represented in green), to compensate for the noise influence.

Readout error mitigation When performing a measurement on a real quantum device, this may be subject to a certain error, which effect is that of modifying the correct output bitstring, thereby affecting the collected probabilities. This can be modeled with a matrix $\mathbf{A}$ multiplying the vector $\vec{p}_{id}$ with the ideal probabilities and producing the noisy measured probabilities $\vec{p}_{me}$

$$\vec{p}_{me} = \mathbf{A}\,\vec{p}_{id}. \tag{3.113}$$

To be able to correct $\vec{p}_{me}$ and retrieve the right values, one should apply the inverse matrix $\mathbf{A}^{-1}$ to the vector of measurements. In principle the matrix $\mathbf{A}$ may be built by preparing and measuring all the possible basis states, which is extremely demanding for bigger systems though, since the number of needed measurements scales exponentially with the number of qubits. One then relies on different approximations which allows to approximate $\mathbf{A}$ and to perform a partial correction on the measurement outcomes.
Also in this case, we do not focus on the details of the procedure, and we exploit the basic functionalities of the IBM *Mthree* software to perform this step.

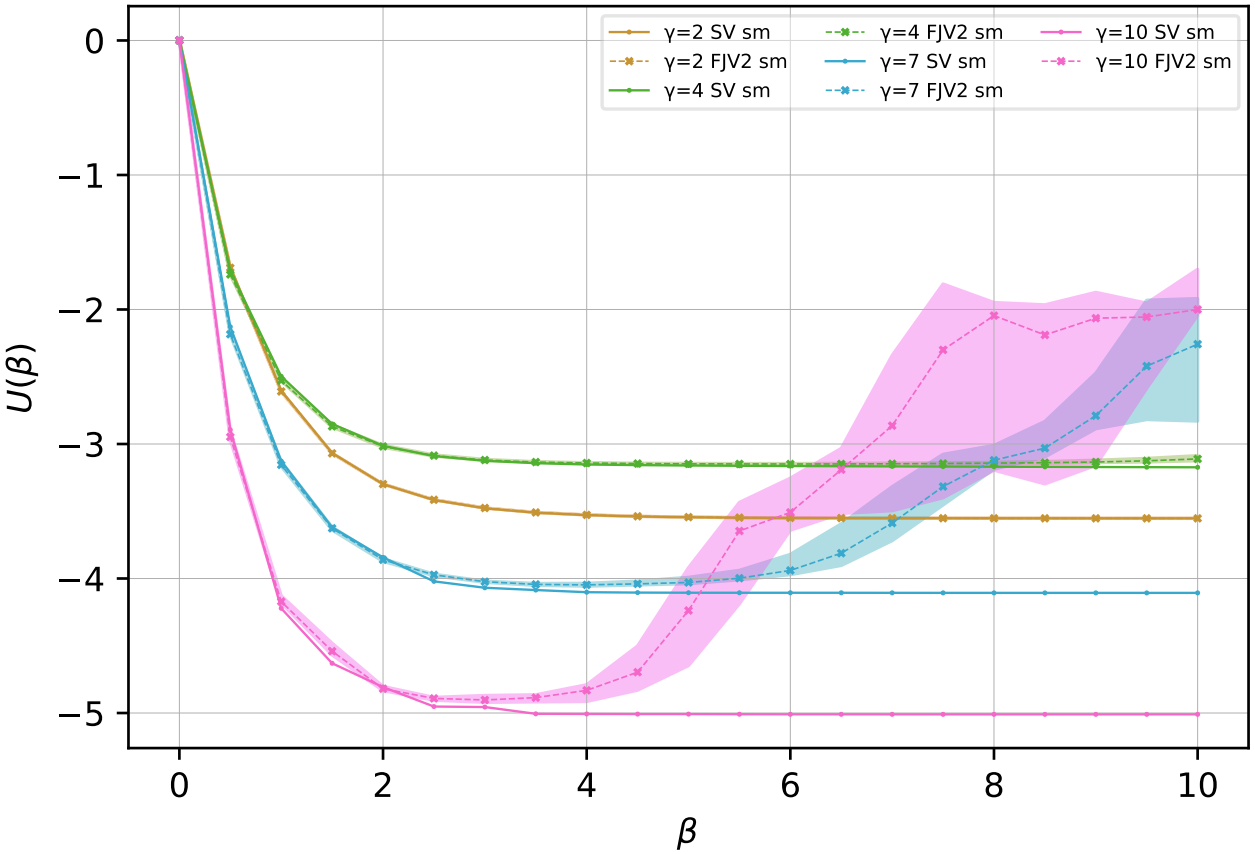


Figure 3.17: Colored plot showing the estimated system energy expectation values, as a function of the inverse temperature β. Different colors correspond to different $\gamma \in \{2, 4, 7, 10\}$ values. Solid lines with dot markers are the values computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the QMETTS algorithm, including the step merge approximation. Dashed lines with 'X' markers are the 50th percentile values (see Section 3.5.3) computed with Qiskit Aer simulator built from the 7 qubit *FakeJakartaV2* Qiskit fake backend, corresponding to the noiseless implementation of the QMETTS algorithm, including the step merge approximation. Shaded regions correspond to the values between the 85th and 15th percentiles.

Noisy simulations results

We repeated the same simulations described in the previous section for the Qiskit Aer *QasmSimulator* (see Section 3.5.3), keeping the same parameter value choices and only changing the Qiskit simulator by building it from the 7 qubit *FakeJakar-*

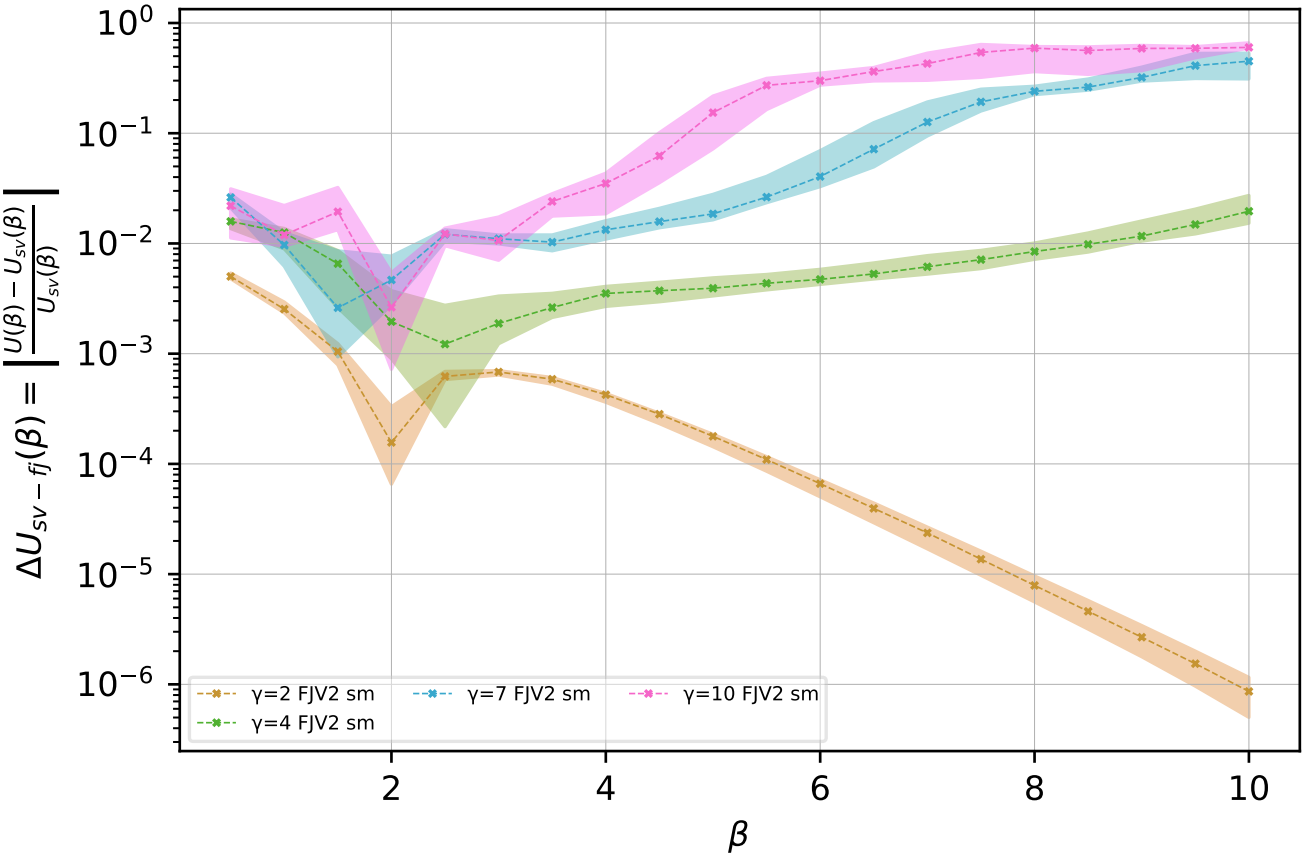


Figure 3.18: Colored plot showing the relative distance between the estimated system energy obtained from the Qiskit Aer *StatevectorSimulator* and *QasmSimulator*, as a function of the inverse temperature β. Different colors correspond to different $\gamma \in \{2, 4, 7, 10\}$ values. The reference values are those computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the QMETTS algorithm, including the step merge approximation. Dashed lines with 'X' markers are the relative errors of the 50th percentile energy values (see Section 3.5.3) computed with Qiskit Aer simulator built from the 7 qubit *FakeJakartaV2* Qiskit fake backend, corresponding to the noiseless implementation of the QMETTS algorithm, including the step merge approximation. Shaded regions correspond to the values between the 85th and 15th percentiles of the energy relative error.

taV2 Qiskit fake backend. The only difference is that we decided to set the number of different seeds run for each simulation generating an observable thermal expectation value estimate to $N_R = 10$ rather than keeping $N_R = 15$: this choice was

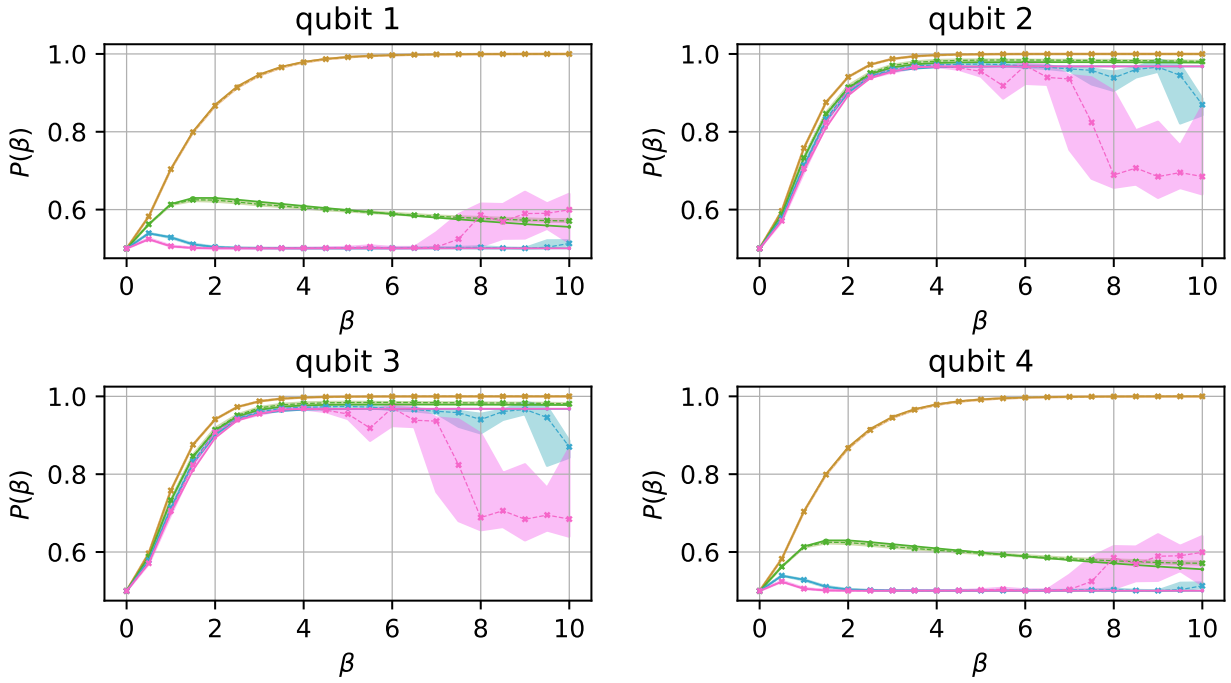


Figure 3.19: Colored plot showing the purities expectation values of each qubit in the system, as a function of the inverse temperature β, qubits 1 and 4 being the external ones while qubits 2 and 3 being the internal ones. Different colors correspond to different $\gamma \in \{2, 4, 7, 10\}$ values. Solid lines with dot markers are the values computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the QMETTS algorithm, including the step merge approximation. Dashed lines with 'X' markers are the 50th percentile values (see Section 3.5.3) computed with Qiskit Aer simulator built from the 7 qubit *FakeJakartaV2* Qiskit fake backend, corresponding to the noiseless implementation of the QMETTS algorithm, including the step merge approximation. Shaded regions correspond to the values between the 85th and 15th percentiles. Legend is the same as in Fig. 3.17, omitted here to favor readability.

motivated by the smaller variance observed on the observable expectation values estimates.

Results are reported in Figs. 3.17, 3.18, 3.19 and 3.20 for the internal energy, its relative error with respect to the values obtained from Qiskit Aer *StatevectorSimulator* with the step merge approximation, the single-qubit purities and the mutual information for all the qubit couples, respectively. As done for the noiseless case, to favor readability, I decided to report only $\gamma \in \{2, 4, 7, 10\}$ values: images with all the computed γ values are reported in Appendix A in Figs. A.5, A.6, A.7 and A.8. In Fig. 3.21 I reported as an example all the 10 runs obtained by chang-

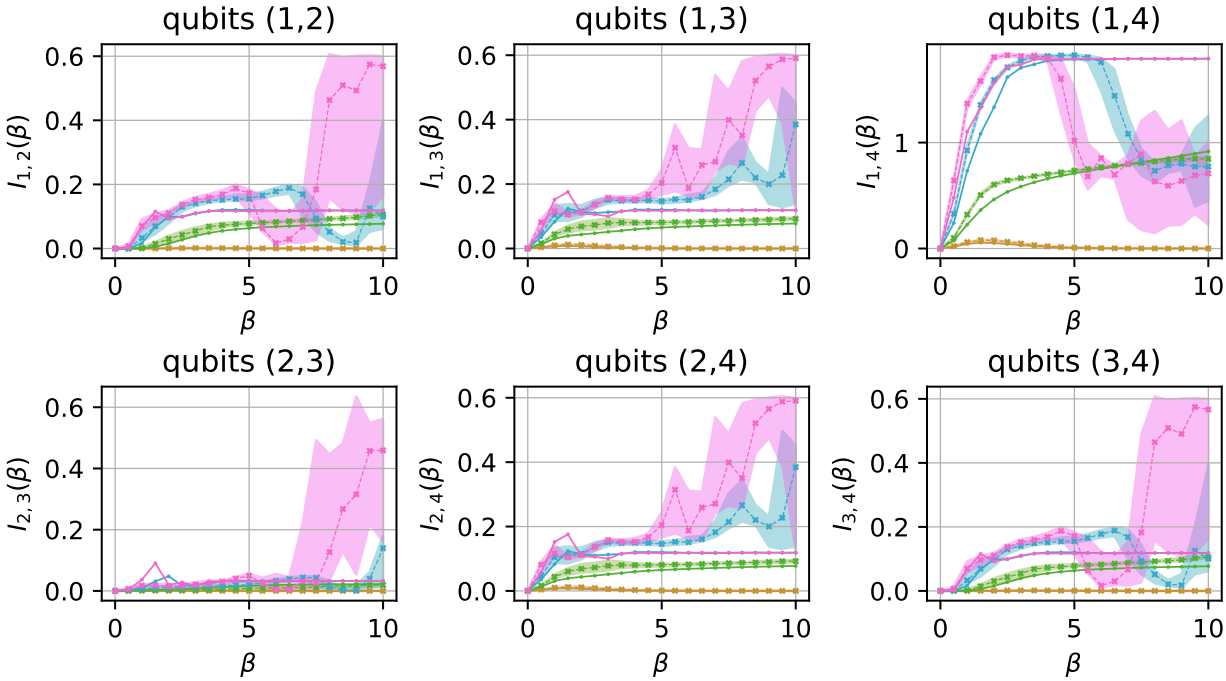


Figure 3.20: Colored plot showing the mutual information expectation values of each qubit pair in the system, as a function of the inverse temperature β, qubits 1 and 4 being the external ones while qubits 2 and 3 being the internal ones. Different colors correspond to different $\gamma \in \{2, 4, 7, 10\}$ values. Solid lines with dot markers are the values computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the QMETTS algorithm, including the step merge approximation. Dashed lines with 'X' markers are the 50th percentile values (see Section 3.5.3) computed with the Qiskit Aer simulator built from the 7 qubit *FakeJakartaV2* Qiskit fake backend, corresponding to the noisy implementation of the QMETTS algorithm, including the step merge approximation. Shaded regions correspond to the values between the 85th and 15th percentiles. Legend is the same as in Fig. 3.17, omitted here to favor readability.

ing the seed of the Qiskit simulator built from the 7 qubit *FakeJakartaV2* Qiskit fake backend for the simulation of the system with $\gamma = 10$.

All these noisy simulations start by closely following the predictions computed with the Qiskit Aer *StatevectorSimulator*, but depending on the specific value γ, they break down after a critical inverse temperature β is reached, greatly and rapidly drifting away from the expectations. Before this effect is triggered, all quantities are correctly behaving as expected, showing the two regimes presented in Section 3.5.2. At higher β values, a new trend is induced, apparently common

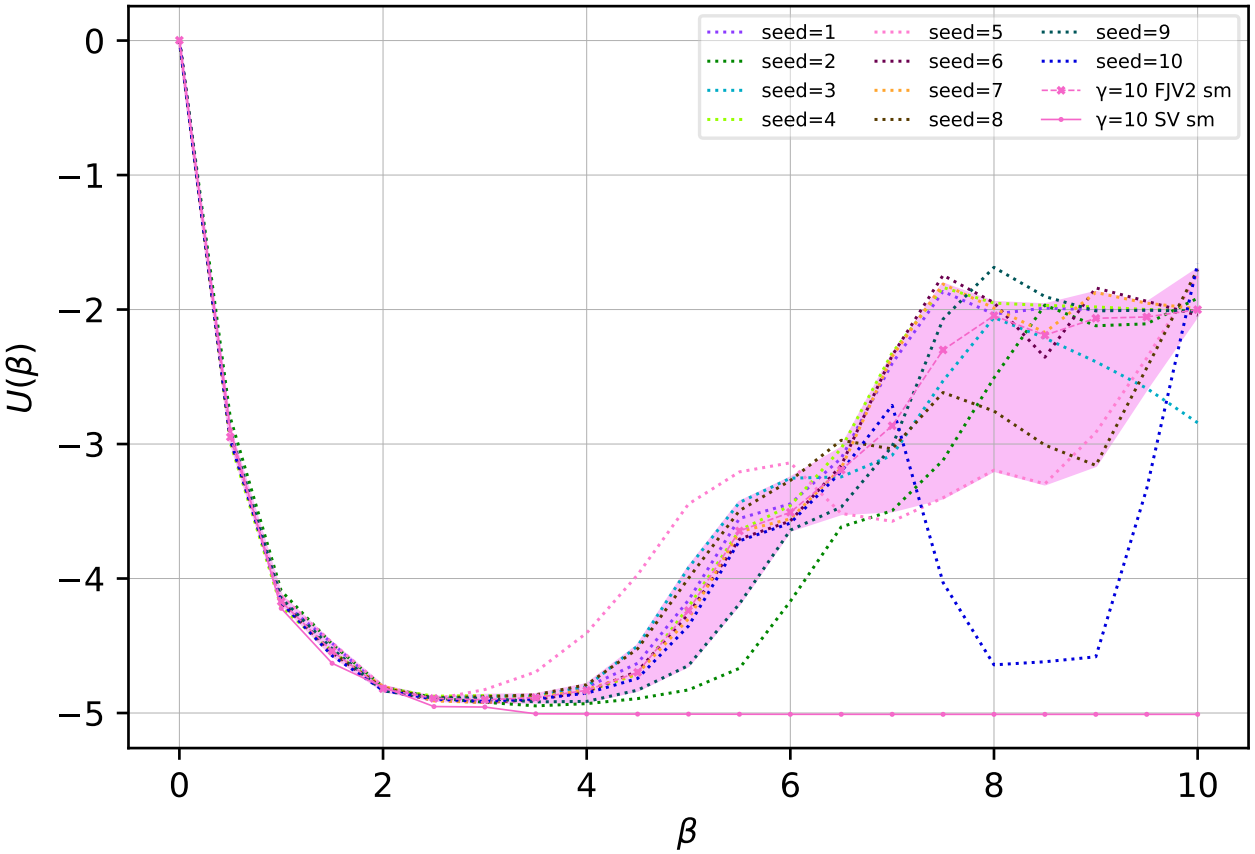


Figure 3.21: Colored plot showing the estimated system energy expectation values for the $\gamma = 10$ simulation, as a function of the inverse temperature β. Dotted lines in different colors are different realizations obtained by changing the Qiskit Aer simulator seed. The solid line with dot markers is the reference simulation computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the QMETTS algorithm, including the step merge approximation. The dashed line with 'X' markers represents the 50th percentile values (see Section 3.5.3) computed with the Qiskit Aer simulator built from the 7 qubit *FakeJakartaV2* Qiskit fake backend, corresponding to the noisy implementation of the QMETTS algorithm, including the step merge approximation. The shaded region corresponds to the values between the 85th and 15th percentiles.

to all the realizations with $\gamma \geq 4$: this behavior is not yet well understood and we reserve to clarify on it in future works.

Conclusions

In this thesis, I address some central topics in the context of quantum computing and many-body physics, in particular following an approach based on the digital, or circuit-based, quantum computing paradigm.
After introducing the basic concepts of this paradigm in Chapter 1, I move on to the description of two main different projects.
The first one, presented in Chapter 2, has a more mathematical and functional nature and consists of the proposal of a novel approach to the representation of quantum operators and observables, which draws inspiration from the properties of the Hubbard-Stratonovich transformation [13, 14]. This is an exact mathematical identity, well known in the context of statistical physics, that helps in reducing the complexity of exponentials by introducing a coupling with an external auxiliary real scalar field. Through this transformation, both real and imaginary-time evolution operators generated by squared Hermitian operators are decomposed as an integral over a Gaussian distributed continuous set of new exponentials, which depend only linearly on the original generators. This may often lead, in principle, to a decoupling of different terms, possibly simplifying the implementation of the single elements of the set. The integral may then be approximately realized by relying on a Monte Carlo formulation. I argue that while this method appears impractical for a direct implementation of quantum gates, it may still prove useful as an alternative technique for measuring observable expectation values. The primary issue of this approach is that the number of required resources typically scales exponentially with the dimensions of the considered system, strongly reducing the range of applicability of such a technique. Despite the proposed algorithm not being particularly optimal or efficient, it still offers a new perspective from which quantum operations may be approached, and could be exploited for test cases, smaller systems, or specific settings where symmetry properties play a key role in reducing the system's degrees of freedom. Relevant directions for future studies include a more focused analysis aimed at finding the most useful application of the proposed formulation and a quantitative comparison of the performances with the main alternative approaches.
The second presented project, described in Chapter 3, is more oriented to specific

physical systems simulation and consists of an algorithm for the estimation of few-body observables thermal expectation values on many-particle systems characterized by all-to-all, number-conserving interactions. The basic idea is that of relying on the QMETTS [16] algorithm for performing the approximation of the expectation values, where the ansatz to the QITE [16] step is built on a set of Givens rotations. These are known to be universal for representing number-conserving unitaries with a quadratic resource cost in the number of particles, which would therefore extend the applicability of the QMETTS algorithm to a wider range of systems than finite-range k-local Hamiltonian systems. The algorithm is applied as a test case to a small system of four neutrino flavors in the two-flavor approximation, which is very easily mapped onto a system of four qubits. The neutrino flavor Hamiltonian is notably characterized by long-range, all-to-all interactions, which give rise to distinctive phenomena such as flavor oscillations and chaotic dynamics. These features drive the system toward rapid thermalization, thereby making it amenable to a thermal description. The contained system dimensions and the exploitation of its symmetry properties allow, in this simple application, for a complete reconstruction of all the two-body reduced density matrices, but the method is more generally suitable to be applied to bigger systems by relying on an efficient sampling procedure which approximates the complete system description. In this work, the algorithm's theoretical ideal predictions are computed and a simulation of its implementation is performed through the Qiskit software. Results are promising, but noise seems to have a strong impact on the performances, which suggests that future developments should include some efforts into building noise resilience and more targeted error mitigation steps. A more quantitative analysis of the actual performance of the proposed algorithm and the specific impact of the exploited approximations (e.g. the step merge procedure) is also left to future development. The final aim is to perform real simulations on existing quantum hardware and quantitatively comparing the performances with other alternative techniques, both quantum and quantum-classical hybrid.

Appendix A

Complete neutrino results

In Sections 3.5.3 and 3.5.4, I presented only the neutrino simulation results computed for a reduced subset of the explored range of γ values, to allow for better readability. For completeness, I hereby report the plots including all performed computations, considering all integer values in the range $[1, 10]$. Simulation parameters are described in Section 3.5.3.

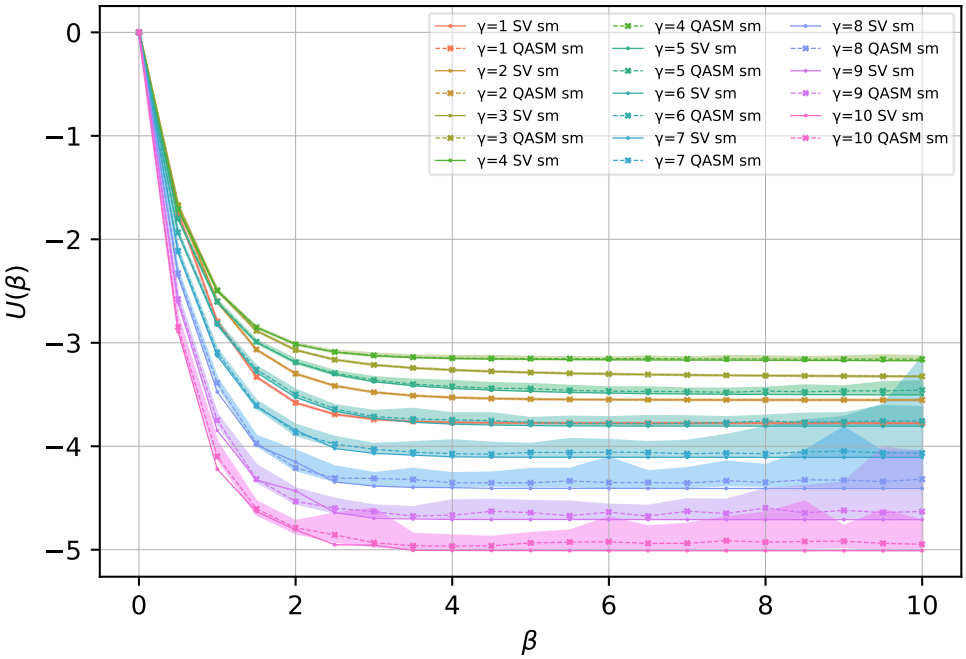


Figure A.1: Colored plot showing the estimated system energy expectation values, as a function of the inverse temperature β. Different colors correspond to different $\gamma \in [1, 10] \cap \mathbb{Z}$ values. Solid lines with dot markers are the values computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the QMETTS algorithm, including the step merge approximation. Dashed lines with 'X' markers are the 50th percentile values (see Section 3.5.3) computed with Qiskit Aer *QasmSimulator*, corresponding to the noiseless implementation of the QMETTS algorithm, including the step merge approximation. Shaded regions correspond to the values between the 85th and 15th percentiles.

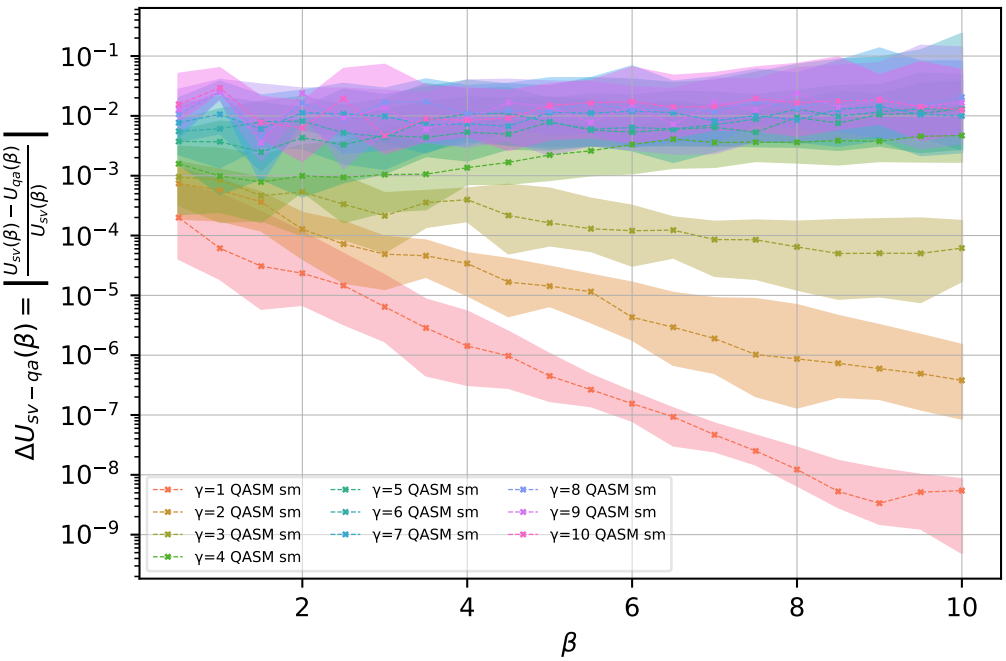


Figure A.2: Colored plot showing the relative distance between the estimated system energy obtained from the Qiskit Aer *StatevectorSimulator* and *QasmSimulator*, as a function of the inverse temperature β. Different colors correspond to different $\gamma \in [1, 10] \cap \mathbb{Z}$ values. The reference values are those computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the QMETTS algorithm, including the step merge approximation. Dashed lines with 'X' markers are the relative errors of the 50th percentile energy values (see Section 3.5.3) computed with Qiskit Aer *QasmSimulator*, corresponding to the noiseless implementation of the QMETTS algorithm, including the step merge approximation. Shaded regions correspond to the values between the 85th and 15th percentiles of the energy relative error.

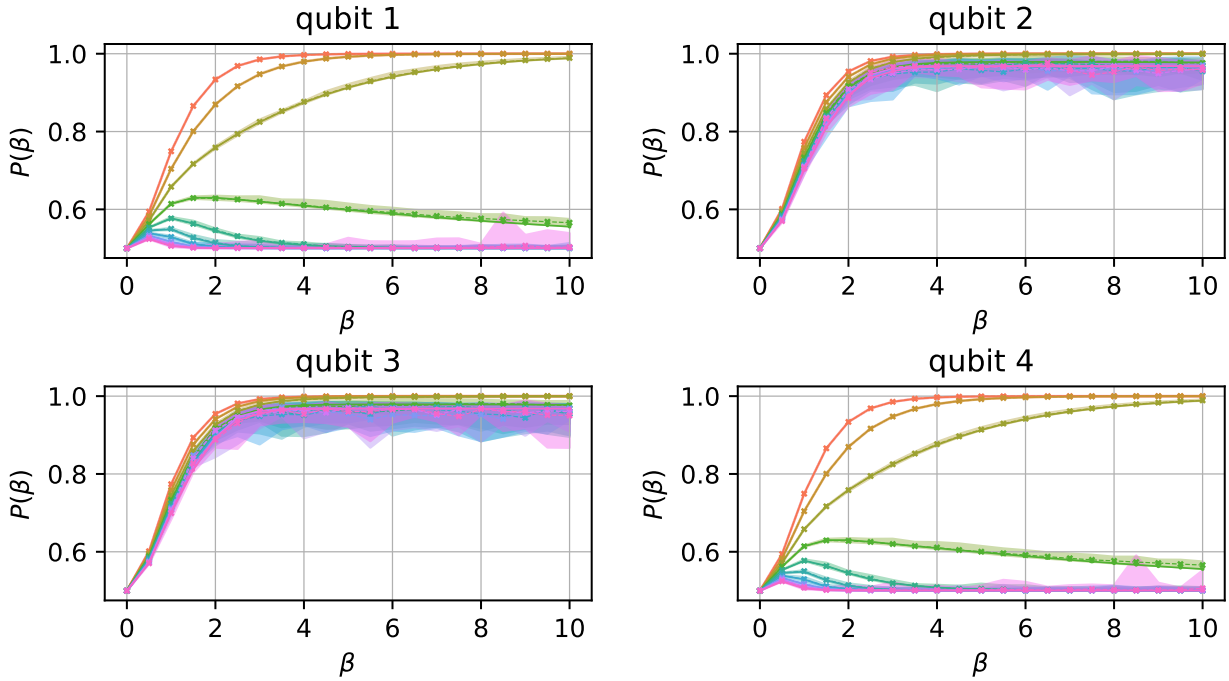


Figure A.3: Colored plot showing the purities expectation values of each qubit in the system, as a function of the inverse temperature β, qubits 1 and 4 being the external ones while qubits 2 and 3 being the internal ones. Different colors correspond to different $\gamma \in [1, 10] \cap \mathbb{Z}$ values. Solid lines with dot markers are the values computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the QMETTS algorithm, including the step merge approximation. Dashed lines with 'X' markers are the 50th percentile values (see Section 3.5.3) computed with Qiskit Aer *QasmSimulator*, corresponding to the noiseless implementation of the QMETTS algorithm, including the step merge approximation. Shaded regions correspond to the values between the 85th and 15th percentiles. Legend is the same as in Fig. A.1, omitted here to favor readability.

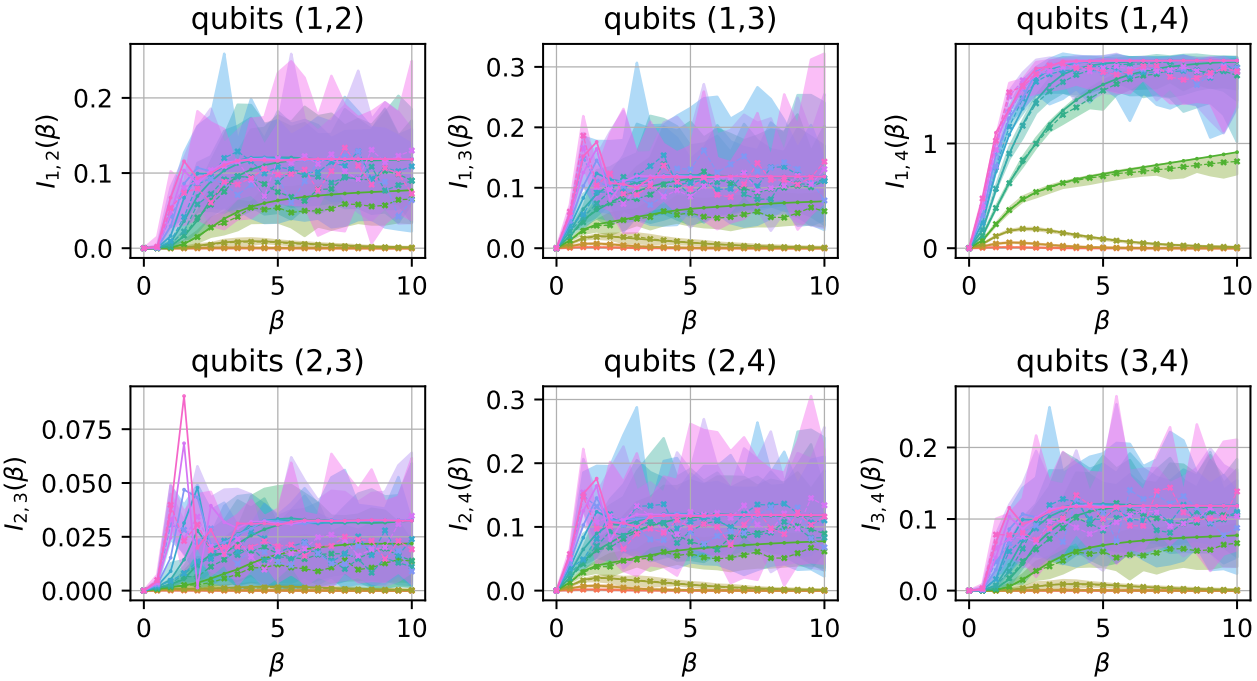


Figure A.4: Colored plot showing the mutual information expectation values of each qubit pair in the system, as a function of the inverse temperature β, qubits 1 and 4 being the external ones while qubits 2 and 3 being the internal ones. Different colors correspond to different $\gamma \in [1, 10] \cap \mathbb{Z}$ values. Solid lines with dot markers are the values computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the QMETTS algorithm, including the step merge approximation. Dashed lines with 'X' markers are the 50th percentile values (see Section 3.5.3) computed with Qiskit Aer *QasmSimulator*, corresponding to the noiseless implementation of the QMETTS algorithm, including the step merge approximation. Shaded regions correspond to the values between the 85th and 15th percentiles. Legend is the same as in Fig. A.1, omitted here to favor readability.

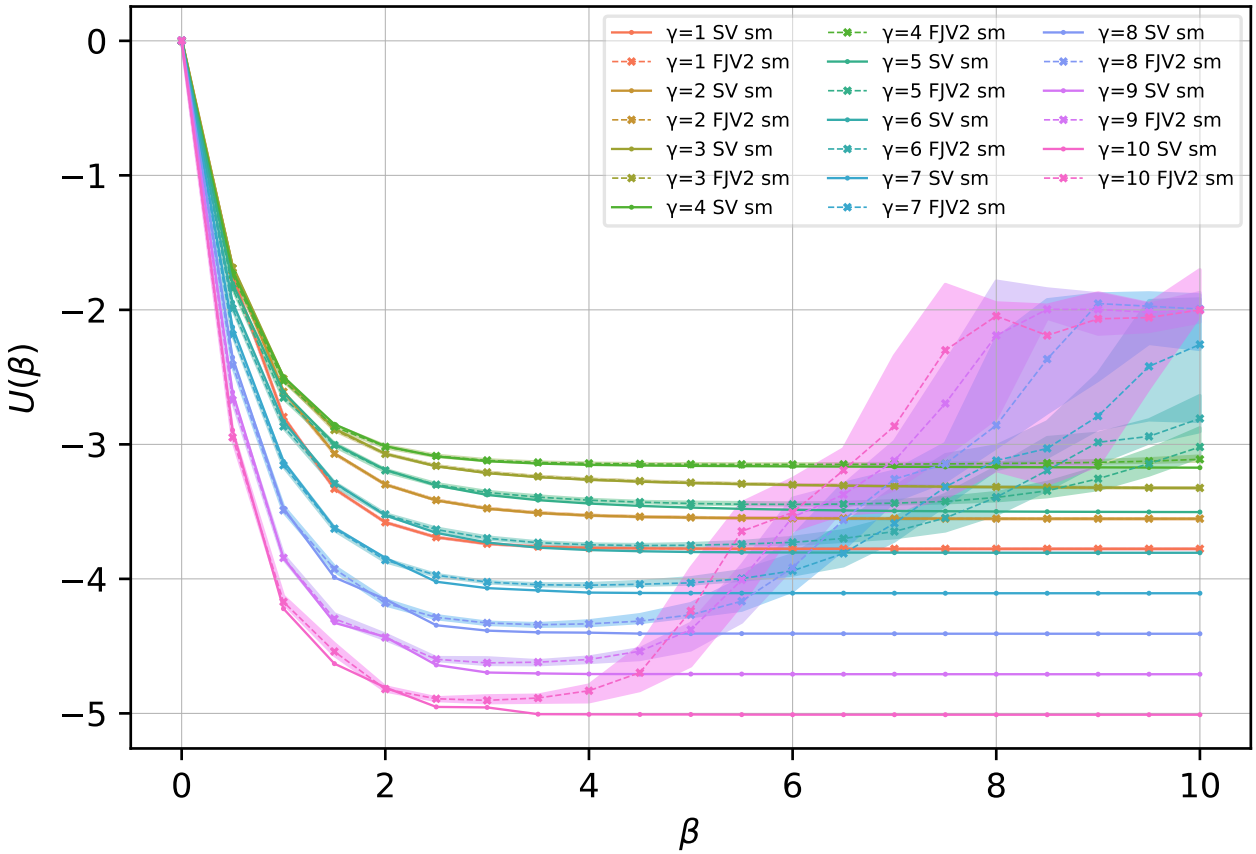


Figure A.5: Colored plot showing the estimated system energy expectation values, as a function of the inverse temperature β. Different colors correspond to different $\gamma \in [1, 10] \cap \mathbb{Z}$ values. Solid lines with dot markers are the values computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the QMETTS algorithm, including the step merge approximation. Dashed lines with 'X' markers are the 50th percentile values (see Section 3.5.3) computed with Qiskit Aer simulator built from the 7 qubit *FakeJakartaV2* Qiskit fake backend, corresponding to the noiseless implementation of the QMETTS algorithm, including the step merge approximation. Shaded regions correspond to the values between the 85th and 15th percentiles.

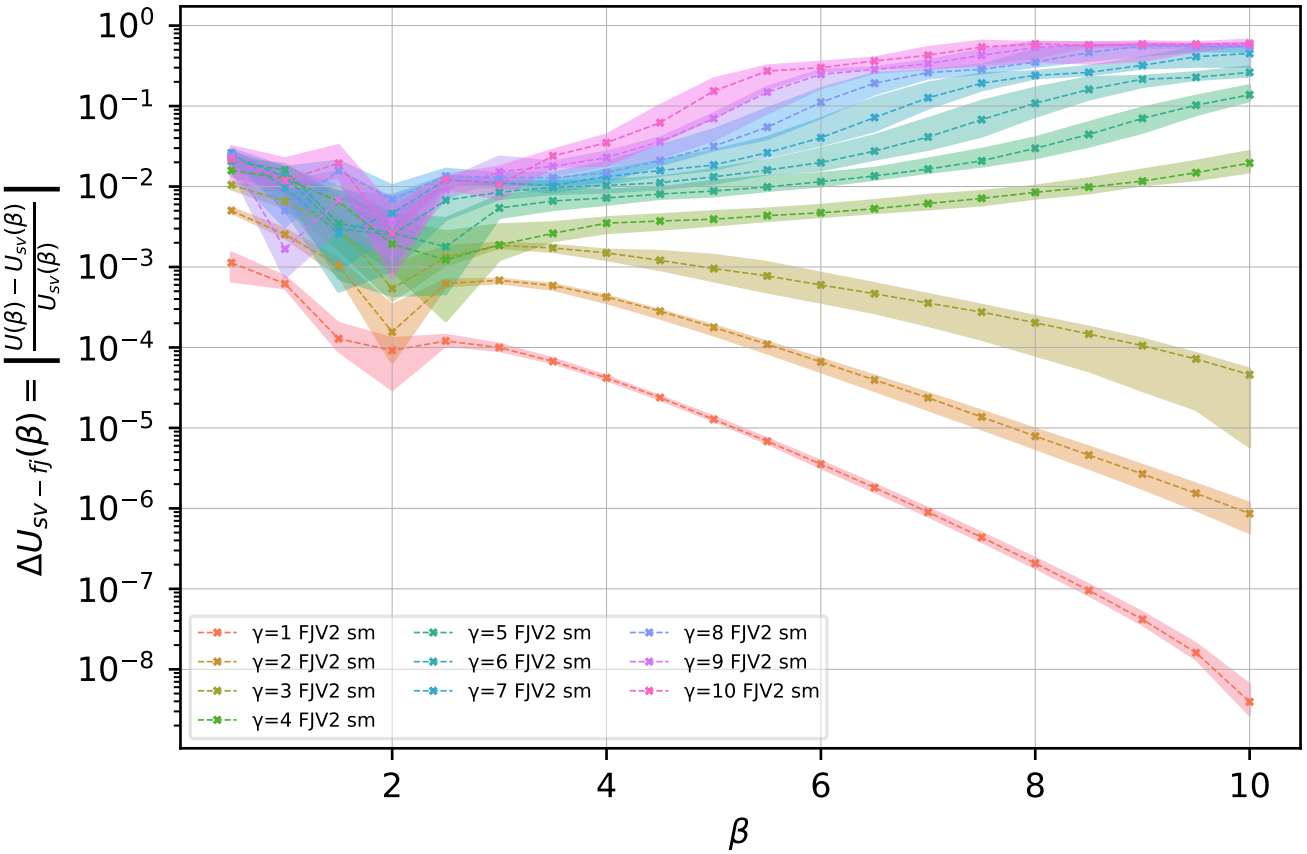


Figure A.6: Colored plot showing the relative distance between the estimated system energy obtained from the Qiskit Aer *StatevectorSimulator* and *QasmSimulator*, as a function of the inverse temperature β. Different colors correspond to different $\gamma \in [1, 10] \cap \mathbb{Z}$ values. The reference values are those computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the QMETTS algorithm, including the step merge approximation. Dashed lines with 'X' markers are the relative errors of the 50th percentile energy values (see Section 3.5.3) computed with Qiskit Aer simulator built from the 7 qubit *FakeJakartaV2* Qiskit fake backend, corresponding to the noiseless implementation of the QMETTS algorithm, including the step merge approximation. Shaded regions correspond to the values between the 85th and 15th percentiles of the energy relative error.

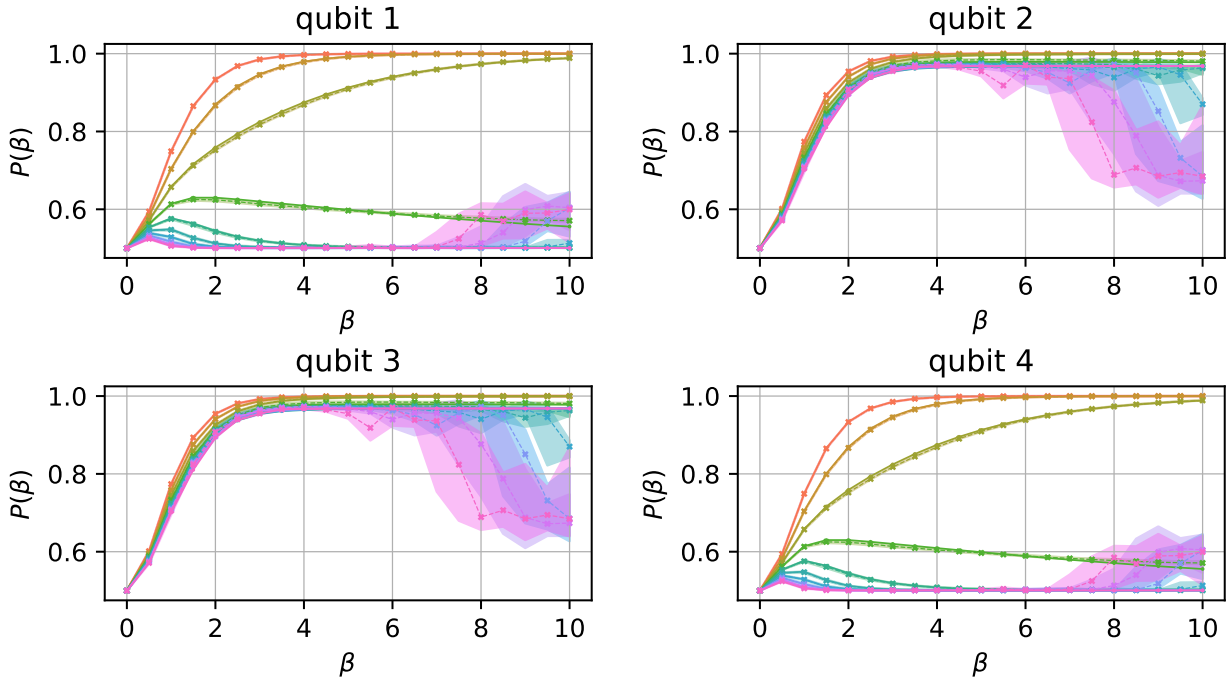


Figure A.7: Colored plot showing the purities expectation values of each qubit in the system, as a function of the inverse temperature β, qubits 1 and 4 being the external ones while qubits 2 and 3 being the internal ones. Different colors correspond to different $\gamma \in [1, 10] \cap \mathbb{Z}$ values. Solid lines with dot markers are the values computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the QMETTS algorithm, including the step merge approximation. Dashed lines with 'X' markers are the 50th percentile values (see Section 3.5.3) computed with Qiskit Aer simulator built from the 7 qubit *FakeJakartaV2* Qiskit fake backend, corresponding to the noiseless implementation of the QMETTS algorithm, including the step merge approximation. Shaded regions correspond to the values between the 85th and 15th percentiles. Legend is the same as in Fig. A.5, omitted here to favor readability.

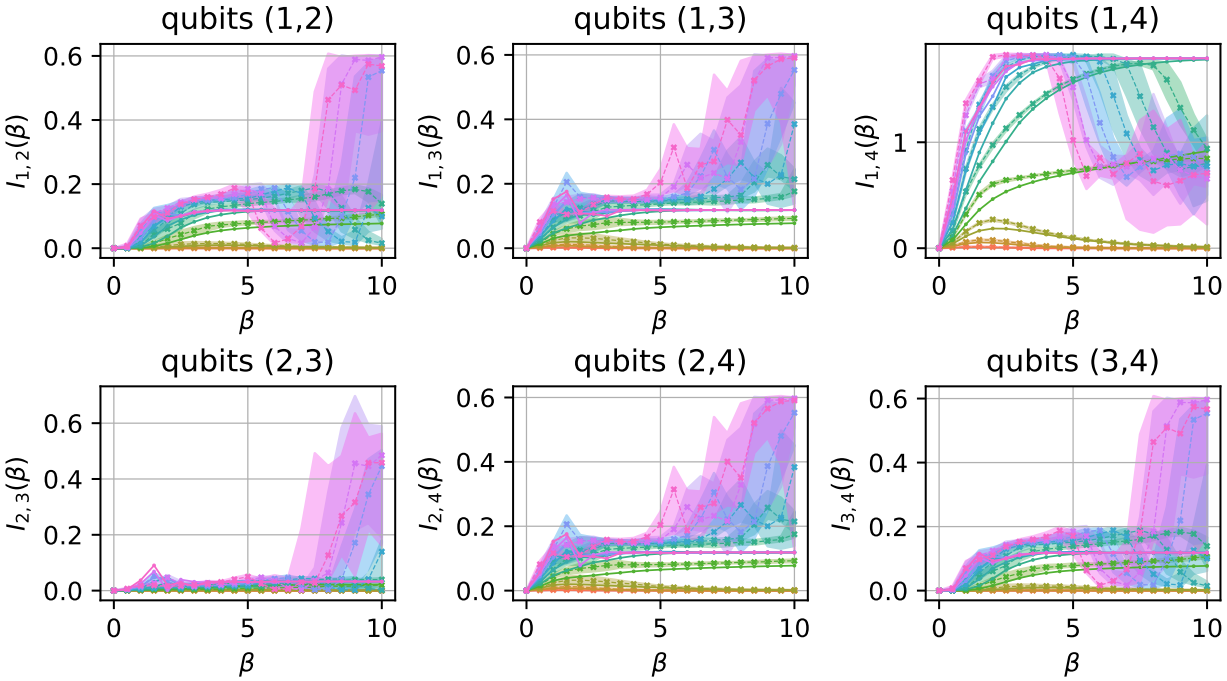


Figure A.8: Colored plot showing the mutual information expectation values of each qubit pair in the system, as a function of the inverse temperature β, qubits 1 and 4 being the external ones while qubits 2 and 3 being the internal ones. Different colors correspond to different $\gamma \in [1, 10] \cap \mathbb{Z}$ values. Solid lines with dot markers are the values computed with Qiskit Aer *StatevectorSimulator*, corresponding to the ideal implementation of the QMETTS algorithm, including the step merge approximation. Dashed lines with 'X' markers are the 50th percentile values (see Section 3.5.3) computed with the Qiskit Aer simulator built from the 7 qubit *FakeJakartaV2* Qiskit fake backend, corresponding to the noisy implementation of the QMETTS algorithm, including the step merge approximation. Shaded regions correspond to the values between the 85th and 15th percentiles. Legend is the same as in Fig. A.5, omitted here to favor readability.

Bibliography

[1] P.W. Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

[2] Richard P Feynman. "Simulating physics with computers". In: *International journal of theoretical physics* 21.6/7 (1982), pp. 467–488.

[3] D. Deutsch. "Quantum theory, the Church-Turing principle and the universal quantum computer". In: *Proceedings of the Royal Society of London Series A* 400.1818 (July 1985), pp. 97–117. DOI: 10.1098/rspa.1985.0070.

[4] Benjamin Schumacher. "Quantum coding". In: *Phys. Rev. A* 51 (4 Apr. 1995), pp. 2738–2747. DOI: 10.1103/PhysRevA.51.2738. URL: https://link.aps.org/doi/10.1103/PhysRevA.51.2738.

[5] Lov K. Grover. "A fast quantum mechanical algorithm for database search". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 1996, pp. 212–219. ISBN: 0897917855. DOI: 10.1145/237814.237866. URL: https://doi.org/10.1145/237814.237866.

[6] Emanuel Knill, Raymond Laflamme, and Wojciech H. Zurek. "Resilient quantum computation: error models and thresholds". In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 454.1969 (Jan. 1998), pp. 365–384. ISSN: 1471-2946. DOI: 10.1098/rspa.1998.0166. URL: http://dx.doi.org/10.1098/rspa.1998.0166.

[7] A.Yu. Kitaev. "Fault-tolerant quantum computation by anyons". In: *Annals of Physics* 303.1 (Jan. 2003), pp. 2–30. ISSN: 0003-4916. DOI: 10.1016/s0003-4916(02)00018-0. URL: http://dx.doi.org/10.1016/S0003-4916(02)00018-0.

[8] Dorit Aharonov and Michael Ben-Or. *Fault-Tolerant Quantum Computation With Constant Error Rate*. 1999. arXiv: quant-ph/9906129 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/9906129.

[9] He-Liang Huang et al. “Superconducting quantum computing: a review”. In: *Science China Information Sciences* 63.8 (June 2020). ISSN: 1869-1919. DOI: 10.1007/s11432-020-2881-9. URL: http://dx.doi.org/10.1007/s11432-020-2881-9.

[10] Colin D. Bruzewicz et al. “Trapped-ion quantum computing: Progress and challenges”. In: *Applied Physics Reviews* 6.2 (May 2019). ISSN: 1931-9401. DOI: 10.1063/1.5088164. URL: http://dx.doi.org/10.1063/1.5088164.

[11] M. AbuGhanem. *Photonic Quantum Computers*. 2024. arXiv: 2409.08229 [quant-ph]. URL: https://arxiv.org/abs/2409.08229.

[12] Loïc Henriet et al. “Quantum computing with neutral atoms”. In: *Quantum* 4 (Sept. 2020), p. 327. ISSN: 2521-327X. DOI: 10.22331/q-2020-09-21-327. URL: http://dx.doi.org/10.22331/q-2020-09-21-327.

[13] R. L. Stratonovich. “On a Method of Calculating Quantum Distribution Functions”. In: *Soviet Physics Doklady* 2 (July 1957), p. 416.

[14] J. Hubbard. “Calculation of Partition Functions”. In: *Phys. Rev. Lett.* 3 (2 July 1959), pp. 77–78. DOI: 10.1103/PhysRevLett.3.77. URL: https://link.aps.org/doi/10.1103/PhysRevLett.3.77.

[15] Alexander Altland and Ben D. Simons. *Condensed Matter Field Theory*. 2nd ed. Cambridge University Press, 2010.

[16] Mario Motta et al. “Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution”. In: *Nature Physics* 16.2 (Nov. 2019), pp. 205–210. ISSN: 1745-2481. DOI: 10.1038/s41567-019-0704-4. URL: http://dx.doi.org/10.1038/s41567-019-0704-4.

[17] Juan Miguel Arrazola et al. “Universal quantum circuits for quantum chemistry”. In: *Quantum* 6 (June 2022), p. 742. ISSN: 2521-327X. DOI: 10.22331/q-2022-06-20-742. URL: http://dx.doi.org/10.22331/q-2022-06-20-742.

[18] Y. Pehlivan et al. “Invariants of collective neutrino oscillations”. In: *Phys. Rev. D* 84 (6 Sept. 2011), p. 065008. DOI: 10.1103/PhysRevD.84.065008. URL: https://link.aps.org/doi/10.1103/PhysRevD.84.065008.

[19] John Preskill. *Quantum computing and the entanglement frontier*. 2012. arXiv: 1203.5813 [quant-ph]. URL: https://arxiv.org/abs/1203.5813.

[20] Seth Lloyd. “Universal Quantum Simulators”. In: *Science* 273.5278 (1996), pp. 1073–1078. DOI: 10.1126/science.273.5278.1073. eprint: https://www.science.org/doi/pdf/10.1126/science.273.5278.1073. URL: https://www.science.org/doi/abs/10.1126/science.273.5278.1073.

[21] E. Joos and H. D. Zeh. "The Emergence of classical properties through interaction with the environment". In: *Z. Phys. B* 59 (1985), pp. 223–243. DOI: 10.1007/BF01725541.

[22] Bryan O'Gorman et al. *Generalized swap networks for near-term quantum computing*. 2019. arXiv: 1905.05118 [quant-ph]. URL: https://arxiv.org/abs/1905.05118.

[23] Charles H. Bennett et al. "Mixed-state entanglement and quantum error correction". In: *Physical Review A* 54.5 (Nov. 1996), pp. 3824–3851. ISSN: 1094-1622. DOI: 10.1103/physreva.54.3824. URL: http://dx.doi.org/10.1103/PhysRevA.54.3824.

[24] Joschka Roffe. "Quantum error correction: an introductory guide". In: *Contemporary Physics* 60.3 (July 2019), pp. 226–245. ISSN: 1366-5812. DOI: 10.1080/00107514.2019.1667078. URL: http://dx.doi.org/10.1080/00107514.2019.1667078.

[25] L. Spagnoli, A. Roggero, and N. Wiebe. *Fault-tolerant simulation of Lattice Gauge Theories with gauge covariant codes*. 2024. arXiv: 2405.19293 [quant-ph]. URL: https://arxiv.org/abs/2405.19293.

[26] Lorenza Viola and Seth Lloyd. "Dynamical suppression of decoherence in two-state quantum systems". In: *Physical Review A* 58.4 (Oct. 1998), pp. 2733–2744. ISSN: 1094-1622. DOI: 10.1103/physreva.58.2733. URL: http://dx.doi.org/10.1103/PhysRevA.58.2733.

[27] Arefur Rahman, Daniel J. Egger, and Christian Arenz. *Learning How to Dynamically Decouple*. 2024. DOI: https://doi.org/10.1103/PhysRevApplied.22.054074. arXiv: 2405.08689 [quant-ph]. URL: https://arxiv.org/abs/2405.08689.

[28] Tomi Johnson, Stephen Clark, and Dieter Jaksch. "What is a quantum simulator?" In: *EPJ Quantum Technology* 1 (Apr. 2014). DOI: 10.1140/epjqt10.

[29] David Elieser Deutsch. "Quantum computational networks — Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences". In: *Proc. Roy. Soc. Lond. A* Volume 425.Issue 1868 (1989), pp. 73–90. DOI: 10.1098/rspa.1989.0099.

[30] Tzu-Chieh Wei. *Measurement-Based Quantum Computation*. Mar. 2021. DOI: 10.1093/acrefore/9780190871994.013.31. URL: http://dx.doi.org/10.1093/acrefore/9780190871994.013.31.

[31] Tameem Albash and Daniel A. Lidar. "Adiabatic quantum computation". In: *Reviews of Modern Physics* 90.1 (Jan. 2018). ISSN: 1539-0756. DOI: 10.1103/revmodphys.90.015002. URL: http://dx.doi.org/10.1103/RevModPhys.90.015002.

[32] Howard M. Wiseman and Gerard J. Milburn. *Quantum Measurement and Control*. Cambridge University Press, 2009.

[33] Daniel Gottesman. *The Heisenberg Representation of Quantum Computers*. 1998. arXiv: quant-ph/9807006 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/9807006.

[34] M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information*. Cambridge Series on Information and the Natural Sciences. Cambridge University Press, 2000. ISBN: 9780521635035. URL: https://books.google.it/books?id=65FqEKQOfP8C.

[35] G. C. Wick. "Properties of Bethe-Salpeter Wave Functions". In: *Phys. Rev.* 96 (4 Nov. 1954), pp. 1124–1134. DOI: 10.1103/PhysRev.96.1124. URL: https://link.aps.org/doi/10.1103/PhysRev.96.1124.

[36] HIROAKI TERASHIMA and MASAHITO UEDA. "NONUNITARY QUANTUM CIRCUIT". In: *International Journal of Quantum Information* 03.04 (Dec. 2005), pp. 633–647. ISSN: 1793-6918. DOI: 10.1142/s0219749905001456. URL: http://dx.doi.org/10.1142/S0219749905001456.

[37] F. Turro et al. "Imaginary-time propagation on a quantum chip". In: *Physical Review A* 105.2 (Feb. 2022). ISSN: 2469-9934. DOI: 10.1103/physreva.105.022440. URL: http://dx.doi.org/10.1103/PhysRevA.105.022440.

[38] Nguyen Van Hoa and Nguyen Tri Lan. "The Hubbard-Stratonovich Transformation and Exchange Boson Contributions". In: *Journal of Physics: Conference Series* 537.1 (Sept. 2014), p. 012004. DOI: 10.1088/1742-6596/537/1/012004. URL: https://dx.doi.org/10.1088/1742-6596/537/1/012004.

[39] Hagen Kleinert. *Path Integrals in Quantum Mechanics, Statistics and Polymer Physics*. Jan. 1995. ISBN: 978-981-02-0196-8. DOI: 10.1142/1081.

[40] John W. Negele and Henri Orland. *Quantum Many-particle Systems*. Westview Press, Nov. 1998. ISBN: 0738200522. URL: http://www.worldcat.org/isbn/0738200522.

[41] In: *Quantum Information and Computation* 12.11 & 12 (Nov. 2012). ISSN: 1533-7146. DOI: 10.26421/qic12.11-12. URL: http://dx.doi.org/10.26421/QIC12.11-12.

[42] M. Sajjad Athar et al. "Status and perspectives of neutrino physics". In: *Progress in Particle and Nuclear Physics* 124 (May 2022), p. 103947. ISSN: 0146-6410. DOI: 10.1016/j.ppnp.2022.103947. URL: http://dx.doi.org/10.1016/j.ppnp.2022.103947.

[43] R. D. Hoffman, S. E. Woosley, and Y.-Z. Qian. "Nucleosynthesis in Neutrino-driven Winds. II. Implications for Heavy Element Synthesis". In: *The Astrophysical Journal* 482.2 (June 1997), pp. 951–962. ISSN: 1538-4357. DOI: 10.1086/304181. URL: http://dx.doi.org/10.1086/304181.

[44] Hans-Thomas Janka. "Explosion Mechanisms of Core-Collapse Supernovae". In: *Annual Review of Nuclear and Particle Science* 62.1 (Nov. 2012), pp. 407–451. ISSN: 1545-4134. DOI: 10.1146/annurev-nucl-102711-094901. URL: http://dx.doi.org/10.1146/annurev-nucl-102711-094901.

[45] Shinya Wanajo et al. "PRODUCTION OF ALL THE r -PROCESS NUCLIDES IN THE DYNAMICAL EJECTA OF NEUTRON STAR MERGERS". In: *The Astrophysical Journal* 789.2 (June 2014), p. L39. ISSN: 2041-8213. DOI: 10.1088/2041-8205/789/2/l39. URL: http://dx.doi.org/10.1088/2041-8205/789/2/L39.

[46] Xinyu Li and Daniel M. Siegel. "Neutrino Fast Flavor Conversions in Neutron-Star Postmerger Accretion Disks". In: *Physical Review Letters* 126.25 (June 2021). ISSN: 1079-7114. DOI: 10.1103/physrevlett.126.251101. URL: http://dx.doi.org/10.1103/PhysRevLett.126.251101.

[47] Stan Woosley and Thomas Janka. "The physics of core-collapse supernovae". In: *Nature Physics* 1.3 (Dec. 2005), pp. 147–154. ISSN: 1745-2481. DOI: 10.1038/nphys172. URL: http://dx.doi.org/10.1038/nphys172.

[48] G. Sigl and G. Raffelt. "General kinetic description of relativistic mixed neutrinos". In: *Nuclear Physics B* 406.1 (1993), pp. 423–451. ISSN: 0550-3213. DOI: https://doi.org/10.1016/0550-3213(93)90175-O. URL: https://www.sciencedirect.com/science/article/pii/055032139390175O.

[49] Yong-Zhong Qian and George M. Fuller. "Matter-enhanced antineutrino flavor transformation and supernova nucleosynthesis". In: *Physical Review D* 52.2 (July 1995), pp. 656–660. ISSN: 0556-2821. DOI: 10.1103/physrevd.52.656. URL: http://dx.doi.org/10.1103/PhysRevD.52.656.

[50] James Pantaleone. "Neutrino oscillations at high densities". In: *Physics Letters B* 287.1 (1992), pp. 128–132. ISSN: 0370-2693. DOI: https://doi.org/10.1016/0370-2693(92)91887-F. URL: https://www.sciencedirect.com/science/article/pii/037026939291887F.

[51] Sergio Pastor and Georg Raffelt. "Flavor Oscillations in the Supernova Hot Bubble Region: Nonlinear Effects of Neutrino Background". In: *Phys. Rev. Lett.* 89 (19 Oct. 2002), p. 191101. DOI: 10.1103/PhysRevLett.89.191101. URL: https://link.aps.org/doi/10.1103/PhysRevLett.89.191101.

[52] Huaiyu Duan et al. "Coherent Development of Neutrino Flavor in the Supernova Environment". In: *Physical Review Letters* 97.24 (Dec. 2006). ISSN: 1079-7114. DOI: 10.1103/physrevlett.97.241101. URL: http://dx.doi.org/10.1103/PhysRevLett.97.241101.

[53] George M. Fuller and Yong-Zhong Qian. "Simultaneous flavor transformation of neutrinos and antineutrinos with dominant potentials from neutrino-neutrino forward scattering". In: *Physical Review D* 73.2 (Jan. 2006). ISSN: 1550-2368. DOI: 10.1103/physrevd.73.023004. URL: http://dx.doi.org/10.1103/PhysRevD.73.023004.

[54] Meng-Ru Wu and Irene Tamborra. "Fast neutrino conversions: Ubiquitous in compact binary merger remnants". English. In: *Physical Review D - Particles, Fields, Gravitation and Cosmology* 95.10 (May 2017). ISSN: 2470-0010. DOI: 10.1103/PhysRevD.95.103007.

[55] A B Balantekin and H Yüksel. "Neutrino mixing and nucleosynthesis in core-collapse supernovae". In: *New Journal of Physics* 7 (Feb. 2005), pp. 51–51. ISSN: 1367-2630. DOI: 10.1088/1367-2630/7/1/051. URL: http://dx.doi.org/10.1088/1367-2630/7/1/051.

[56] Joshua D. Martin, Changhao Yi, and Huaiyu Duan. "Dynamic fast flavor oscillation waves in dense neutrino gases". In: *Physics Letters B* 800 (Jan. 2020), p. 135088. ISSN: 0370-2693. DOI: 10.1016/j.physletb.2019.135088. URL: http://dx.doi.org/10.1016/j.physletb.2019.135088.

[57] Joshua D. Martin et al. "Equilibration of quantum many-body fast neutrino flavor oscillations". In: *Phys. Rev. D* 108 (12 Dec. 2023), p. 123010. DOI: 10.1103/PhysRevD.108.123010. URL: https://link.aps.org/doi/10.1103/PhysRevD.108.123010.

[58] Luca D'Alessio et al. "From quantum chaos and eigenstate thermalization to statistical mechanics and thermodynamics". In: *Advances in Physics* 65.3 (May 2016), pp. 239–362. ISSN: 1460-6976. DOI: 10.1080/00018732.2016.1198134. URL: http://dx.doi.org/10.1080/00018732.2016.1198134.

[59] Steven R. White. "Minimally Entangled Typical Quantum States at Finite Temperature". In: *Physical Review Letters* 102.19 (May 2009). ISSN: 1079-7114. DOI: 10.1103/physrevlett.102.190601. URL: http://dx.doi.org/10.1103/PhysRevLett.102.190601.

[60] E M Stoudenmire and Steven R White. "Minimally entangled typical thermal state algorithms". In: *New Journal of Physics* 12.5 (May 2010), p. 055026. ISSN: 1367-2630. DOI: 10.1088/1367-2630/12/5/055026. URL: http://dx.doi.org/10.1088/1367-2630/12/5/055026.

[61] Benjamin Hall et al. "Simulation of collective neutrino oscillations on a quantum computer". In: *Phys. Rev. D* 104 (6 Sept. 2021), p. 063009. DOI: 10.1103/PhysRevD.104.063009. URL: https://link.aps.org/doi/10.1103/PhysRevD.104.063009.

[62] Valentina Amitrano et al. "Trapped-ion quantum simulation of collective neutrino oscillations". In: *Phys. Rev. D* 107 (2 Jan. 2023), p. 023007. DOI: 10.1103/PhysRevD.107.023007. URL: https://link.aps.org/doi/10.1103/PhysRevD.107.023007.

[63] Valentina Amitrano. "Digital Quantum Computing for Many-Body Simulations". PhD thesis. Università degli studi di Trento, 2023. DOI: https://dx.doi.org/10.15168/11572_398733.

[64] A.B. Balantekin and G.M. Fuller. "Constraints on neutrino mixing". In: *Physics Letters B* 471.2 (1999), pp. 195–201. ISSN: 0370-2693. DOI: https://doi.org/10.1016/S0370-2693(99)01371-4. URL: https://www.sciencedirect.com/science/article/pii/S0370269399013714.

[65] David O. Caldwell, George M. Fuller, and Yong-Zhong Qian. "Sterile neutrinos and supernova nucleosynthesis". In: *Phys. Rev. D* 61 (12 2000), p. 123005. DOI: 10.1103/PhysRevD.61.123005. URL: https://link.aps.org/doi/10.1103/PhysRevD.61.123005.

[66] Luca Spagnoli et al. "Collective neutrino oscillations in three flavors on qubit and qutrit processors". In: *Physical Review D* 111.10 (May 2025). ISSN: 2470-0029. DOI: 10.1103/gjr1-lf8s. URL: http://dx.doi.org/10.1103/gjr1-lf8s.

[67] C. W. Kim and A. Pevsner. *Neutrinos in physics and astrophysics*. Vol. 8. 1993.

[68] Carlo Giunti. "Quantum mechanics of neutrino oscillations". In: *11th International School on Particles and Cosmology*. May 2001. arXiv: hep-ph/0105319.

[69] Marek Zralek. "From kaons to neutrinos: Quantum mechanics of particle oscillations". In: *Acta Phys. Polon. B* 29 (1998). Ed. by J. Spalek, pp. 3925–3956. arXiv: hep-ph/9810543.

[70] L. Wolfenstein. "Neutrino oscillations in matter". In: *Phys. Rev. D* 17 (9 May 1978), pp. 2369–2374. DOI: 10.1103/PhysRevD.17.2369. URL: https://link.aps.org/doi/10.1103/PhysRevD.17.2369.

[71] S. P. Mikheyev and A. Yu. Smirnov. "Resonance Amplification of Oscillations in Matter and Spectroscopy of Solar Neutrinos". In: *Sov. J. Nucl. Phys.* 42 (1985), pp. 913–917.

[72] O. Fackler and J. Tran Thanh Van, eds. *'86 massive neutrinos in astrophysics and in particle physics. Proceedings, 6th Moriond Workshop of the 21st Rencontres de Moriond, Tignes, France, January 25-February 1,1986.* 1986.

[73] Huaiyu Duan, George M. Fuller, and Yong-Zhong Qian. "Collective Neutrino Oscillations". In: *Ann. Rev. Nucl. Part. Sci.* 60 (2010), pp. 569–594. DOI: 10.1146/annurev.nucl.012809.104524. arXiv: 1001.2799 [hep-ph].

[74] Huaiyu Duan, George M. Fuller, and Yong-Zhong Qian. "Collective neutrino flavor transformation in supernovae". In: *Phys. Rev. D* 74 (12 Dec. 2006), p. 123004. DOI: 10.1103/PhysRevD.74.123004. URL: https://link.aps.org/doi/10.1103/PhysRevD.74.123004.

[75] R. F. Sawyer. "Speed-up of neutrino transformations in a supernova environment". In: *Phys. Rev. D* 72 (4 Aug. 2005), p. 045003. DOI: 10.1103/PhysRevD.72.045003. URL: https://link.aps.org/doi/10.1103/PhysRevD.72.045003.

[76] Jonas Larson. "Integrability versus quantum thermalization". In: *Journal of Physics B: Atomic, Molecular and Optical Physics* 46.22 (Nov. 2013), p. 224016. DOI: 10.1088/0953-4075/46/22/224016. URL: https://dx.doi.org/10.1088/0953-4075/46/22/224016.

[77] Marcos Rigol, Vanja Dunjko, and Maxim Olshanii. "Thermalization and its mechanism for generic isolated quantum systems". In: *Nature* 452.7189 (Apr. 2008), pp. 854–858. ISSN: 1476-4687. DOI: 10.1038/nature06838. URL: http://dx.doi.org/10.1038/nature06838.

[78] Naoto Shiraishi and Takashi Mori. "Systematic Construction of Counterexamples to the Eigenstate Thermalization Hypothesis". In: *Phys. Rev. Lett.* 119 (3 July 2017), p. 030601. DOI: 10.1103/PhysRevLett.119.030601. URL: https://link.aps.org/doi/10.1103/PhysRevLett.119.030601.

[79] Steven Diamond and Stephen Boyd. "CVXPY: A Python-embedded modeling language for convex optimization". In: *Journal of Machine Learning Research* 17.83 (2016), pp. 1–5.

[80] Akshay Agrawal et al. "A rewriting system for convex optimization problems". In: *Journal of Control and Decision* 5.1 (2018), pp. 42–60.

[81] MOSEK ApS. *The MOSEK Python Fusion API manual. Version 11.0.* 2025. URL: https://docs.mosek.com/latest/pythonfusion/index.html.

[82] Daniel Gottesman. "Theory of fault-tolerant quantum computation". In: *Physical Review A* 57.1 (Jan. 1998), pp. 127–137. ISSN: 1094-1622. DOI: 10.1103/physreva.57.127. URL: http://dx.doi.org/10.1103/PhysRevA.57.127.