Universidad de Antofagasta
Facultad de Ciencias Básicas

Doctorado en Física Mención Física – Matemática

Design of optimization tools for quantum information theory

Daniel Uzcátegui Contreras

Profesor Tutor: Dardo Miguel Goyeneche

Antofagasta, Chile, enero 2022

Universidad de Antofagasta
Facultad de Ciencias Básicas

Doctorado en Física Mención Física – Matemática

Design of optimization tools for quantum information theory

Daniel Uzcátegui Contreras

Tesis presentada ante la ilustre Universidad de Antofagasta para optar al grado académico de Doctor en Física
Mención Física – Matemática

Profesor Tutor: Dardo Miguel Goyeneche

Comité de Evaluación:
Álvaro Restuccia
Freddy Lastra
Algo Delgado
Leonardo Neves
Gustavo Cañas

Antofagasta, Chile, enero 2022

Dedicated to my parents: María and Mario.

Acknowledgements

Here, I would like to thank many people who has helped me to make this thesis possible.

First and foremost, I would like to thank my advisor, Prof. Dardo Goyeneche, for all his guidance, encouragement and continuous support on both academic and personal level. I greatly appreciate his patience to help me go through this journey.

I would also like to thank all my professors in the program, for their support and for providing me with many tools for my further development.

I would like to express my appreciation to my fellow students, for all their help and support, and for all the special moments we shared.

My gratitude to the Universidad de Antofagasta, to the Departamento de Física of the Universidad de Antofagasta and to all the people who works to make our PhD program possible.

List of publications and ongoing projects

This thesis is based on the following publications and projects:

[A] S. Gómez, D. Uzcátegui, I. Machuca, E. S. Gómez, S. P. Walborn, G. Lima, and D. Goyeneche., *Optimal strategy to certify quantum nonlocality,* Sci Rep 11, 20489 (2021).

[B] D. Uzcátegui, G. Senno and D. Goyeneche., *Fast and simple quantum state estimation*, 2021 J. Phys. A: Math. Theor. **54** 085302.

[C] D. Uzcátegui, G. Senno and D. Goyeneche., *An algorithm for the Quantum Marginal Problem.* In preparation.

Other related publications during my PhD studies:

[D] Contreras, Daniel Uzcátegui; Goyeneche, Dardo; Turek, Ondřej and Václavíková, Zuzana., *Circulant matrices with orthogonal rows and off-diagonal entries of absolute value 1,* Communications in Mathematics, vol.29, no.1, 2021, pp.15-34.

Abstract

In this thesis, we present optimization tools for different problems in quantum information theory. First, we introduce an algorithm for *quantum estate estimation*. The algorithm consists of orthogonal projections on intersecting hyperplanes, which are determined by the probability distributions and the measurement operators. We show its performance, in both runtime and fidelity, considering realistic errors. Second, we present a technique for certifying *quantum non-locality*. Given a set of bipartite measurement frequencies, this technique finds a Bell inequality that maximizes the gap between the local hidden variable and the quantum value of a Bell inequality. Lastly, to study the quantum marginal problem, we introduce an operator and develop an algorithm, which takes as inputs a set of quantum marginals and eigenvalues, and outputs a density matrix, if exists, compatible with the prescribed data.

Contents

List of Figures

Introduction

The development of Quantum Mechanics in the early decades of the 20th century brought significant progress in the understanding of the microscopic world. One of the elements of the theory is the *state*; incorporated by Erwin Schrödinger in his equation, the state is a crucial piece for describing quantum systems. Thus, the importance of knowing the quantum state became evident since the beginning of quantum mechanics. In view of this, Wolfang Pauli posed the question of whether the results of measuring momenta and coordinates of a quantum system would be enough to determine its state [1]; this is known as the *Pauli's problem*. Since then, many *quantum state estimation* techniques have been developed. Determining the state is ubiquitous for the experimental verification of fundamental aspects of nature predicted by quantum theory and, with the advent of quantum technologies and its promising applications, it is also important for characterization of quantum devices. The increasing number of quantum systems implemented in these devices, e.g. quantum computers, demands for more efficient and scalable methods to determine the quantum state.

The implications of quantum mechanics also opened a debate about fundamental issues; most famously Albert Einstein, one of its founders, in a paper co-authored with Boris Podolsky and Nathan Rosen (EPR) [2], argued that quantum mechanics was not a *complete theory*, in the sense that it does not make predictions with certainty. According to Einstein, to complete quantum mechanics it was necessary the introduction of *local deterministic hidden variables (LHV)* [3]. Nearly thirty years later, John Bell settled many of the concerns posed in the EPR paper [4]: he provided, in the form of inequalities, an operational test to determine whether a set of correlations arise within the LHV frame. The most remarkable of Bell's findings is that some predictions from quantum theory can violate these inequalities, thus giving origing to the phenomenon of *nonlocality*. Nowadays, *Bell Nonlocality*[1] is an area whose main aim is to study and characterize Bell type inequalities. Nonlocality is considered a valuable resource in quantum information theory, with applications in quantum key distribution protocols for secure communication [5], random number

1 *Nonlocality* is used to refer to both: the phenomenon and the area of study.

certification [6], among others.

Physical quantum systems formed by many bodies are challenging to understand. Particularly, the relationship between the parts and a whole, namely, to determine whether a set of subsystems, each of them described by a quantum state, is compatible with a global quantum state describing all the parts. This problem is known as the *Quantum Marginal Problem (QMP)* and has been studied since the beginning of quantum mechanics. The QMP is relevant in many fields, especially in quantum chemistry to understand the formation of molecules [7], and in condensed matter physics to study the properties of solids [8, 9]. Despite the fact that a solution to the QMP in its most general form has been elusive, important contributions has been made when considering sub-classes of the problem.

In this thesis, we present methods to tackle some of the problems mentioned above. These methods are intended to be implemented in practical applications, e.g. in labs focused on quantum information experiments, and also to study problems in quantum information theory. In chapter 1, we introduce the basic concepts required to understand subsequent ideas. In Chapter 2, we develop an algorithm for *quantum state estimation*, show its properties and performance; the main virtues of this method are simplicity and performance, when compared with state-of-the-art techniques. In Chapter 3, we present a method for *quantum non-locality certification*, based on the optimization of a function using experimental data. This optimization seeks to maximize the gap between the LHV upper bound and the quantum mechanical maximum violation of a Bell inequality using the available data and, thus, not requiring a new experiment. Finally, in Chapter 4, we present an algorithm for the *quantum marginal problem*, designed to find a multipartite quantum state describing a global system from the knowledge of some reduced parts of the system. All the codes of these tools are in the appendix and are also available in Github.

1 Preliminaries

In this chapter we introduce some notions and concepts that are necessary for the development of the ideas in the next chapters. We do not intend to discuss those notions in much depth, but to point out directly to the mathematical properties and physical meaning and illustrate with examples to make the concepts more clear.

1.1 Quantum States

In the Quantum realm, the *state* of a physical system is mathematically described by an element of finite-dimensional Hilbert space $\mathcal{H}$ [10]. In Dirac's notation, a d-dimensional quantum state can be expanded in an orthonormal basis $\{|i\rangle\}$ as

$$|\psi\rangle = \sum_{i=0}^{d-1} c_i \, |i\rangle \, . \tag{1.1}$$

Eq. (1.1) is termed a *pure state* and the symbol $|\ldots\rangle$ is called a *ket*. $\langle\psi|$ is an element of the dual space $\mathcal{H}^*$ called *bra*. For $|\psi\rangle, |\phi\rangle \in \mathcal{H}$, the operation $\langle\psi|\phi\rangle$ is called the *inner product* and it produces a complex number. Quantum states are normalized, which means that $\||\psi\rangle\|_2 = \sqrt{\langle\psi|\psi\rangle} = 1$, with $\|\ldots\|_2$ the l_2-norm.

Qubits

A *Qubit*, the fundamental unit of quantum information theory, is a two-level quantum system. Electronic or nuclear spins of an atom and the polarization of a photon are examples of qubits found in nature. For a qubit, the state (1.1) is $|\psi\rangle = c_0 \, |0\rangle + c_1 \, |1\rangle$, with $c_0, c_1 \in \mathbb{C}$. A parametrization of this state is

$$|\psi\rangle = \cos(\theta/2) \, |0\rangle + e^{i\phi} \sin(\theta/2) \, |1\rangle \, , \tag{1.2}$$

with $0 \leq \theta \leq \pi$ and $0 \leq \phi \leq 2\pi$. Geometrically, the pair (θ, ϕ) corresponds to a point on a unit sphere, as shown in Fig. 1.1. Note that any point on the sphere is a valid pure quantum state.

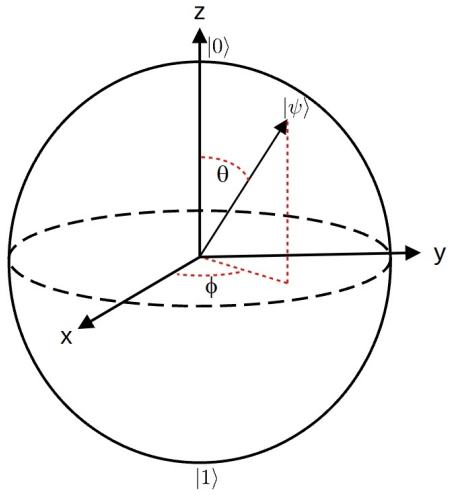


Figure 1.1: Bloch Sphere representation for Qubits. The state $|0\rangle$ is at the north Pole ($\theta = 0$) and $|1\rangle$ is at the south pole ($\theta = \pi$).

The Density Operator

A quantum state can also be expressed as a *Density Operator* or *Density Matrix*. For the state in Eq. (1.1), its density matrix representation ρ is computed as $\rho = |\psi\rangle\langle\psi|$. Density matrices can result from a statistical mixture of pure states $|\psi_i\rangle$

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|, \tag{1.3}$$

where $\sum_i p_i = 1$ and $p_i \geq 0$. There is no restriction on the number of terms in the right hand side of Eq. (1.3). States in the form of Eq. (1.3) are called *Mixed States*. A density matrix satisfying $\rho^2 = \rho$ is pure, thus pure states can be considered a special case of mixed states. The purity of a density matrix is quantified by $\mathrm{Tr}(\rho^2)$. Density matrices are *bounded operators*[2] acting on a Hilbert space and have the following properties:

i) They are linear

$$\rho\left(\sum_i c_i |\psi_i\rangle\right) = \sum_i c_i (\rho|\psi_i\rangle), \quad \text{with } |\psi_i\rangle \in \mathcal{H} \text{ and } c_i \in \mathbb{C}, \tag{1.4}$$

2 Bounded means that, for a norm $\|\ldots\|$, density operators satisfy:

$$\|\rho\| := \sup\{\|\rho|\psi\rangle\| \;\big|\; |\psi\rangle \in \mathcal{H} \text{ and } \| |\psi\rangle \| = 1\} < \infty$$

ii) hermitian $\rho = \rho^\dagger$,

iii) positive-semidefinite, denoted $\rho \geq 0$. This is,

$$\langle\varphi|\,\rho\,|\varphi\rangle \geq 0 \quad \text{for all} \quad |\varphi\rangle \in \mathcal{H}, \tag{1.5}$$

iv) and normalized $\mathrm{Tr}(\rho) = 1$.

The set of *Density Operators* acting on a Hilbert space $\mathcal{H}$ is denoted $B(\mathcal{H})$. For $\rho_\alpha \in B(\mathcal{H})$, with α the state's label, and probabilities p_α satisfying $\sum_\alpha p_\alpha = 1$, the state

$$\rho = \sum_\alpha p_\alpha \rho_\alpha, \tag{1.6}$$

is also an element of $B(\mathcal{H})$. This is, $B(\mathcal{H})$ is a *Convex set* and Eq. (1.6) is a convex combination of the states ρ_α. The border of this convex set is determined by the set of pure states.

The Bloch vector

The density matrix of a d-dimensional quantum system may be written as [11]

$$\rho = \frac{\hat{I}}{d} + \frac{1}{2}\sum_{i=1}^{d^2-1} x_i \sigma_i, \tag{1.7}$$

with $\{\sigma_i\}$ the set of $d^2 - 1$ generators of the special unitary group ($SU(d)$) and $\hat{I}$ the identity operator of dimension d. The generators σ_i satisfy the relations $\mathrm{Tr}(\sigma_i\sigma_j) = 2\delta_{ij}$, where δ_{ij} is the Kronecker delta function. Given a density matrix ρ, we can use these properties to calculate the entries of the vector $\vec{r} = (x_1, \ldots, x_{d^2-1})$ as $x_i = \mathrm{Tr}\big(\rho\sigma_i\big)$; $\vec{r}$ is known as the *Bloch vector*. For $d = 2$, Eq. (1.7) becomes

$$\rho = \frac{1}{2}\left(\hat{I} + \sum_{i=1}^{3} x_i \sigma_i\right), \tag{1.8}$$

with $\{\sigma_i\}_{i=1}^3$ the Pauli Matrices

$$\sigma_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_2 = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad \text{and} \quad \sigma_3 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{1.9}$$

and $x_i \in [-1, 1]$ for $i = 1, 2, 3$. The Bloch vector has magnitude $\|\vec{r}\| \leq 1$, where $\|\vec{r}\| = 1$ and $\|\vec{r}\| < 1$ hold for pure and mixed states, respectively. Thus, a qubit

density matrix corresponds to a point in a unit ball called the *Bloch sphere*[3] whose position in the ball is given by $\vec{r}$; pure states are located on the surface and mixed states inside the ball. For $\vec{r} = (0, 0, 0)$, Eq. (1.8) becomes $\rho = \hat{I}/2$, which is called the *maximally mixed state*.

1.2 Composite systems

The most general pure state vector of a quantum system composed of N bodies can be expressed using the *Tensor Product* [12]:

$$|\psi\rangle = \sum_{i_1=0}^{d_1-1} \dots \sum_{i_N=0}^{d_N-1} a_{i_1 \dots i_N} |i_1\rangle \otimes \dots \otimes |i_N\rangle \quad a_{i_1 \dots i_N} \in \mathbb{C}, \tag{1.10}$$

with the set $\{|i_1\rangle \otimes \dots \otimes |i_N\rangle\}$ forming an orthonormal basis in the composite space $\mathcal{H}^{d_1} \otimes \dots \otimes \mathcal{H}^{d_N}$. Conventionally, to make the notation cleaner, $|i_1\rangle \otimes \dots \otimes |i_N\rangle$ is often written just as $|i_1 \dots i_N\rangle$. It is not always possible to write the state $|\psi\rangle$ of an N-body quantum system as a *product state*

$$|\psi\rangle = |\varphi_1\rangle \otimes |\varphi_2\rangle \otimes \dots \otimes |\varphi_N\rangle , \tag{1.11}$$

with $|\varphi_i\rangle$ describing the state of the i-th subsystem. States which can be factorized as Eq. (1.11) are called *separable*. Physically, separable states are *uncorrelated*, this is, measurement outcomes on individual parts of the system do not depend on measurement outcomes obtained on other bodies. Non-separable pure states, those that cannot be expressed as Eq. (1.11), are *entangled*. In the density matrix representation, a mixed state ρ is entangled when it cannot be written as a convex combination of product states [13], this is

$$\rho \neq \sum_i p_i \rho_1^i \otimes \dots \otimes \rho_N^i, \tag{1.12}$$

with $p_i \geq 0$ and $\sum_i p_i = 1$. In general, determining whether a density matrix is separable or not is a hard task. Nonetheless, there exist some criteria to decide separability in some special cases, some of which will be discussed later in this chapter.

For a 2-body system, with local dimensions $d_1 = 2$ and $d_2 = 2$, Eq. (1.10) becomes

$$|\psi\rangle = a_{00} |00\rangle + a_{01} |01\rangle + a_{10} |10\rangle + a_{11} |11\rangle . \tag{1.13}$$

3 For historical reasons it is not called the *Bloch Ball*.

By, for example, setting $a_{00} = a_{11} = 1/\sqrt{2}$ and $a_{01} = a_{10} = 0$ in Eq. (1.13), one obtains the maximally entangled state $|\psi\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$. With coefficients $a_{00} = 1$ and $a_{11} = a_{01} = a_{10} = 0$, Eq. (1.13) becomes $|\psi\rangle = |00\rangle = |0\rangle \otimes |0\rangle$, which is a separable state .

1.2.1 Partial Trace and Marginals

Given the state of a multipartite quantum system, one might be interested in describing just a part of it, namely, to know the density matrix of a reduced part of the whole system. In general, for a quantum system formed by N bodies, with labels in the set $\mathcal{J} = \{1, \ldots, N\}$, and state described by the density matrix $\rho_{\mathcal{J}} \in B(\mathcal{H}^{d_{\mathcal{J}}})$, the state $\rho_{\mathcal{I}}$ of a reduced part of the system can be calculated as

$$\rho_{\mathcal{I}} = \mathrm{Tr}_{\mathcal{I}^c}(\rho_{\mathcal{J}}) = \sum_i (\langle i_I| \otimes \mathbb{1}_{I^c}) \rho_{\mathcal{J}} (|i_I\rangle \otimes \mathbb{1}_{I^c}), \tag{1.14}$$

where $\mathcal{I} \subset \mathcal{J}$, $\mathcal{I}^c$ is the complement of $\mathcal{I}$ with respect to $\mathcal{J}$ (see Fig. 1.2) and $< \{|i_I\rangle\}$ is an orthonormal basis for the subsystem $\mathcal{I}$. Reductions of a density matrix are called *marginals*. Similar to the trace operation, partial trace is independent of the choice of basis.

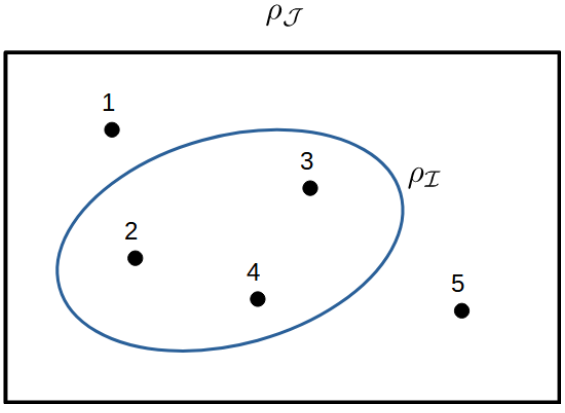


Figure 1.2: Illustration of quantum marginals. Enclosed by the rectangle we have a quantum system formed by 5 bodies and described by $\rho_{\mathcal{J}}$, with $\mathcal{J} = \{1, 2, 3, 4, 5\}$. The subsystem formed by the bodies 2, 3 and 4, is described by the marginal density matrix $\rho_{\mathcal{I}}$.

For an observable $\hat{M}$ (see section 1.4) acting on $B(\mathcal{H}^{d_{\mathcal{I}}})$, the expectation value is computed as $\langle \hat{M} \rangle = \mathrm{Tr}(\hat{M}\rho_{\mathcal{I}})$. Although we can calculate $\rho_{\mathcal{I}}$, the subsystem $\mathcal{I}$ is not separated from the whole system $\mathcal{J}$ and thus, for physical consistency,

$\langle \hat{M} \rangle$ has to satisfy

$$\langle \hat{M} \rangle = \mathrm{Tr}(\hat{M}\rho_{\mathcal{I}}) = \mathrm{Tr}\Big((\hat{M} \otimes \mathbb{1}_{\mathcal{I}^c})\rho_{\mathcal{J}}\Big). \tag{1.15}$$

It turns out that the partial trace $\mathrm{Tr}_{\mathcal{I}^c}$, defined in Eq.(1.14), is the *unique* linear operation that maps $\rho_{\mathcal{J}}$ into $\rho_{\mathcal{I}}$ such that Eq.(1.15) is satisfied for all quantum states $\rho_{\mathcal{J}}$.

1.3 Separability criteria

The Schmidt Decomposition

For a bipartite system, the state in Eq. (1.10) is given by

$$|\psi\rangle = \sum_{i_A=0}^{d_1-1} \sum_{i_B=0}^{d_2-1} a_{i_A i_B} |i_A\rangle |i_B\rangle , \tag{1.16}$$

where $\{|i_A\rangle\}$ and $\{|i_B\rangle\}$ are sets of orthonormal bases for the systems A and B, respectively. Any bipartite pure state (1.16) can also be written in the *Schmidt Decomposition*

$$|\psi\rangle = \sum_{i=0}^{r-1} \lambda_i |\alpha_i\rangle |\beta_i\rangle , \tag{1.17}$$

The number r in (1.17) is called the *Schmidt rank* or *Schmidt number*, and corresponds to the number of non-zero singular values $\{\lambda_i\}$ of the *singular value decomposition* $USV^\dagger$ of the Coefficients Matrix $[a_{i_A i_B}]$, with $a_{j_A j_B} = \langle j_A| \otimes \langle j_B| |\psi\rangle$ [4]. The sets of vectors $\{|\alpha_i\rangle\}$ and $\{|\beta_i\rangle\}$ are the columns of the unitary matrices U and V, respectively; they form two orthonormal bases. $S = [\Lambda \;\; \mathbf{0}]^T$, with Λ a square diagonal matrix with entries $(\lambda_0, \ldots, \lambda_{r-1})$. If $r = 1$, then $\lambda_0 = 1$ and Eq. (1.17) becomes $|\psi\rangle = |\alpha_0\rangle |\beta_0\rangle$. This is, vector states with Schmidt rank equal to one are separable. On the other hand, states with $r > 1$ are non-separable (or entangled).

Positive Partial Transposition

In a product basis, a bipartite density matrix ρ_{AB} can be expanded as

4 For this we use the fact that $\langle j_A| \otimes \langle j_B| |i_A\rangle \otimes |i_B\rangle = \delta_{j_A i_A}\delta_{j_B i_B}$.

$$\rho_{AB} = \sum_{m,n}^{d_A} \sum_{\mu,\nu}^{d_B} \rho_{m\mu;n\nu} |m\rangle \langle n| \otimes |\mu\rangle \langle \nu| \,. \tag{1.18}$$

The operation of transposing only one of the subsystems , for example B, is called *Partial Transposition* and is denoted $\rho_{AB}^{T_B}$, with coefficients $\rho_{m\mu;n\nu}^{T_B} = \rho_{mv;n\mu}$.

The state of a bipartite separable density matrix is $\rho_{AB} = \sum_i p_i \rho_A^i \otimes \rho_B^i$. For this case, the partial transpose $\rho_{AB}^{T_B} = \sum_i p_i \rho_A^i \otimes \rho_B^{iT}$ is always a positive matrix, since $\rho_B^{iT} \geq 0$. For entangled states, however, $\rho_{AB}^{T_B}$ might result in a matrix with negative eigenvalues. The same occurs for $\rho_{AB}^{T_A}$.

Asher Peres [14] showed that, for the bipartite case, partial transposition resulting in a positive matrix is a necessary condition for separability; this is called the *Positive Partial Transpose (PPT)* criterion. Later, Horodecki et al. [15] proved that for 2×2 and 2×3 systems, positivity of a bipartite density matrix under partial transposition is necessary and sufficient for separability. That is why the PPT criterion is also known as the *Peres–Horodecki criterion.*

Entanglement Witnesses

The Schmidt Decomposition and the PPT criterion are criteria to determine separability in the bipartite case. A more general approach is the concept of an *entanglement witness* [16]. Let S_{sep} and S_{ent} be the sets of all separable and entangled states, respectively. An observable $\hat{W}$ (see section 1.4) is an entanglement witness if and only if it satisfies the following properties:

i) $\mathrm{Tr}\left(\hat{W}\sigma\right) \geq 0$ for all $\sigma \in S_{sep}$,

ii) $\mathrm{Tr}\left(\hat{W}\rho_{ent}\right) < 0$ for at least one entangled state ρ_{ent}.

which are necessary and sufficient conditions to determine entanglement. These properties have a geometrical interpretation (see Fig. 1.3). All the states ρ satisfying $\mathrm{Tr}(\hat{W}\rho) = 0$ form a hyperplane that separates the entire set of states in two parts: the left of the hyperplane ($\mathrm{Tr}(\hat{W}\rho) \geq 0$) contains the set S_{sep}, to the right ($\mathrm{Tr}(\hat{W}\rho) < 0$) are the entangled states *detected* by W. From Fig. 1.3 is clear that some witnesses detect more entangled states than others, especially the hyperplanes tangent to S_{sep}: these witnesses are called *optimal* [17]. In Ref. [15] Horodecki et al. proved that there exists a witness for each $\rho \in S_{ent}$, but constructing witnesses is not trivial.

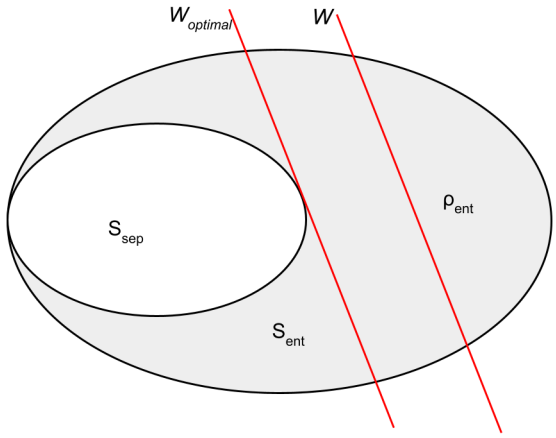


Figure 1.3: Schematic representation of the space of states. The red lines correspond to witnesses. $\hat{W}_{optimal}$ touches S_{sep} at only one state: a separable state ρ that satisfies $\mathrm{Tr}(\hat{W}\rho) = 0$.

Since witnesses are observables, they are an important tool to experimentally detect entanglement. Thus, if in some experiment we measure $\mathrm{Tr}\left(\hat{W}\rho\right) < 0$, we can assure that ρ is entangled.

1.4 Observables and Measurements

1.4.1 Observables

Measurable properties of quantum systems are called *observables* and are represented by hermitian linear operators acting on a Hilbert Space. The spectral decomposition of an observable $\hat{A}$ can be written as

$$\hat{A} = \sum_{n=0}^{d-1} \lambda_n |\lambda_n\rangle\langle\lambda_n|, \tag{1.19}$$

with $\{\lambda_i\}_{i=0}^{d-1}$ the set of eigenvalues (or *spectra*) and $\{|\lambda_i\rangle\}_{i=0}^{d-1}$ the set of eigenvectors forming an orthonormal basis. The hermitian property $\hat{A} = \hat{A}^\dagger$ guarantees that the eigenvalues $\lambda_0, \lambda_1, \ldots, \lambda_{d-1}$ are all real numbers. The eigenvalues correspond to the possible outcomes when a measurement of $\hat{A}$ is performed on a quantum system. If the system to be measured is in the state ρ, then the

probability of measuring the eigenvalue λ_i is given by the Born Rule[18]

$$p(\lambda_i) = \mathrm{Tr}\left(\rho \hat{\Pi}_i\right). \tag{1.20}$$

with $\hat{\Pi}_i = |\lambda_i\rangle\langle\lambda_i|$. The completeness property $\sum_{i=0}^{d-1} \hat{\Pi}_i = \hat{I}$, with $\hat{I}$ the identity operator, ensures that $\sum_i p(\lambda_i) = 1$ for any quantum state ρ. In practice, to estimate $p(\lambda_i)$, one would have to repeat the measurement procedure on an ensemble of identically prepared quantum systems to compute the frequency with which the eigenvalue λ_i is obtained. For a vector state $|\psi\rangle$, the expectation value $\langle\hat{A}\rangle$ of an observable $\hat{A}$ is computed as $\langle\hat{A}\rangle = \langle\psi|\,\hat{A}\,|\psi\rangle$. For a density operator ρ, $\langle\hat{A}\rangle = \mathrm{Tr}(\rho\hat{A})$.

1.4.2 General Quantum Measurements

Operators $\hat{\Pi}_i$ from Eq. (1.20), which are called *von Neumann Measurements*, are a particular case of a more general formulation called *Positive Operator-Valued Measure (POVM)*. A set $\{\hat{E}_i\}$ of operators form a POVM if they satisfy the following properties

i) They are hermitian $\hat{E}_i = \hat{E}_i^\dagger$,

ii) positive semidefinite $\hat{E}_i \geq 0$,

iii) and complete $\sum_i \hat{E}_i = \hat{I}$.

Operators $\hat{E}_i$ admit the following decomposition

$$\hat{E}_i = \hat{M}_i \hat{M}_i^\dagger, \tag{1.21}$$

with $\hat{M}_i$ a *measurement operator*. In general, the probability p_i of measuring the outcome i associated with the operator $\hat{M}_i$ is given by the Born's rule

$$p_i = \mathrm{Tr}\left(\hat{M}_i \rho \hat{M}_i^\dagger\right) = \mathrm{Tr}\left(\hat{E}_i \rho\right). \tag{1.22}$$

Von Neumann measurements $\hat{\Pi}_i$, besides the three properties above, are orthonormal; this is, $\hat{\Pi}_i\hat{\Pi}_j = \hat{\Pi}_i\delta_{ij}$. Thus, when $\hat{M}_i = \hat{\Pi}_i$, the POVM element $\hat{E}_i = \hat{\Pi}_i\hat{\Pi}_i = \hat{\Pi}_i$. Von Neumann measurements are also known as *Projective Valued Measure (PVM)* or *projectors*. Unlike PVMs, whose maximum number of possible operators $\hat{\Pi}_i$ is equal to the dimension of the Hilbert space d, there is no restriction on the number of elements in the set $\{\hat{E}_i\}$ forming a POVM.

In general, if a quantum system is initially in a state described by the density matrix ρ and the measurement of this quantum system produces the outcome

associated to the measurement operator $\hat{M}_i$, then the post-measurement state ρ' is given by

$$\rho' = \frac{\hat{M}_i \rho \hat{M}_i^\dagger}{\mathrm{Tr}(\hat{M}_i \rho \hat{M}_i^\dagger)}, \tag{1.23}$$

Eq. (1.23) is called the *projection postulate*.

POVMs are ubiquitous for many problems in quantum information theory; it has been shown that POVMs can perform much better than PVMs in many tasks, such as quantum state discrimination [19], quantum state estimation [20], quantum key-distribution [21] and randomness certification [22].

Mutually Unbiased Bases (MUB)

Let $\{|\alpha_i\rangle\}$ and $\{|\beta_j\rangle\}$ be two orthonormal bases of a d-dimensional Hilbert space. These bases are said to be *mutually unbiased* if they satisfy

$$|\langle\alpha_i|\beta_j\rangle|^2 = \frac{1}{d}, \quad \text{for every } i, j = 0, \ldots, d-1 \tag{1.24}$$

Also, a set of m bases are MUB if they are pairwise MUB. Eq. (1.24) can also be written in terms of projectors as

$$\mathrm{Tr}(P_i Q_j) = \frac{1}{d}, \tag{1.25}$$

with $P_i = |\alpha_i\rangle\langle\alpha_i|$ and $Q_j = |\beta_j\rangle\langle\beta_j|$. For example, for $d = 2$ the bases

$$\left\{\frac{|0\rangle + |1\rangle}{\sqrt{2}}, \frac{|0\rangle - |1\rangle}{\sqrt{2}}\right\}, \left\{\frac{|0\rangle + i\,|1\rangle}{\sqrt{2}}, \frac{|0\rangle - i\,|1\rangle}{\sqrt{2}}\right\}, \text{ and } \{|0\rangle, |1\rangle\},$$

corresponding to the eigenvectors of σ_1, σ_2 and σ_3, form three MUB.

MUB find applications in many problems in quantum information theory, including quantum state estimation [23, 24], entanglement detection [25] and quantum cryptography [26]. For the case of d being a prime number, Ivonovic [23] showed, by explicit construction, that there exist $d+1$ MUB. An alternative construction is given in [27], by defining the following unitary operators

$$X\,|j\rangle = |j+1\rangle, \tag{1.26}$$
$$Z\,|j\rangle = \omega^j\,|j\rangle, \tag{1.27}$$

with $\{|0\rangle, \ldots, |d-1\rangle\}$ an orthonormal basis and $\omega = \exp(2\pi i/d)$. It is clear from Eqs. (1.26) and (1.27) that $X(Z)^k\,|j\rangle = (\omega^k)^j\,|j+1\rangle$. For d a prime number,

S. Bandyopadhyay et al. showed (see theorem 2.3 in [27]) that the eigenvectors bases of the operators $Z, X, XZ, X(Z)^2, \ldots, X(Z)^{d-1}$ form a set of $d+1$ MUBs. Wooters and Fields [28] proved that $d+1$ is the maximum number of possible orthonormal bases satisfying (1.24), although it is not known whether a maximal set of $d+1$ mutually unbiased bases exists in arbitrary dimension d. In the same work the authors also proved that there exist $d+1$ MUB when $d = p^n$, with p a prime number and n a positive integer.

Symmetric Informationally Complete POVM (SIC-POVM)

Let $\{|\psi_i\rangle\}$ be a set of d^2 normalized vectors belonging to a d-dimensional Hilbert space. These vectors are *symmetric* if

$$|\langle\psi_i|\psi_j\rangle|^2 = \frac{d\delta_{ij}+1}{d+1}, \quad \text{for } i \neq j \tag{1.28}$$

for all the $|\psi_i\rangle$ in the set. Equivalently, a set of $\{\hat{\Pi}_i\}$ of d^2 subnormalized rank-1 projectors are symmetric if

$$\mathrm{Tr}\left(\hat{\Pi}_i\hat{\Pi}_j\right) = \frac{d\delta_{ij}+1}{d+1}, \quad \text{for } i \neq j \tag{1.29}$$

for all $\hat{\Pi}_i = |\psi_i\rangle\langle\psi_i|$.

A POVM is said to be *informationally complete (IC)* if a quantum state is completely determined by the probabilities obtained from the Born rule (1.22). Thus, an informationally complete POVM $\{\hat{\Pi}_i\}_{i=0}^{d^2}$ satisfying the completeness property $\sum_i d^{-1}\hat{\Pi}_i = \hat{I}$, with elements $\hat{\Pi}_i$ satisfying the symmetric property (1.29), defines a *Symmetric Informationally Complete POVM (SIC-POVM)*. Exact solutions for these measurements exist in dimensions $d = 2-53$ and 63 more dimensions in the range $57 \leq d \leq 5799$ [29]. Numerical solutions have been found for $d = 2-193$ and $d = 204, 224, 255, 288, 528, 1155$ and 2208 [29–32]. For example, the vectors $|\psi_0\rangle = |0\rangle$, $|\psi_1\rangle = (|0\rangle + \sqrt{2}|1\rangle)/\sqrt{3}$, $|\psi_2\rangle = (|0\rangle + \sqrt{2}e^{i\frac{2\pi}{3}}|1\rangle)/\sqrt{3}$ and $|\psi_3\rangle = (|0\rangle + \sqrt{2}e^{i\frac{4\pi}{3}}|1\rangle)/\sqrt{3}$ form a SIC-POVM for a qubit system.

A *complex projective t-design* [33] is formed by a finite set of m normalized vectors $\{|\psi_i\rangle\}$ satisfying

$$\frac{1}{m^2}\sum_{i,j}|\langle\psi_i|\psi_j\rangle|^{2t} = \binom{d+t-1}{t}^{-1}, \tag{1.30}$$

where $|\psi_i\rangle \in \mathcal{H}$. It turns out that maximal set of $d+1$ MUB and SIC-POVMs are complex projective 2-designs [31, 34]. For these two type of measurements, any d-dimensional density matrix ρ admits a decomposition

$$\rho = (d+1)\sum_{i=1}^{m} p_i \hat{\Pi}_i - \hat{I}, \tag{1.31}$$

with $m = d+1$ for MUB and $m = d^2$ for SIC-POVM. Thus, we can use (1.31) to reconstruct a quantum state given the experimental probabilities p_i obtained from having performed measurements $\hat{\Pi}_i$. Another advantage of SIC-POVMs is that they minimize the statistical error in the quantum state reconstruction process [35].

1.5 Bell inequalities

A *Bell scenario* is defined as the arrangement consisting of a given number of parties, the number of measurement settings and the number of possible outcomes for each measurement. In a Bell scenario involving two spatially separated parts (Alice and Bob), a physical system is prepared in a quantum state at some point midway between them, and then part of it is sent to Alice and the remaining part to Bob (see Fig. 1.4). On her part of the system, Alice can perform one out of m distinct measurements, labeled by x, which produce one out of N distinct outcomes, labeled by a (and similarly Bob, with measurements labeled by y and outcomes labeled by b). Here, Alice and Bob are free to choose their respective measurements in an arbitrary manner, but they are not allowed to communicate their outcomes during the experiment. On the other hand, they can agree on any pre-established strategy. After carrying out a finite number of identically prepared experiments, we can use the collected data to estimate the joint probabilities $p(a,b|x,y)$ satisfying $p(a,b|x,y) \geq 0$ and $\sum_{a,b} p(a,b|x,y) = 1$.

Bell inequalities are defined as

$$\sum_{xyab} s_{xy}^{ab} p(a,b|x,y) \leq C, \tag{1.32}$$

with $-1 \leq s_{xy}^{ab} \leq 1$. The set $\{p(ab|xy)\}$ of joint probabilities satisfying the *locality* condition

$$p(a,b|x,y) = \int d\lambda q(\lambda) p_\lambda(a|x) p_\lambda(b|y), \tag{1.33}$$

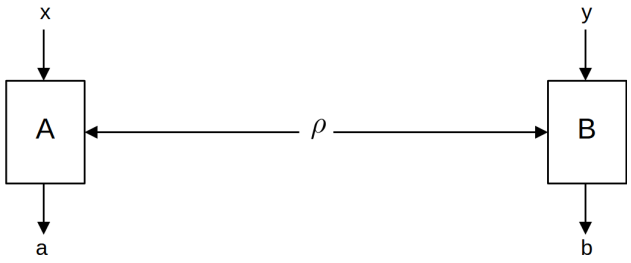


Figure 1.4: Bipartite Bell scenario. Alice (A) can choose among distinct measurements labeled by $x = 0, \ldots, m-1$, with possible outcomes labeled by $a = 0, \ldots, N-1$. Bob (B) can choose among distinct measurements labeled by $y = 0, \ldots, m-1$, with possible outcomes labeled by $b = 0, \ldots, N-1$.

is called the *local set* and is denoted $\mathcal{L}$. In a local theory, all the predictions $p(ab|xy)$ are "predetermined" and this is encoded in the *hidden variable* λ, whose values are distributed according to the probability density $q(\lambda)$. Equivalently, the condition (1.33) can be expressed as a *local deterministic model*, where the joint probabilities $p(a, b|x, y)$ are statistically independent, meaning that $p(a, b|x, y) = p(a|x)p(b|y)$, and the local probabilities $p(a|x)$ and $p(b|y)$ can only take the values 0 and 1. The upper bound C in Eq. (1.32), called the local bound, corresponds to the maximum value that the left-hand side of (1.32) can achieve considering all possible local deterministic strategies. The set $\mathcal{L}$ forms a polytope. Formally, polytopes are defined as the *convex hull* of a finite set [36]. We may also think of polytopes as a set of half-spaces intercepting each other and enclosing a region of an n-dimensional space.

In 1964, J.S Bell [4] showed that some predictions $p(a, b|x, y)$ involving quantum systems prepared in a entangled state are incompatible with a local theory[5]. This is, for an experiment involving a quantum system in the state ρ, there exist predictions given by

$$p(a, b|x, y) = \mathrm{Tr}(\rho M_a^x \otimes M_b^y), \tag{1.34}$$

that violate Bell inequalities and, therefore, do not satisfy Eq. (1.33). M_a^x and M_b^y are the measurements operators of Alice and Bob, respectively. The set of correlations given by Eq. (1.34) is denoted $\mathcal{Q}$ and it contains the local set ($\mathcal{L} \subset \mathcal{Q}$). In Fig. 1.5 we show, in addition to $\mathcal{Q}$ and $\mathcal{L}$, the *no-signaling* set $\mathcal{NS}$, which is formed by the entire set joint probabilities $\{p(a, b|x, y)\}$ satisfying the no-signaling constraints [37]

5 *Locality*, *local theory* and *local hidden-variable* are frequently used as synonymous in literature.

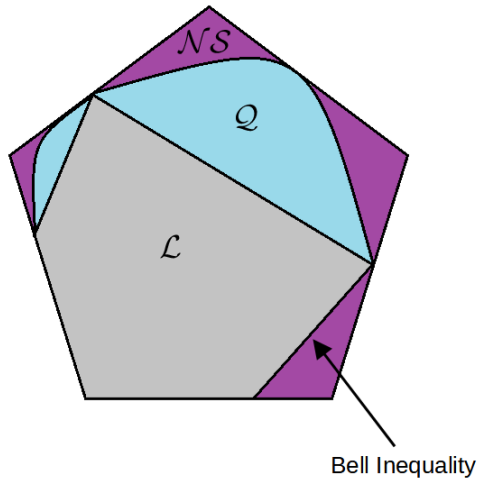


Figure 1.5: Here we depict the $\mathcal{L} \subset \mathcal{Q} \subset \mathcal{NS}$ relation between the no-signaling ($\mathcal{NS}$), quantum ($\mathcal{Q}$) and local ($\mathcal{L}$) sets. $\mathcal{L}$ is a polytope and its facets are Bell inequalities given by Eq. (1.32). Also, unlike the set $\mathcal{Q}$, $\mathcal{NS}$ is a polytope.

$$\sum_{b=0}^{N-1} p(a,b|x,y) = \sum_{b=0}^{N-1} p(a,b|x,y') =: p_A(a|x),$$
$$\sum_{a=0}^{N-1} p(a,b|x,y) = \sum_{a=0}^{N-1} p(a,b|x',y) =: p_B(b|y). \tag{1.35}$$

Although quantum mechanics agrees the no-signalling principle, there exist probabilities $p(a,b|x,y)$ satisfying (1.35) that do not belong to $\mathcal{Q}$ [37], which implies that $\mathcal{Q} \subset \mathcal{NS}$. Note that Alice's marginal probabilities $p_A(a|x)$ are independent of Bob's choice of y (analogously for Bob's marginals $p_B(b|y)$). Thus, the no-signaling constraints ensure that no instantaneous exchange of messages between Alice and Bob is possible, preventing Alice's results to be affected by Bob's choice of measurements (and vice versa). The sets $\mathcal{L}$, $\mathcal{Q}$ and $\mathcal{NS}$ are *convex*, meaning that if $p(a,b|x,y)$ and $p(a',b'|x',y')$ belong to one of these sets, then $\beta p(a,b|x,y) + (1-\beta)p(a',b'|x',y')$ also belongs to the set, for any $0 \leq \beta \leq 1$.

For a bipartite Bell scenario with $x,y \in \{0,1\}$ and $a,b \in \{0,1\}$, a very important inequality is the *CHSH* [38]:

$$p(0,0|0,0)+p(0,0|0,1)+p(0,0|1,0)-p(0,0|1,1)-p_A(0|0)-p_B(0|0) \leq 0. \tag{1.36}$$

Considering probabilities $p(a, b|x, y)$ resulting from quantum measurements, inequality (1.36) can be violated up to the maximum value of $1/\sqrt{2} - 1/2$.

Bell inequalities can also be written using *correlations*. For the bipartite scenario, the correlations are given by the expectations values of the form $\langle \hat{A}_x \otimes \hat{B}_y \rangle$, with $\hat{A}_x$ and $\hat{B}_y$ Alice's and Bob's observables, respectively. Measurements of $\hat{A}_x$ and $\hat{B}_y$ can produce one of two possible outcomes, namely their eigenvalues, labeled +1 and −1. Using correlations the CHSH inequality can be equivalently written as

$$\langle \hat{A}_0 \otimes \hat{B}_0 \rangle + \langle \hat{A}_0 \otimes \hat{B}_1 \rangle + \langle \hat{A}_1 \otimes \hat{B}_0 \rangle - \langle \hat{A}_1 \otimes \hat{B}_1 \rangle \leq 2. \tag{1.37}$$

Considering, for example, the maximally entangled state $|\psi\rangle = (|01\rangle - |10\rangle)/\sqrt{2}$ and observables $\hat{A}_x = \hat{x} \cdot \vec{\sigma}$ and $\hat{B}_y = \hat{y} \cdot \vec{\sigma}$, with $\hat{x}$ and $\hat{y}$ unitary vectors indicating the direction in which the measurements are performed, we obtain $\langle \hat{A}_x \otimes \hat{B}_y \rangle = -\hat{x} \cdot \hat{y}$. For certain directions $\hat{x}$ and $\hat{y}$, is possible to achieve $Q = 2\sqrt{2}$, see [39]. It is always possible to transform an inequality from the correlations notation into probabilities. In the case of a bipartite scenario, we can do that by applying the transformation $\langle \hat{A}_x \hat{B}_y \rangle = \sum_{a,b} (-1)^{a+b} p(a, b|x, y)$.

2 Quantum State Estimation

In this chapter, we present an iterative method to study the multipartite state estimation problem. We demonstrate convergence for any given set of informationally complete generalized quantum measurements in every finite dimension. Our method exhibits fast convergence in high dimension and strong robustness under the presence of realistic errors, both in state preparation and measurement stages. In particular, for mutually unbiased bases and tensor product of generalized Pauli observables, it converges in a single iteration.

2.1 Introduction

Quantum state estimation (QSE) is the process of estimating the density matrix from measurements performed over an ensemble of identically prepared quantum systems. In the early days of quantum theory, W. Pauli posed the question of whether position and momentum probability distributions univocally determine the state of a quantum particle [1], something that holds in classical mechanics. However, quantum states belong to an abstract Hilbert space whose dimension exponentially increases with the number of particles of the system. Thus, more information than classically expected is required to determine the state. Since then, there has been an increasing interest to estimate the state of a quantum system from a given set of measurements and several methods appeared. For instance, standard state tomography [40] reconstructs d-dimensional density matrices from $O(d^3)$ rank-one *Projective Valued Measures (PVM)*, whereas *mutually unbiased bases (MUB)* [23, 24] and *Symmetric Informationally Complete (SIC) Positive Operator Valued Measures (POVM)* [20] do the same task with $O(d^2)$ rank-one measurement projectors. In general, any tight quantum measurement [35], equivalently any complex projective 2-design, is informationally complete [33].

Quantum state estimation finds applications in communication systems [41], dissociating molecules [42] and characterization of optical devices [43]. It is a standard tool for verification of quantum devices, e.g. estimating fidelity of two photon CNOT gates [44], and has been used to characterize quantum states of trapped ions [45], cavity fields [46], atomic ensembles [47] and photons [48].

Aside from the experimental procedure of conducting a set of informationally

complete measurements on a system, QSE requires an algorithm for reconstructing the state from the measurement statistics. From a variety of techniques proposed, the approaches featuring in the majority of experiments are variants of linear inversion (LI) and maximum-likelihood quantum state estimation (MLE) [49]. As its name suggests, with LI one determines the state of the quantum system under consideration by inverting the measurement map solving a set of linear equations with the measurement data as input. For relevant families of informationally-complete set of measurements, analytical expressions for the inverse maps are known, significantly speeding up the whole reconstruction effort, see e.g. [50]. MLE aims to estimate the state that maximizes the probability of obtaining the given experimental data set, among the entire set of density matrices, as we will see in section 2.2. Within the different implementations of MLE, those currently achieving the best runtimes are variants of a projected-gradient-descent scheme, see [51, 52]. Algorithms based on variants of linear inversion [53, 54] are typically faster than those implementing MLE when the inversion process is taken from already existing inversion formulas [50]. On the other hand, when restrictions on the rank of the state being reconstructed apply, techniques based on the probabilistic method of compressed-sensing have proven to be very satisfactory [55–57]. Also, the statistics based on five rank-one projective measurements is good enough to have high fidelity reconstruct of rank-one quantum states, even under the presence of errors in both state preparation and measurement stages [58].

In Section 2.2, we give a brief introduction to some of the most efficient quantum state estimation techniques. In Section 2.3, we introduce the main ingredient of our algorithm: the *Physical Imposition Operator*, a linear operator having an intuitive geometrical interpretation. In Section 2.4, we present our iterative algorithm for quantum state estimation based on the physical imposition operator and prove its convergence. In Section 2.4.1, we show that for a wide class of quantum measurements, which include mutually unbiased bases and tensor product of generalized Pauli observables for N qudit systems, convergence is achieved in a single iteration. In Section 2.5, we numerically study the performance of our algorithm in terms of runtime and fidelity estimation.

2.2 Quantum state estimation techniques

In this section we review two very well-known techniques for QSE.

Semidefinite Programming

Let us consider the set $\mathbb{S}^n = \{X \in \mathbb{R}^{n\times n} | X = X^T\}$, this is, $\mathbb{S}^n$ is the set of $n \times n$ real symmetric matrices. A convex optimization problem with a linear objective function and unknown variable X is a *semidefinite program (SDP)* [59] if it can be stated as

$$\begin{aligned} &\text{maximize} && \mathrm{Tr}(CX), \\ &\text{subject to} && \mathrm{Tr}(A_i X) = b_i, \quad i = 1, 2, ..., m \\ & && X \geq 0, \end{aligned} \tag{2.1}$$

with $X, C, A_1, \ldots, A_m \in \mathbb{S}^n$ and $b_i \in \mathbb{R}$. An SDP written as (2.1) is in its *standard (or primal) form*. An SDP involving matrices with complex entries can be transformed to a real one using the fact that [59]

$$X \geq 0 \iff \begin{bmatrix} \Re X & -\Im X \\ \Im X & \Re X \end{bmatrix} \geq 0, \tag{2.2}$$

with $\Re X$ and $\Im X$ the real and imaginary part of X, respectively[6]. There is a problem associated to the primal (2.1), which consists in solving

$$\begin{aligned} &\text{minimize} && b^T y, \\ &\text{subject to} && \sum_{j=1}^{m} y_j A_j - C \geq 0, \\ & && y \geq 0. \end{aligned} \tag{2.3}$$

Eq. (2.3) is called the *dual* of the primal (2.1), with $y \in \mathbb{R}^m$ the unknown variable. Semidefinite programming is an extension of *linear programming* and relies on sophisticated methods, such as the *interior-point methods* [59], to solve convex optimization problems. Semidefinite programming has been applied in many fields, including, among others, control theory and combinatorial optimization, for example, in the *MAX-CUT* problem [60]. Semidefinite programming also finds applications in Quantum Information Theory, for example, in discrimination of Quantum States[61], the Quantum Marginal Problem [8, 62] and Bell Inequalities [63]. The problem of determining a density matrix ρ (see Eq. (1.22)) given the experimental probabilities p_i obtained from a set of m measurements

6 In most of convex optimization literature the symbol "$\succeq$" is used in place of "$\geq$" when referring to positive semidefinite matrices. For example, instead of $X \geq 0$, it is written $X \succeq 0$.

$\hat{M}_i$, can be formulated as an SDP:

$$\begin{aligned} &\text{maximize} && \text{Tr}(\rho) = 1, \\ &\text{subject to} && \text{Tr}(\hat{M}_i \rho) = p_i, \quad i = 1, 2, ..., m \\ & && \rho \geq 0. \end{aligned} \tag{2.4}$$

which is maximizing the constant function 1.

Maximum Likelihood Estimation (MLE)

The modern version of the MLE technique was developed and widespread by Robert Fisher between 1912 and 1922 [64]. Since then, MLE has successfully been applied for *parameter estimation* and *mathematical modeling* in many fields, ranging from psychology [65], finance [66], machine learning [67], among others. Given some observed data $y = [y_0, \ldots, y_{m-1}]^T$, with the y_is obtained from some experiment, MLE consists in finding the parameters $w = [w_0, \ldots, w_{m-1}]^T$ that maximizes the Likelihood function $L(y|w)$; the vector w is called the MLE *estimate*. In practice, for computational convenience, maximizing the log-likelihood $\log(L(y|w))$ is preferred. The Likelihood function $L(y|w)$ corresponds to *the model* and must be carefully chosen to be representative of the *phenomena or process* that generated the data y.

In Quantum Mechanics, MLE is perhaps the most known technique for QSE [68] and consists in maximizing the likelihood function

$$L(f|\rho) = \prod_k \text{Tr}\left(\rho \hat{M}_k\right)^{N f_k}, \tag{2.5}$$

this is, MLE consists in determining the density matrix ρ that maximizes the likelihood function $L(f|\rho)$ given the experimental relative frequencies $f = [f_0, \ldots, f_{m-1}]^T$. The frequency $f_i = n_i/N$ is the ratio between the number n_i of occurrences in the detector $\hat{M}_i$ and the number N of identically and independently prepared quantum systems over which the measurements are performed. Considering the log-likelihood function $\log(L(f|\rho))$, the QSE problem can be stated as

$$\begin{aligned} &\text{minimize} && -\frac{1}{N}\log(L(f|\rho)) = -\sum_{k=0}^{m-1} f_k \text{Tr}\left(\rho \hat{M}_k\right), \\ &\text{subject to} && \rho \geq 0 \quad \text{and} \quad \text{Tr}(\rho) = 1. \end{aligned} \tag{2.6}$$

Methods, such as projected-gradient descent [51], have been applied to solve the constrained optimization problem in Eq.(2.6).

2.3 Imposing physical information

Consider an experimental procedure $\mathcal{P}$ that prepares a quantum system in some *unknown* state ρ. Let us assume that, given some prior knowledge about $\mathcal{P}$, our best guess for ρ is the state ρ_0, which could be even the maximally mixed state in absence of prior information. Next, we perform a POVM measurement A composed by m_A outcomes, i.e $A = \{E_i\}_{i\leq m_A}$ on an ensemble of systems independently prepared according to $\mathcal{P}$, obtaining the outcome statistics $\vec{p} = \{p_i\}_{i\leq m_A}$. Given this newly acquired information,

how can we update ρ_0 to reflect our new state of knowledge about the system?

To tackle this question, we introduce the *physical imposition operator*, an affine map that updates ρ_0 with a matrix containing the probabilities p_i.

DEFINITION 2.3.1 (Physical imposition operator). *Let $A = \{E_i\}_{i\leq m_A}$ be a POVM acting on a d-dimensional Hilbert space $\mathcal{H}$ and let $\vec{p} \in \mathbb{R}^{m_A}$ be a probability vector. The physical imposition operator (PIO) associated to E_i and p_i is the map*

$$T_{E_i}^{p_i}(\rho) = \rho + \frac{(p_i - \mathrm{Tr}[\rho E_i])E_i}{\mathrm{Tr}(E_i^2)}, \tag{2.7}$$

for every $i \leq m_A$.

In order to clarify the meaning of the physical imposition operator (2.7) let us assume for the moment that A is a projective measurement. In such a case, operator $T_{E_i}^{p_i}(\rho)$ takes a quantum state ρ, removes the projection along the direction E_i, i.e. it removes the physical information about E_i stored in the state ρ, and imposes a new projection along this direction, weighted by the probability p_i. Here, p_i can be either taken from experimental data or simulated by Born's rule, with respect to a target state to reconstruct. Note that operator $\rho' = T_{E_i}^{p_i}(\rho)$ reflects the experimental knowledge about the quantum system. As we will show in Section 2.4, a successive iteration of PIO along an informationally complete set of quantum measurements allows us to univocally reconstruct the quantum state. After several impositions of all involved PIO, the sequence of quantum states successfully converges to a quantum state containing all the imposed physical information, as we demonstrate in Theorem 2.4.1. To simplify

notation, along the work we drop the superscript p_i in $T_{E_i}^{p_i}$ when the considered probability p_i is clear from the context.

Let us now state some important facts about PIOs that easily arise from (2.7). From now on, $\mathfrak{D}(\rho,\sigma) := \mathrm{Tr}[(\rho-\sigma)^2]$ denotes the Hilbert-Schmidt distance between states ρ and σ.

PROPOSITION 2.3.1. *The following properties hold for any POVM $\{E_i\}_{i\leq m_A}$ and any quantum states ρ,σ acting on $\mathcal{H}$:*

1. *Imposition of physical information:* $\mathrm{Tr}[T_{E_i}^{p_i}(\rho)E_i] = p_i$.
2. *Composition:*
$$T_{E_j}^{p_j}\circ T_{E_i}^{p_i}(\rho) = T_{E_i}^{p_i}(\rho) + T_{E_j}^{p_j}(\rho) - \rho - \frac{\big(p_i - \mathrm{Tr}(\rho E_i)\big)\mathrm{Tr}(E_iE_j)E_j}{\mathrm{Tr}(E_j^2)\mathrm{Tr}(E_i^2)}.$$
3. *Non-expansiveness:* $\mathfrak{D}(T_{E_j}^{p_j}(\rho), T_{E_j}^{p_j}(\sigma)) \leq \mathfrak{D}(\rho,\sigma)$.

Proof. Items *1* and *2* easily arise from (2.7). In order to show the non-expansiveness stated in Item *3*, let us apply (2.7) to two quantum states ρ and σ, acting on $\mathcal{H}$, i.e.
$$T_{E_i}^{p_i}(\rho) = \rho + \frac{(p_i - \mathrm{Tr}[\rho E_i])E_i}{\mathrm{Tr}(E_i^2)}, \tag{2.8}$$
$$T_{E_i}^{p_i}(\sigma) = \sigma + \frac{(p_i - \mathrm{Tr}[\sigma E_i])E_i}{\mathrm{Tr}(E_i^2)}. \tag{2.9}$$
Subtracting (2.8) from (2.9)
$$T_{E_i}(\rho) - T_{E_i}(\sigma) = (\rho-\sigma) - \frac{\mathrm{Tr}[(\rho-\sigma)E_i]E_i}{\mathrm{Tr}(E_i^2)}, \tag{2.10}$$
where we have dropped the upper index p_i from $T_{E_i}^{p_i}$. Now, let us compute $\mathfrak{D}(T_{E_j}(\rho), T_{E_j}(\sigma))^2 = \mathrm{Tr}\big[\big(T_{E_i}(\rho) - T_{E_i}(\sigma)\big)\big(T_{E_i}(\rho) - T_{E_i}(\sigma)\big)^\dagger\big]$. Thus,
$$\begin{aligned}\mathfrak{D}(T_{E_j}(\rho), T_{E_j}(\sigma))^2 &= \mathfrak{D}(\rho,\sigma)^2 - 2\frac{\mathrm{Tr}[(\rho-\sigma)E_i]\mathrm{Tr}[(\rho-\sigma)E_i]}{\mathrm{Tr}(E_i^2)} \\ &\quad + \frac{\big(\mathrm{Tr}[(\rho-\sigma)E_i]\big)^2\mathrm{Tr}(E_i^2)}{\big(\mathrm{Tr}(E_i^2)\big)^2} \\ &= \mathfrak{D}(\rho,\sigma)^2 - \frac{\big(\mathrm{Tr}[(\rho-\sigma)E_i]\big)^2}{\mathrm{Tr}(E_i^2)},\end{aligned} \tag{2.11}$$

where $\mathfrak{D}(\rho,\sigma)^2 = \mathrm{Tr}\left[(\rho-\sigma)(\rho-\sigma)^\dagger\right]$. Therefore, $\mathfrak{D}(T_{E_j}(\rho), T_{E_j}(\sigma)) \leq \mathfrak{D}(\rho,\sigma)$ and item *3* holds. ■

Some important observations arise from Prop. 2.3.1. First, for $j = i$ in the above item *2* we find that

$$T_{E_i}^2(\rho) = T_{E_i}(\rho). \tag{2.12}$$

Note that any quantum state $\sigma = T_{E_i}(\rho)$ is a fixed point of T_{E_i}, i.e. $T_{E_i}(\sigma) = \sigma$, which simply arises from (2.12). Roughly speaking, quantum states already having the physical information we want to impose are fixed points of the map T_{E_j}. This key property allows us to apply dynamical systems theory [69] to study the quantum state estimation problem. We consider the alternating projection method, firstly studied by Von Neumann [70] for the case of two alternating projections and generalized by Halperin to any number of projections [71], which strongly converges to a point onto the intersection of a finite number of affine subspaces [72].

In Theorem 2.4.1, we will show that composition of all physical imposition operators associated to an informationally complete POVM produces a map having a unique attractive fixed point, i.e., the solution to the quantum state tomography problem. The uniqueness of the fixed point guarantees a considerable speed up of the method in practice, as any chosen seed monotonically approaches to the solution of the problem.

To simplify notation, we consider a single physical imposition operation $\mathcal{T}_A$ for an entire POVM A, defined as follows

$$\mathcal{T}_A = T_{E_{m_A}} \circ \cdots \circ T_{E_1}. \tag{2.13}$$

For any PVM A, operator $\mathcal{T}_A$ reduces, up to an additive term proportional to the identity (omitted), to

$$\mathcal{T}_A(\rho) = \sum_{i=1}^{m_A} T_{E_i}(\rho), \tag{2.14}$$

what follows from considering (2.13) and Prop. 2.3.1. The additive property (2.14) holding for PVM measurements plays an important role, as it helps to reduce the runtime of our algorithm. This additive property holding for PVM measurements plays an important role, as it helps to reduce the runtime of our algorithm. Precisely, this fact allows us to apply *Kaczmarz method* [73]. Kaczmarz method considers projections over the subspace generated by the intersection of all associated hyperplanes, defined by the linear system of equations (Born's rule).

DEFINITION 2.3.2 (Generator state). *Given a POVM* $A = \{E_i\}_{i \leq m_A}$ *and a probability vector* $\vec{p} \in \mathbb{R}^{m_A}$, *a quantum state* ρ_{gen} *is called* generator state *for* $\vec{p}$ *if* $\mathrm{Tr}(\rho_{gen} E_i) = p_i$, *for every* $i \leq m_A$.

Note that ρ_{gen} is a fixed point of $\mathcal{T}_{E_i}$, according to (2.7) and (2.13). State ρ_{gen} plays an important role to implement numerical simulations, as it guarantees to generate sets of probability distributions compatible with the existence of a positive semidefinite solution to the quantum state tomography problem.

To conclude this section, note that map $\mathcal{T}_A$ defined in (2.14) has a simple interpretation in the Bloch sphere for a qubit system, see Fig. 2.1. The image of $\mathcal{T}_A$, i.e. $\mathcal{T}_A[\mathrm{Herm}(\mathcal{H}_2)]$, is a plane that contains the disk

$$D_A^{\vec{p}} = \{z = p_2 - p_1 \mid z = \mathrm{Tr}(\rho\sigma_z),\ p_i = \mathrm{Tr}(\rho E_i),\ \rho \geq 0, \mathrm{Tr}(\rho) = 1\},$$

i.e., the disk contains the full set of generator states ρ_{gen}. Note that $\mathcal{T}_A$ is not a completely positive trace preserving (CPTP) map, as $\mathcal{T}_A[\mathrm{Herm}(\mathcal{H}_2)]$ extends beyond the disk $D_A^{\vec{p}}$, i.e. outside the space of states. Indeed, for any state ρ that is not a convex combination of projectors E_i, there exists a probability distribution $\vec{p}$ such that $\mathcal{T}_A(\rho)$ is not positive semi-definite. Roughly speaking, any point inside the Bloch sphere from Fig. 2.1 but outside the blue vertical line is projected by $\mathcal{T}_A$ outside the sphere, for a sufficiently small disk $D_A^{\vec{p}}$.

2.4 Algorithm for quantum state estimation

In the practice of quantum state estimation, one collects a set of probability distributions $\vec{p}_1, \ldots, \vec{p}_\ell$ from a set of ℓ POVM measurements $A_1, \ldots, A_\ell$, with $A_k = \{E_i^{(k)}\}_{i \leq m_k}$, implemented over an ensemble of physical systems identically prepared in a quantum state ρ_{gen}. The statistics collected allows a unique state reconstruction when considering an *informationally-complete* (IC) sets of observables $A_1, \ldots, A_\ell$ under the absence of sources of errors. Our algorithm for quantum state estimation, Algorithm 1 below, defines a sequence of hermitian operators ρ_n, not necessarily composed by quantum states, that converges to the unique quantum state that is solution to the reconstruction problem, i.e. the quantum state ρ_{gen}. For the moment, we assume error-free state estimation in our statements. The algorithm applies to any finite dimensional Hilbert space $\mathcal{H}$, and any informationally complete set of quantum measurements.

In Algorithm 1, $\mathcal{B}(\mathcal{H})$ denotes the set of density operators over $\mathcal{H}$. The presence of errors in real probabilities may lead algorithm 1 to find a matrix ρ with negative eigenvalues. In that case, the last step finds the estimate state ρ_{est}

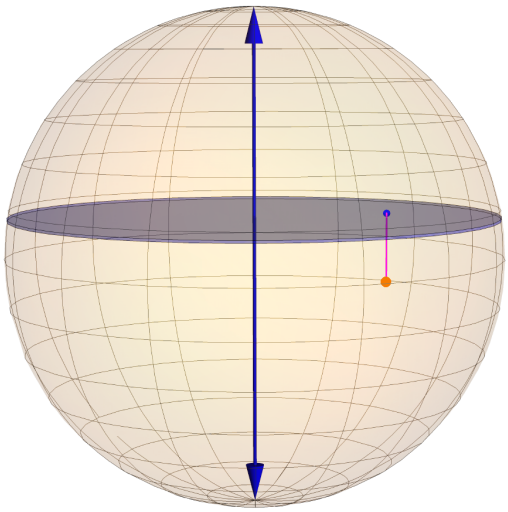


Figure 2.1: Bloch sphere representation for a single qubit system and PVM measurements. The blue arrows define the eigenvectors of σ_z. The grey disk represents the entire set of quantum states ρ_{gen} satisfying equations $p_j = \mathrm{Tr}(\rho_{gen} E_j)$, $j = 0, 1$, where E_0 and E_1 are rank-one eigenprojectors of an observable and $\{p_j\}$ the set of probabilities experimentally obtained. The action of $\mathcal{T}_A$ over the initial state ρ_0 (orange dot) is the orthogonal projection to the plane that contains the disk (blue dot) [color online].

for which $\mathfrak{D}(\rho, \rho_{est})$ is minimum; after finding the spectral decomposition $\rho = U diag(\vec{\lambda}) U^\dagger$, the best estimate is $\rho_{est} = U diag\left(\left[\vec{\lambda} - x_0 \vec{1}\right]^+\right) U^\dagger$ [50], where the function $[\cdot]^+$ has components $[\vec{y}]_i^+ = max\{y_i, 0\}$ and x_0 is such that $f(x_0) = 0$, with $f(x) = 2 + \mathrm{Tr}(\rho) - dx + \sum_{i=1}^{d} |\lambda_i - x|$.

Theorem 2.4.1 below asserts the convergence of Algorithm 1 when the input frequencies are exact, i.e. Born-rule, probabilities of an IC set of POVMs.

THEOREM 2.4.1. *Let $A_1, \ldots, A_\ell$ be a set of informationally complete POVMs acting on a Hilbert space $\mathcal{H}$, associated to a set of probability distributions $\vec{p}_1, \ldots, \vec{p}_\ell$ compatible with the existence of a quantum state. Therefore, Algorithm 1 converges to the unique solution to the quantum state tomography problem.*

Proof. First, from item *1* in Prop. 2.3.1 the generator state ρ_{gen} is a fixed point of each imposition operator $\mathcal{T}_{A_i}$, for every chosen informationally complete POVM measurement $A_1, \ldots, A_\ell$. Hence, ρ_{gen} is a fixed point of the composition of all involved operators. Moreover, this fixed point is unique, as there is no other quantum state having the same probability distributions for the considered measurements, as $A_1, \ldots, A_\ell$ are informationally complete. Here, we are assuming error-free probability distributions. Finally, convergence of

Algorithm 1: Quantum state estimation algorithm.

Input: dimension $d \in \mathbb{N}$, POVMs $A_1, \ldots, A_\ell$ acting on $\mathcal{H}$, experimental frequencies $\vec{f}_1, \ldots, \vec{f}_\ell \in \mathbb{R}^m$ and accuracy $\epsilon \in [0, 1]$.
Output: estimate $\rho_{est} \in \mathcal{B}(\mathcal{H})$.

$\rho_0 = \mathbb{I}/d$
$\rho = \mathcal{T}_{A_\ell} \circ \cdots \circ \mathcal{T}_{A_1}(\rho_0)$
repeat
  $\rho_{old} = \rho$
  $\rho = \mathcal{T}_{A_\ell} \circ \cdots \circ \mathcal{T}_{A_1}(\rho_{old})$
until $\mathfrak{D}(\rho, \rho_{old}) \leq \epsilon$
return $\arg\min_{\rho_{est} \in \mathcal{B}(\mathcal{H})} \mathfrak{D}(\rho, \rho_{est})$

our sequences is guaranteed by Theorem 1 in [74], which states that successive iterations of non-expansive projections converge to a common fixed point of the involved maps. ■

Here, compatibility refers to the existence of a quantum state associated to exact probability distributions $\vec{p}_1, \ldots, \vec{p}_\ell$ what is guaranteed when probabilities come from a generator state ρ_{gen}. Theorem 2.4.1 asserts, in other words, that the composite map $\mathcal{T}_{A_\ell} \circ \cdots \circ \mathcal{T}_{A_1}$ defines a dynamical system having a unique attractive fixed point. The successive iterations of Algorithm 1 define a *Picard sequence* [75]:

$$\begin{aligned} \rho_0 &= \mathbb{I}/d, \\ \rho_n &= \mathcal{T}_{A_\ell} \circ \cdots \circ \mathcal{T}_{A_1}(\rho_{n-1}), \ n \geq 1. \end{aligned} \tag{2.15}$$

Note that for arbitrarily chosen set of measurements, the composition of physical imposition operators depends on its ordering. According to Theorem 2.4.1, this ordering does not affect the success of the convergence in infinitely many steps. However, in practice one is restricted to a finite sequence, where different orderings produce different quantum states as an output. Nonetheless, such difference tends to zero when the state ρ_n is close to the attractive fixed point, i.e. solution to the state tomography problem. According to our experience from numerical simulations, we did not find any advantage from considering a special ordering for composition of operators.

Figure 2.2 shows the convergence of ρ_n in the Bloch sphere representation for a single qubit system and three PVMs taken at random. For certain families of measurements, e.g. mutually unbiased bases and tensor product of Pauli

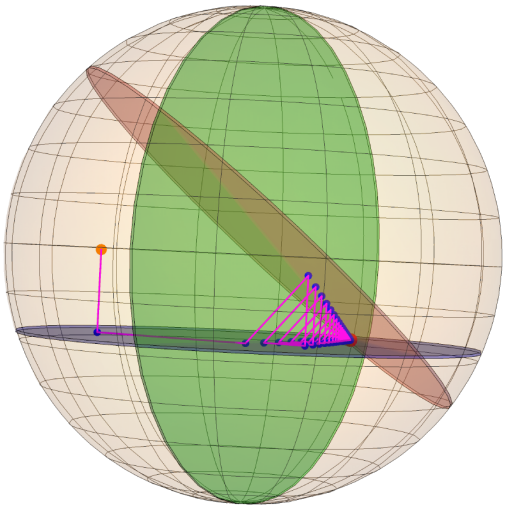


Figure 2.2: Graphical representation of the convergence of Algorithm 1 in the Bloch sphere for a single qubit system. We show convergence for three incompatible PVMs A_1, A_2 and A_3, defining disks D_1 (grey), D_2 (green) and D_3 (red) in the Bloch sphere. The initial state ρ_0 (orange dot), which we have chosen different from $\mathbb{I}/2$ only for graphical purposes, is first projected to D_1. The corresponding point in D_1 is then projected to D_2 and that projection is later projected to D_3. The iteration of this sequence of projections successfully converges to the generator state ρ_{gen} (red dot), the unique solution to the quantum state tomography problem [color online].

matrices, the resulting Picard sequences and, therefore, Algorithm 1 converges in a single iteration, see Prop. 2.4.2. That is, $\rho_n = \rho_1$ for every $n \geq 1$. We numerically observed this same behaviour for the 3^N product Pauli eigenbases in the space of N-qubits, with $1 \leq N \leq 8$, conjecturing that it holds for every $N \in \mathbb{N}$, see Section 2.5.2.

2.4.1 Single iteration convergence

When considering maximal sets of mutually unbiased bases, the Picard sequences featuring in Algorithm 1 converge in a single iteration. This is so because the associated imposition operators commute for MUB. This single-iteration convergence is easy to visualize in the Bloch sphere (see Fig. 2.3) for a qubit system, as the three disks associated to three MUB are mutually orthogonal, and orthogonal projections acting over orthogonal planes keep the impositions within the intersection of the disks. The same argument also holds in every dimension. Let us formalize this result.

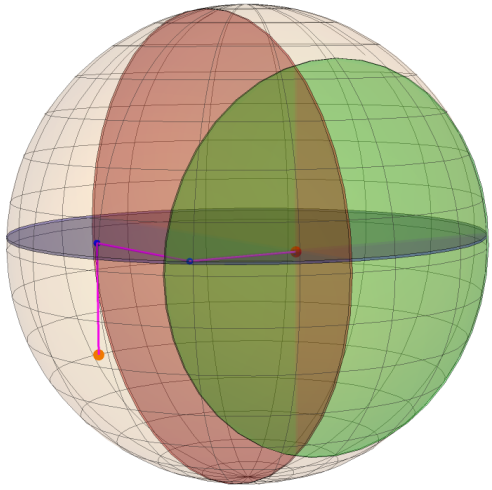


Figure 2.3: Here we show the convergence of the algorithm when MUBs are considered. We see that it only requires to project once on each plane to converge to the solution.

PROPOSITION 2.4.1. *Let T_A and T_B be two physical imposition operators associated to two mutually unbiased bases A and B. Therefore,*

$$T_B \circ T_A = T_A \circ T_B = T_A + T_B - \mathbb{I}, \tag{2.16}$$

with $\mathbb{I}$ the identity operator. In particular, note that T_A and T_B commute.

Proof. First, it is simple to show that $\mathcal{T}_A(\rho) = \rho_0 + \sum_{j=0}^{m_A-1} \Pi_j(\rho - \rho_0)\Pi_j$ for any PVM A, where $\Pi_j = E_j$ are the subnormalized rank-one PVM elements[7]. Thus, we have

$$\begin{aligned} T_B \circ T_A(\rho_0) &= \rho_0 + \sum_{j=0}^{m_A-1} \Pi_j^A(\rho - \rho_0)\Pi_j^A \\ &\quad + \sum_{k=0}^{m_B-1} \Pi_k^B \left[\rho - \left(\rho_0 + \sum_{j=0}^{m_A-1} \Pi_j^A(\rho - \rho_0)\Pi_j^A\right)\right]\Pi_k^B, \\ &= \rho_0 + \sum_{j=0}^{m_A-1} \Pi_j^A(\rho - \rho_0)\Pi_j^A + \sum_{k=0}^{m_B-1} \Pi_k^B(\rho - \rho_0)\Pi_k^B \\ &\quad + \sum_{j,k} \Pi_k^B \Pi_j^A(\rho - \rho_0)\Pi_j^A \Pi_k^B. \end{aligned}$$

7 We thank Felix Huber for highlighting this property

On the other hand, from item 2 in Prop. 2.3.1

$$\begin{aligned}\sum_{j,k} \Pi_k^B \Pi_j^A (\rho - \rho_0) \Pi_j^A \Pi_k^B &= \sum_{j,k} \mathrm{Tr}(\Pi_j^A \Pi_k^B) \mathrm{Tr}\big((\rho - \rho_0)\Pi_j^A\big)\Pi_k^B, \\ &= \gamma(A,B) \sum_{j,k} \mathrm{Tr}\big((\rho - \rho_0)\Pi_j^A\big)\Pi_k^B, \\ &= \gamma(A,B)\mathrm{Tr}(\rho - \rho_0)\mathbb{1} = 0.\end{aligned}$$

Therefore, we have

$$\begin{aligned}T_B \circ T_A(\rho_0) &= \rho_0 + \sum_{j=0}^{m_A - 1} \Pi_j^A (\rho - \rho_0)\Pi_j^A + \sum_{k=0}^{m_B - 1} \Pi_k^B (\rho - \rho_0)\Pi_k^B, \\ &= T_A(\rho_0) + T_B(\rho_0) - \rho_0, \end{aligned} \tag{2.17}$$

for any initial state ρ_0. So, we have $T_B \circ T_A = T_A \circ T_B = T_A + T_B - \mathbb{I}$. ■

Also, it is easy to see from Item *2*, Prop. 2.3.1 that operators T_{E_i} commute when considering E_i equal to the tensor product local Pauli group. In this case, operators E_i do not form a POVM but given that they define an orthogonal basis in the matrix space, they are an informationaly complete set of observables. Let us now show the main result of this section:

PROPOSITION 2.4.2. *Algorithm 1 converges, in a single iteration, to the unique solution of the quantum state tomography problem for product of generalized Pauli operators and also for $d+1$ mutually unbiased bases, in any prime power dimension d.*

Proof. For generalized Pauli operators, commutativity of imposition operators comes from orthogonality condition $\mathrm{Tr}(E_i E_j)$, see item *2* in Prop. 2.3.1. Thus, we have

$$\begin{aligned}\rho_n &= (T_{E_{d^2}} \circ \cdots \circ T_{E_1})^n(\rho_0), \\ &= T_{E_{d^2}}^n \circ \cdots \circ T_{E_1}^n(\rho_0), \\ &= T_{E_{d^2}} \circ \cdots \circ T_{E_1}(\rho_0), \end{aligned} \tag{2.18}$$

where the second step considers commutativity and the last step the fact that every T_j, $j = 1, \ldots, d+1$ is a projection. On the other hand, from Theorem 2.4.1 we know that $\rho_n \longrightarrow \rho_{gen}$ when $n \longrightarrow \infty$, for any generator state ρ_{gen}. From combining this result with (2.18) we have

$$T_{E_{d^2}} \circ \cdots \circ T_{E_1}(\rho_0) = \rho_{gen}, \tag{2.19}$$

for any seed ρ_0 and any generator state ρ_{gen}, in any prime power dimension d. For MUB the result holds in the same way, where commutativity between the associated imposition operators associated to every PVM arises from, see Prop. 2.4.1. ■

We observe from simulations that the speedup predicted by Prop. 2.4.2 has no consequences in the reconstruction fidelity of our method, which is actually higher than the one provided by MLE.

2.5 Numerical study

Theoretical developments from Sections 2.3 and 2.4 apply to the ideal case of error free probabilities coming from an exact generator state ρ_{gen}. In practice, probabilities are estimated from frequencies, carrying errors due to finite statistics. Moreover, the states being prepared in each repetition of the experiment are affected by unavoidable systematic errors. These sources of errors imply that the output of Algorithm 1 is typically outside the set of quantum states when considering experimental data. We cope with this situation by finding the closest quantum state to the output, called ρ_{est} in Hilbert-Schmidt (a.k.a. Frobenius) distance , for which there are closed-form expressions [50]. In the following, we provide numerical evidence for robustness of our method in the finite-statistics regime with white noise affecting the generator states, i.e. errors at the preparation and measurement stages. That is, we consider noisy states of the form $\tilde{\rho}(\lambda) = (1-\lambda)\rho + \lambda\mathbb{1}/d$, where λ quantifies the amount of errors. We understand there are more sophisticated techniques to consider errors, e.g. ill-conditioned measurement matrices [52]. Nonetheless, we believe the consideration of another model to simulate a small amount of errors would not substantially change the exhibited results. We reconstruct the state for N-qubit systems with $1 \leq N \leq 8$, by considering the following sets of measurements: a) Mutually unbiased bases, b) Tensor product of local Pauli bases and c) A set of $d+1$ informationally complete bases taken at random with Haar distribution. The last case does not have a physical relevance but illustrates the performance of our algorithm for a set of measurements defined in an unbiased way. As a benchmark, we compare the performance of our method with the conjugate gradient, accelerated-gradient-descent (CG-AGP) implementation of Maximum Likelihood Estimation (MLE) [51]. Computations were conducted on an Intel core i5-8265U laptop with 8gb RAM. For the CG-AGP algorithm, we used the implementation provided by authors of Ref. [51], see Ref. [76]. We provide an implementation of our Algorithm 1 in Python [77], see Appendix A, together with the code to run the simulations presented in the current section.

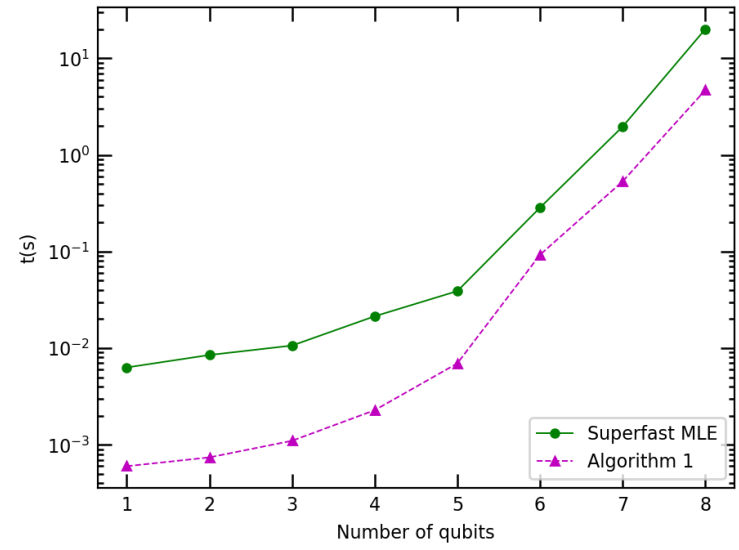
2.5.1 Mutually unbiased bases

Figure 2.4 shows performance of Algorithm 1 in the reconstruction of N-qubit density matrices from the statistics of a maximal set of $2^N + 1$ MUBs. We consider a generator state ρ_{gen} in dimension d, taken at random according to the Haar measure distribution, with the addition of a 10% level of white noise, i.e. $\tilde{\rho}(\lambda) = (1-\lambda)\rho_{gen} + \lambda\mathbb{I}/2^N$, with $\lambda = 0.1$. Here, it is important to remark that fidelities are compared with respect to the generator state ρ_{gen}, so that the additional white noise reflects the presence of systematic errors in the state preparation process. Probabilities are estimated from frequencies, i.e. $f_j = \mathcal{N}_j/\mathcal{N}$ with $\mathcal{N}_j$ the number of counts for outcome j of some POVM and $\mathcal{N} = \sum_j \mathcal{N}_j$ the total number of counts. Our simulations consider $\mathcal{N} = 100 \times 2^N$ samples per measurement basis. Our figure of merit is the fidelity $F(\rho_n, \rho_{gen}) = \mathrm{Tr}[\sqrt{\sqrt{\rho_{gen}}\rho_n\sqrt{\rho_{gen}}}]^2$ between the reconstructed state after n iterations ρ_n and the generator state ρ_{gen}. Runtime of the algorithm is averaged over 50 independent runs, each of them considering a generator state ρ_{gen} chosen at random according to the Haar measure.

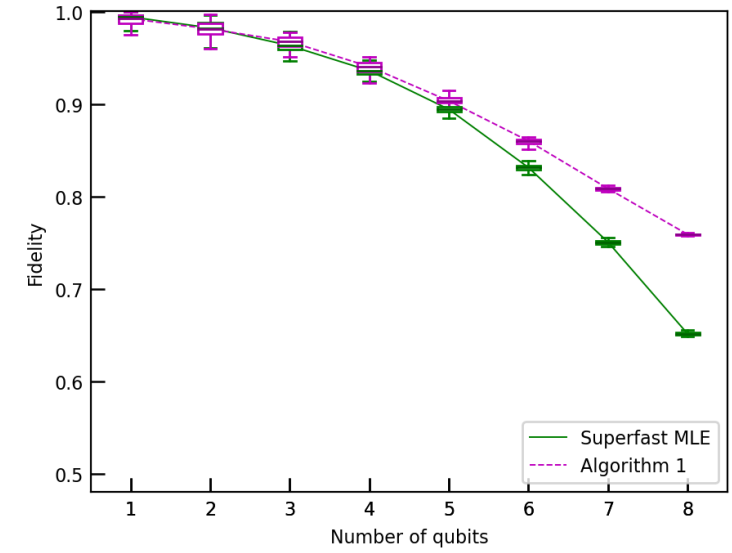
Table 2.1 shows errors in the fidelity of both algorithm 1 and the Superfast algorithm for different number of qubits. To determine the standard errors we used the *bootstrapping* technique [78]. Here, we apply 50 trials of Monte Carlo simulations using the bootstrap countings, which are distributed according to the probability distributions obtained from $\tilde{\rho}(\lambda)$. Then, the state is estimated from the simulated countings and the fidelity with respect to ρ_{gen} is calculated.

Table 2.1: Errors in the estimation of both algorithm 1 and the Superfast algorithm when considering MUB. $\bar{F}$ and δF are the mean value and the standard error of the fidelity, respectively.

Number of qubits	$\bar{F} \pm \delta F$ (Algorithm 1)	$\bar{F} \pm \delta F$ (Superfast)
1	0.9898 ± 0.0009	0.9921 ± 0.0007
2	0.9656 ± 0.0014	0.9703 ± 0.0018
3	0.9544 ± 0.0012	0.9617 ± 0.0011
4	0.9004 ± 0.0010	0.9361 ± 0.0010
5	0.8611 ± 0.0007	0.8949 ± 0.0006
6	0.8062 ± 0.0005	0.8340 ± 0.0005
7	0.7587 ± 0.0003	0.7505 ± 0.0004
8	0.7201 ± 0.0001	0.6504 ± 0.0002



(a)



(b)

Figure 2.4: Performance of Algorithm 1 and the CG-AGP Superfast MLE method from [51], for the reconstruction of N-qubit states from a maximal set of $d + 1 = 2^N + 1$ mutually unbiased basis (MUB) in dimension $d = 2^N$. Generator state ρ is chosen at random by considering the Haar measure distribution, subjected to 10% of white noise and finite statistics satisfying Poissonian distribution. For simulations, we consider 100×2^N samples. Fig. 2.4 (a) considers runtime of the algorithm in seconds, averaged over 50 trials, whereas 2.4 (b) shows bloxplots for the 50 trials datasets of the fidelity between the target and obtained state, the lines cross the medians. It is worth to mention that our simulations were performed in Python whereas Ref. [51] considers Matlab.

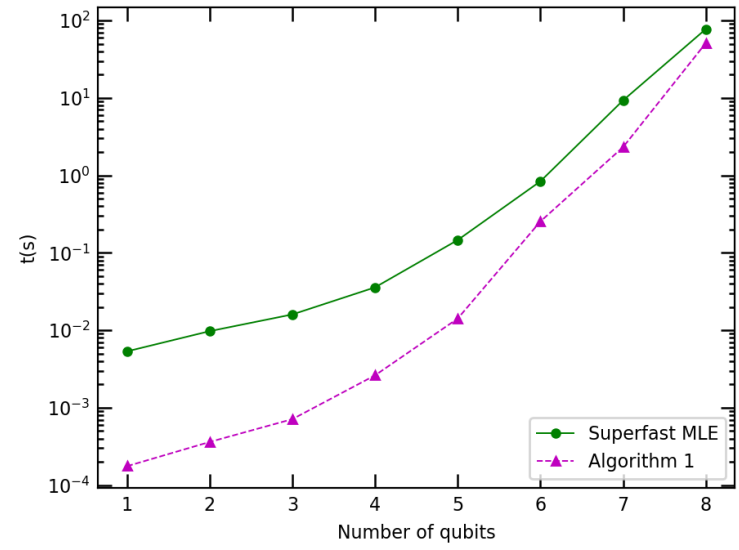
2.5.2 N-qubit Pauli bases

Here, we consider the reconstruction of N-qubit density matrices from the 3^N PVMs determined by all possible combinations of the the tensor product of single qubit Pauli eigenbases, for $N = 1, \ldots, 8$, which are informationally complete. Similarly to the case of MUBs, Picard sequences $\rho_n = T^n_{Pauli}(\rho_0)$ converge in a single iteration when product of Pauli measurements are considered, for any generator state ρ_{gen} and any initial state ρ_0. Figure 2.5 shows performance of a single iteration of these Picard sequences, where the generator state ρ_{gen} is taken at random, according to the Haar measure. Algorithm CG-AGP exploits the product structure of the N-qubit Pauli bases to speedup its most computationally expensive part: the computation of the probabilities given by the successive estimates in the MLE optimization. It does so by working with reduced density matrices which, in turn, imply an efficient use of memory. In order to have a fair comparison with our method, we decided to include the time to compute the N-qubit observables from the single Pauli observables in the total runtime of our algorithm. In practice, however, one would preload them into memory, as they are, of course, not a function of the input, i.e. of the observed probabilities.

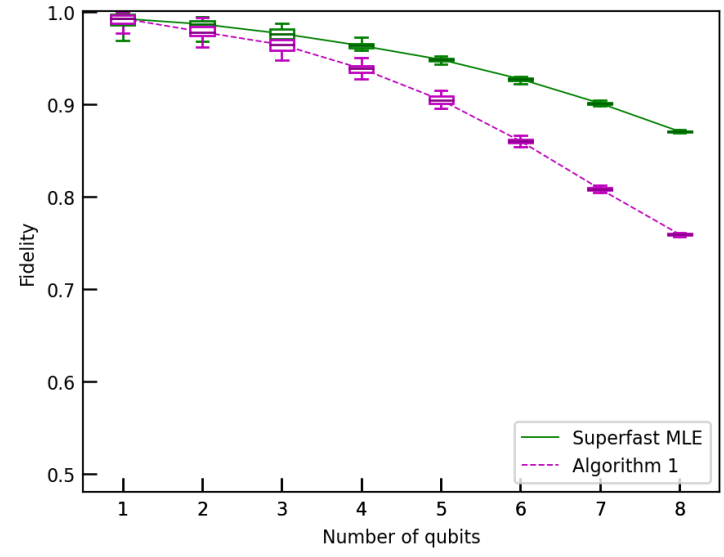
Errors in the fidelity of both algorithm 1 and the Superfast, when using Pauli bases, are shown in table 2.2. The errors were determined using the bootstrapping technique (see section 2.5.1).

Table 2.2: Errors in the estimation of both algorithm 1 and the Superfast algorithm when considering Pauli bases. $\bar{F}$ and δF are the mean value and the standard error of the fidelity, respectively

Number of qubits	$\bar{F} \pm \delta F$ (Algorithm 1)	$\bar{F} \pm \delta F$ (Superfast)
1	0.9761 ± 0.0022	0.9783 ± 0.0021
2	0.9792 ± 0.0010	0.9871 ± 0.0008
3	0.9528 ± 0.0010	0.9737 ± 0.0006
4	0.9246 ± 0.0007	0.9632 ± 0.0007
5	0.8913 ± 0.0004	0.9476 ± 0.0005
6	0.8524 ± 0.0002	0.9270 ± 0.0003
7	0.8066 ± 0.0001	0.9017 ± 0.0002
8	0.7712 ± 0.0001	0.8708 ± 0.0001



(a)



(b)

Figure 2.5: Performance of Algorithm 1 and the CG-AGP Superfast MLE [51], for the reconstruction of N-qubit states from 3^N PVM given by products of the eigenbases of local Pauli observables σ_X, σ_Y and σ_Z. Generator states ρ are chosen at random (Haar measure), subjected to 10% of white noise and finite statistics satisfying Poissonian distribution, considering 100×2^N samples per PVM. Fig. 2.5 (a) considers runtime of the algorithm in seconds, whereas 2.5 (b) shows bloxplots for the 50 trials datasets of the fidelity between the target and obtained state, the lines cross the medians. We consider simulations in Python, whereas Ref. [51] considers Matlab.

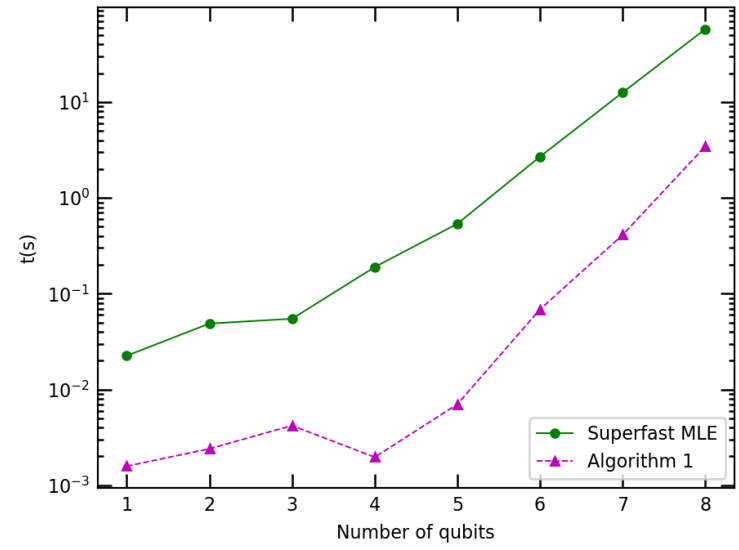
2.5.3 Random measurements for N-qubit systems

The simulations in the preceding subsections correspond to informationally complete sets of measurements for which Algorithm 1 converges in a single iteration. To test whether the advantage over [51] hinges critically on this fact, we have numerically tested our algorithm with sets of PVMs selected at random, with respect to the Haar measure. In Fig. 2.6 we show that in this case, the advantage in fidelity increases substantially, compared to Figs. 2.4 and 2.5. Table 2.2 shows the errors in the fidelity of both algorithm 1 and the Superfast for the case of random measurements.

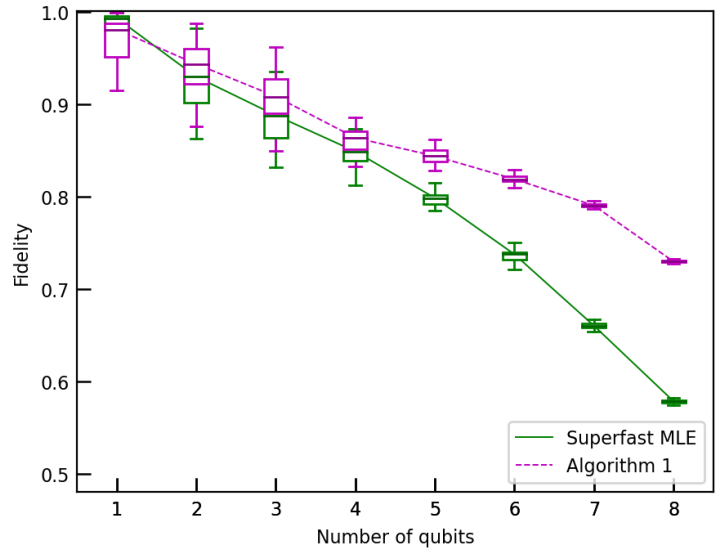
Table 2.3: Errors in the estimation of both algorithm 1 and the Superfast algorithm when considering random bases. $\bar{F}$ and δF are the mean value and the standard error of the fidelity, respectively.

Number of qubits	$\bar{F} \pm \delta F$ (Algorithm 1)	$\bar{F} \pm \delta F$ (Superfast)
1	0.9744 ± 0.0017	0.9871 ± 0.0017
2	0.9050 ± 0.0021	0.9037 ± 0.0082
3	0.8641 ± 0.0016	0.8793 ± 0.0050
4	0.8719 ± 0.0010	0.8459 ± 0.0019
5	0.8391 ± 0.0007	0.7973 ± 0.0014
6	0.7997 ± 0.0003	0.7352 ± 0.0007
7	0.7628 ± 0.0002	0.6605 ± 0.0005
8	0.7331 ± 0.0001	0.5790 ± 0.0002

Finally, we would like to mention the *Projective Least Squares* (PLS) quantum state reconstruction [50]. This method outperforms both in runtime and fidelity our Algorithm 1. This occurs when the linear inversion procedure required by the method *is not* solved but taken from analytically existing reconstruction formula. Existing inversion formulas are known for complex projective 2-designs, measurement composed by stabilizer states, Pauli observables and uniform/covariant POVM, see [50]. However, when taking into account the cost of solving the linear inversion procedure, our method has a considerable advantage over PLS. For instance, PLS does not have such efficient speed up for a number of physically relevant observables for which there is no explicit inversion known, including the following cases: a) discrete Wigner functions reconstruction for arbitrary dimensional bosons and fermions quantum systems from discrete quadratures, that can be treated as observables by considering Ramsey techniques [79], b) reconstruction of single quantized cavity mode from magnetic dipole measurements with Stern-Gerlach aparatus [80], c) minimal



(a)



(b)

Figure 2.6: Performance of Algorithm 1 and the CG-AGP Superfast MLE method from [51] for the reconstruction of N-qubit states from a a set of $d+1 = 2^N + 1$ basis chosen Haar-random in dimension $d = 2^N$. Generator state ρ is chosen under the Hilbert-Schmidt measure, subjected to 10% of white noise. Measurement statistics are estimated from $\mathcal{N} = 100 \times 2^N$ identical copies. Fig. 2.6 (a) considers the runtime of the algorithm in seconds, averaged over 50 trials, whereas 2.6 (b) shows bloxplots for the 50 trials datasets of the fidelity between the target and obtained state, the lines cross the medians.

state reconstruction of d-dimensional quantum systems from POVM consisting on d^2 elements, inequivalent to SIC-POVM [81], d) spin s density matrix state reconstruction from Stern-Gerlach measurements [82], e) Quantum state tomography for multiparticle spin 1/2 systems [83], neither reduced to mutually unbiased bases nor local Pauli measurements.

For both algorithm 1 and the CG-AGP algorithm, considering any of the measurements studied in this section, fidelity decreases when the amount of white noise is increased, as expected.

2.6 Discussion and conclusions

We introduced an iterative method for quantum state estimation of density matrices from any informationally complete set of quantum measurements in any finite dimensional Hilbert space. We demonstrated convergence to the unique solution for any informationally complete or overcomplete set of POVMs, see Theorem 2.4.1. The method, based on dynamical systems theory, exhibited a simple and intuitive geometrical interpretation in the Bloch sphere for a single qubit system, see Figs. 2.1 and 2.2. Algorithm 1 revealed a fast convergence for a wide class of measurements, including mutually unbiased bases and tensor product of generalized Pauli observables for an arbitrary large number of particles having d internal levels. Furthermore, numerical simulations revealed strong robustness under the presence of realistic errors in both state preparation and measurement stages, see Figs. 2.4 to 2.6. We provided an easy to use code developed in Python to implement our algorithm, see [76].

As interesting future lines of research, we pose the following list of open issues: (*i*) Find an upper bound for fidelity reconstruction of Algorithm 1 as a function of errors and number of iterations; (*ii*) Characterize the full set of quantum measurements for which Algorithm 1 converges in a single iteration; (*iii*) Extend our method to quantum process tomography.

3 Quantum non-locality

Certification of quantum nonlocality plays a central role in practical applications like device-independent quantum cryptography and random number generation protocols. In this chapter, we introduce a technique to find a Bell inequality with the largest possible gap between the quantum prediction and the classical local hidden variable limit for a given set of measurement frequencies. Our method represents an efficient strategy to certify quantum nonlocal correlations from experimental data without requiring extra measurements, in the sense that there is no Bell inequality with a larger gap than the one resulting from our method. Furthermore, we also reduce the photodetector efficiency required to close the detection loophole. We illustrate our technique by improving the detection of quantum nonlocality from experimental data obtained with weakly entangled photons.

3.1 Introduction

Quantum nonlocality plays a fundamental role in flourishing quantum technologies, such as device [5, 84] and semi-device [85] independent quantum cryptography, device-independent quantum secure direct communication against collective attacks [86], and genuine random number generation [6], as well as fundamental aspects of quantum physics. In these applications, certification of nonlocality is typically required. In the ideal case, for any set of nonlocal correlations, there exists a Bell inequality that is violated. However, certification of nonlocality can be hard to achieve in practice due to the presence of experimental errors. This is especially true when the optimal quantum state, i.e. the state producing the maximal violation of a given Bell inequality, is weakly entangled [87]. This problem plays a relevant role even when considering tight Bell inequalities, as also these inequalities might be maximally violated by partially entangled quantum states [88, 89]. Tight Bell inequalities are particularly useful as they are known to maximize the randomness that can be certified in a Bell scenario. For instance, a recent experiment deal with quantum nonlocality certification by using near-ideal two-qubit states for weakly entangled quantum systems [90].

Bell inequalities can be used to certify that a set of correlations cannot

be described by a local hidden variable (LHV) model. Some bipartite Bell inequalities can have a large ratio between the quantum and LHV limits, equal to $\sqrt{n}/\log n$, for n settings and n outputs in n dimensional Hilbert spaces [91]. From the experimental perspective, a larger theoretical violation increases the chance to certify quantum nonlocality in the laboratory. Nonetheless, sometimes experiments are not conclusive to certify nonlocality. Under such situation, one can simply choose another Bell inequality with a larger gap between the LHV and quantum values, thus increasing the chances for success. However, this change typically involve a new experimental implementation, as the optimal settings of the new Bell inequality most likely differ from the original one. This procedure consumes additional time and resources in the laboratory. Thus, a fundamental question arises:

> *Can we certify quantum nonlocality from experimental data that previously failed to violate a target Bell inequality?*

To start studying the problem, an asymptotically optimal analysis can be done to reject local realism of a given statistical data [92–94]. In this work, we find necessary and sufficient conditions to provide a conclusive answer to the above question, for any bipartite scenario. In addition, we present an optimization method that finds a Bell inequality that maximizes the chances to detect quantum nonlocality, among the entire set of inequalities of a given scenario. This method is particularly useful to certify nonlocality when considering weakly entangled quantum states. For instance, we successfully certify nonlocality for quantum states having smaller concurrence than those studied in a recent work [95]. Our technique finds a wide range of practical applications including communication complexity problems, where the advantage in communication is an increasing function of the quantum-LHV value gap [96, 97].

3.2 Method

In this section, we introduce a method that provides the largest possible gap between the quantum and LHV predictions for any given set of experimental data. In particular, this procedure allows us to determine whether a set of experimental data is genuinely nonlocal or not, i.e. whether there is a Bell inequality that can certify quantum nonlocality from the noisy data. Our method represents an efficient certification of nonlocal correlations, that can be applied to experimental data without requiring extra measurements. In other words, we produce a Bell inequality that maximizes the chances to detect

quantum nonlocality from a given set of statistical data among the entire set of Bell inequalities.

Let us express Eq. (1.32) in the following form

$$\sum_{x,y=0}^{m-1} \sum_{a,b=0}^{d-1} s_{xy}^{ab}\, p(a,b|x,y) \leq C, \tag{3.1}$$

where $p(a,b|x,y)$ is the probability of obtaining outcomes $a,b \in \{0,\ldots,d-1\}$ when inputs $x,y \in \{0,\ldots,m-1\}$ are chosen by two observers, Alice and Bob, respectively. Here, C denotes the maximal value of the left-hand side in (3.1) that can be achieved by considering *local hidden variable* (LHV) theories, whereas quantum mechanics might predict a violation of this inequality [4]. Without loss of generality, we can restrict our attention to parameters within the set $-1 \leq s_{xy}^{ab} \leq 1$, for every $a,b = 0,\ldots,d-1$ and $x,y = 0,\ldots,m-1$. In order to obtain the LHV value C, we have to optimize the left hand side in (3.1) over all possible local deterministic strategies. Here, locality means statistical independence of simultaneous and distant events, i.e. $p(a,b|x,y) = p(a|x)p(b|y)$, and deterministic means that all probabilities take values 0 or 1 only, restricted to the normalization conditions.

Quantum joint probability distributions satisfy the no-signaling principle, i.e. information cannot be instantaneously transmitted between distant parties. In particular, the outcome of one party cannot reveal information about the input of the other. That is expressed in the no-signaling conditions (1.35), that we rewrite in the following form

$$\sum_{b=0}^{d-1} p(a,b|x,y) = \sum_{b=0}^{d-1} p(a,b|x,y') =: p_A(a|x), \tag{3.2}$$

and

$$\sum_{a=0}^{d-1} p(a,b|x,y) = \sum_{a=0}^{d-1} p(a,b|x',y) =: p_B(b|y), \tag{3.3}$$

for every $x \neq x'$ and $y \neq y'$, where $p_A(a|x)$ and $p_B(b|y)$ are the marginal probability distributions associated to Alice and Bob, respectively.

Let us now consider a set of relative frequencies $f(a,b|x,y)$ of occurrence for outcomes a,b when x,y is measured by Alice and Bob, respectively, obtained from experimental data. The no-signaling constraints (3.2) and (3.3) are not exactly satisfied for experimental data. However, they can be recovered by

minimizing the Kullback-Leible divergence [98]:

$$D_{KL}(\vec{f}||\vec{P}) = \sum_{a,b,x,y} f(x,y) f(a,b|x,y) \log_2 \left[\frac{f(a,b|x,y)}{p(a,b|x,y)} \right], \tag{3.4}$$

where $f(x,y)$ is the relative frequency of implementing a measurement x by Alice and y by Bob, and $p(a,b|x,y)$ the optimization variables, consisting of a joint probability distribution within the framework of quantum mechanics. The minimization procedure (3.4) is equivalent to maximizing the likelihood of producing the observed frequency $p(a,b|x,y)$, see Appendix D1 in [98].

The quantum prediction of a Bell inequality (3.1), defined by coefficients s_{xy}^{ab}, is given by

$$Q = \sum_{x,y=0}^{m-1} \sum_{a,b=0}^{d-1} s_{xy}^{ab} \, p(a,b|x,y), \tag{3.5}$$

having associated an error propagation ΔQ; see section 3.3 for a detailed treatment of errors. An experimentally obtained probability distribution $p(a,b|x,y)$, associated to errors $\Delta p(a,b|x,y)$, is certainty nonlocal if $Q - C > \Delta Q$, for a given Bell inequality. However, sometimes quantum nonlocality cannot be revealed due to the relatively high amount of errors. This especially occurs when a weakly entangled quantum state produces the maximal violation of the inequality, where the gap between the lhv value and the maximal violation is very small. Under such situation, the method introduced here provides a new Bell inequality that increases the chances to prove quantum nonlocality for a given set of probability distributions $p(a,b|x,y)$, associated to experimental errors $\Delta p(a,b|x,y)$. The method consists in solving the following nonlinear problem:

$$R = \max_{s} \frac{Q - \Delta Q + dm}{C + dm}, \tag{3.6}$$

for a fixed set of statistical data, where the optimization is implemented over all coefficients s_{xy}^{ab} defining a Bell inequality (3.1). The shifting factor dm introduced in (3.6) avoids divergence of the function R; otherwise, the output inequality would be any such that the LHV value C vanishes. Optimization (3.6) is typically hard to implement, due to the presence of a large amount of local maximum values.

To solve this problem, we implement the *Sequential Least Squares Programming* (SLSQP) [99] algorithm, using routines from the scientific library (Scipy) of the Python Programming Language [100]. SLSQP is an efficient method to numerically solve constrained nonlinear optimization problems with bounds,

well suited to solve the following problem

$$\begin{array}{ll} \text{maximize} & R(\vec{s}), \qquad \text{with } \vec{s} = \{s_{xy}^{ab}\} \\ \text{subject to} & -1 \leq s_{xy}^{ab} \leq 1. \end{array}$$

Our strategy consists in running this optimization for a given number of trials. In the first trial, a random seed real vector $\vec{x}_0$, with entries taken in the range $[-1, 1]$, is given to the routine. After the first optimization procedure the algorithm outputs a vector $\vec{s}_0$. Then, the seed for the second trial is taken as $\vec{x}_1 = (\vec{s}_0 + \vec{x}_0)/2$. We update the seed as $\vec{x}_{k+1} = (\vec{s}_k + \vec{x}_k)/2$ during the trials and keep the output $\vec{s}_k = \vec{s}$ for which $R(\vec{s})$ is the greatest. The solution to the optimization problem posed above is stored in vector $\vec{s}$. This approach is highly efficient to avoid getting stuck in local maximum values, although there is no way to certify that the reached solution is the global maximum. The code for this optimization procedure was written in Python, see Appendix B, and is available at GitHub [101].

PROPOSITION 3.2.1. *A Bell inequality having LHV value C, for which a quantum value Q is achieved with errors ΔQ, is genuinely nonlocal if and only if $R > 1$ in Eq.(3.6).*

Proof. Let s_{xy}^{ab} be the parameters of the Bell inequality producing the maximum value R in (3.6). Therefore, it is simple to show that $Q - \Delta Q - C = (R-1)(C+dm)$. Therefore, the statistical data is nonlocal, i.e. $Q - \Delta Q - C > 0$, if and only if $R > 1$. Given that R takes the maximal possible value among all Bell inequalities of the scenario, if $R \leq 1$ then there is no Bell inequality that can detect quantum nonlocality for the given statistical data. ■

Here, genuinely nonlocal means that there exists a Bell inequality (3.1), associated to coefficients s_{xy}^{ab}, such that the amount of experimental errors do not overpass the gap between the quantum violation Q and the LHV prediction C. Therefore, $R > 1$ is equivalent to saying that there is a way to experimentally certify quantum nonlocality for a given set of experimental data. On the other hand, if $R \leq 1$ then there is no way to tell whether there exists a linear Bell inequality, as defined in Eq. (1.32), that detects nonlocality for the given probabilities.

Note that the gap of a Bell inequality $Q - \Delta Q - C$ can be artificially enlarged by a multiplicative factor $\kappa > 1$ by considering a Bell inequality having a rescaled set of coefficients $\tilde{s}_{xy}^{ab} = \kappa s_{xy}^{ab}$. In order to avoid this scaling problem, it is convenient to refer to the *relative gap*, given by the gap of a Bell inequality

for which $C = 1$, which can be assumed without loss of generality. Thus, the global maximum value of R in (3.6) implies the largest possible relative gap, as we show below.

PROPOSITION 3.2.2. *The Bell inequality associated to the global maximum of the function R, introduced in* (3.6)*, produces the largest possible relative gap between the LHV and quantum values, among all linear Bell inequalities of a given scenario.*

Proof. Without loss of generality, we can restrict our attention to Bell inequalities for which $C = 1$. Indeed, this can always be done by considering the following rescaling of a given Bell inequality: $Q' = Q/C$, $\Delta Q' = \Delta Q/C$ and $C' = 1$. Here, we used the fact that error propagation is linear as a function of the coefficients s_{xy}^{ab} defining the Bell inequality. After the rescaling, optimization of R is equivalent to maximize $Q' - \Delta Q'$ along all Bell inequalities satisfying $C' = 1$, according to (3.6). This is equivalent to maximizing the gap $Q' - \Delta Q' - 1 = Q' - \Delta Q' - C'$. ∎

In section 3.4, we apply our optimization method to the so-called *tilted Bell inequality* [87]. This family of inequalities was used to demonstrate that almost two bits of randomness can be extracted from a quantum system prepared in a weakly entangled state. This property makes the tilted Bell inequality an ideal candidate to test our method, due to the hardness to certify nonlocality in such case.

3.3 Error propagation

This section provides a general expression for the experimental error obtained by measuring the Bell inequality value. In particular, we show how errors in the photon counting number due to finite statistics propagates to ΔQ. Consider the following general expression for a Bell inequality:

$$Q = \sum_{x,y=0}^{m-1} \sum_{a,b=0}^{d-1} s_{xy}^{ab} p(ab|xy) + \sum_{x=0}^{m-1} \sum_{a=0}^{d-1} s_x^a p_A(a|x) + \sum_{y=0}^{m-1} \sum_{b=0}^{d-1} s_y^b p_B(b|y). \tag{3.7}$$

Here we include both joint and marginal probability distributions. As typical, the marginal probabilities are calculated from the joint probabilities, and we average over all possible x (or y), i.e. $p(a|x) = m^{-1} \sum_{y=0}^{m-1} \sum_{b=0}^{d-1} p(ab|xy)$ and $p(b|y) = m^{-1} \sum_{x=0}^{m-1} \sum_{a=0}^{d-1} p(ab|xy)$. Replacing these quantities in Eq.(3.7) and

rewriting Q in term of the coincidence count $c(ab|xy)$ we get

$$Q = \sum_{x,y=0}^{m-1} \sum_{a,b=0}^{d-1} \frac{c(ab|xy)}{\sum_{\alpha\beta} c(\alpha\beta|xy)} \left[s_{xy}^{ab} + \frac{s_x^a}{m} + \frac{s_y^b}{m} \right], \tag{3.8}$$

with $p(ab|xy) = c(ab|xy)\left(\sum_{\alpha\beta} c(\alpha\beta|xy)\right)^{-1}$. Finally, Gaussian error propagation and the Poisson statistics of the recorded coincidence count are considered to calculate ΔQ. The Possonian nature of the coincidence counts gives squared error $(\Delta c(ab|xy))^2 = c(ab|xy)$. The general expression for the experimental error is then

$$\Delta Q = \sqrt{\sum_{abxy} \left(\frac{\partial Q}{\partial c(ab|xy)} \right)^2 c(ab|xy)}, \tag{3.9}$$

and straightforward calculation leads to

$$\frac{\partial Q}{\partial c(a'b'|x'y')} = \frac{1}{\left(\sum_{ab} c(ab|x'y')\right)^2} \left[\left(S_{x'y'}^{a'b'} + \frac{1}{m}\left(S_{x'}^{a'} + S_{y'}^{b'} \right) \right) \sum_{ab} c(ab|x'y') - \sum_{ab} \left(S_{x'y'}^{ab} + \frac{1}{m}\left(S_{x'}^{a} + S_{y'}^{b} \right) \right) c(ab|x'y') \right]. \tag{3.10}$$

3.4 Tilted Bell inequality

Certification of quantum nonlocality from experimental data is a challenging task in general. In a recent work, a genuine random number generation protocol based on quantum nonlocality was experimentally demonstrated [90]. Here, authors certified randomness from a physical system prepared in quantum states with concurrences $C = 0.986$, $C = 0.835$ and $C = 0.582$. However, certification failed for $C = 0.193$ and $C = 0.375$, due to the high sensibility of certifying quantum violation under the presence of errors. The inequality in question is the tilted Bell inequality [87]. This inequality is well suited for our study, since it allows us to test our method for different levels of entanglement, from maximally entangled to separable, by just modifying the parameter α:

$$\alpha[p_A(0|0) - p_A(1|0)] + \sum_{x,y=0}^{1} \sum_{a,b=0}^{1} (-1)^{xy} [p(a = b|xy) - p(a \neq b|xy)] \leq C_\alpha, \tag{3.11}$$

where $p_A(a|x)$ is the marginal probability distribution for Alice, $C_\alpha = \alpha + 2$ and $Q_\alpha = \sqrt{8 + 2\alpha^2}$, $\alpha \in [0, 2]$. The quantum state producing the maximal violation in (3.11) is given by $|\psi(\theta)\rangle = \cos(\theta)|00\rangle + \sin(\theta)|11\rangle$, where $\theta = 2^{-1}\arcsin\left(\sqrt{\frac{1-(\alpha/2)^2}{1+(\alpha/2)^2}}\right)$. Optimal settings are provided by eigenvector bases of the following observables:

$$A_0 = \sigma_z, \quad A_1 = \sigma_x, \quad B_0 = \cos(\mu)\sigma_z + \sin(\mu)\sigma_x \quad \text{and} \quad B_1 = \cos(\mu)\sigma_z - \sin(\mu)\sigma_x$$

where $\mu = \arctan\left(\sqrt{\frac{1-(\alpha/2)^2}{1+(\alpha/2)^2}}\right)$ and $\sigma_x = \sigma_1$ and $\sigma_z = \sigma_3$ the Pauli Matrices defined in Eq. (1.9).

We tested optimization (3.6) for a photonic Bell inequality experiment, see Appendix B in Ref. [102] for experimental details. The experiment consisted of a high-quality source of polarization-entangled Bell states of the form $|\psi(\theta)\rangle$. Entanglement of this optimal quantum state can be characterized by its concurrence, given by $C = \sin(2\theta)$. By using optimization (3.6), we improve the experimental violation of the tilted Bell inequality for high concurrences and, more importantly, we successfully demonstrate quantum nonlocality for low values of concurrence, i.e. $C = 0.375$ and $C = 0.193$, something that failed to be proven when considering the tilted Bell inequality (3.11) *from the same statistical data*. In Fig. 3.1 we show the number of standard deviations of the quantum-LHV gap for the tilted Bell inequality and the one resulting from our optimization procedure (3.6) (more details in Table 3.1).

Table 3.1: Summary of results. The concurrence was obtained from quantum state tomography. Q_α refers to the experimental values of the Tilted Bell inequality. C_α is the LHV bound for each α. After implementing the method mentioned in section II, the results obtained are presented in the values Q and C.

Concurrence	0.193	0.375	0.582	0.835	0.986
Q_α	3.890	3.686	3.418	3.108	2.819
$(\pm)\Delta Q_\alpha$	0.006	0.008	0.008	0.007	0.004
C_α	3.921	3.702	3.372	2.937	2.002
$\frac{Q_\alpha - C_\alpha}{\Delta Q_\alpha}$	−4.85	−1.93	5.83	25.44	213.06
Q	1.417	1.505	1.314	1.557	1.819
$(\pm)\Delta Q$	0.001	0.007	0.007	0.006	0.004
C	1.412	1.381	1.000	1.000	1.000
$\frac{Q-C}{\Delta Q}$	3.88	16.68	44.53	89.87	213.57

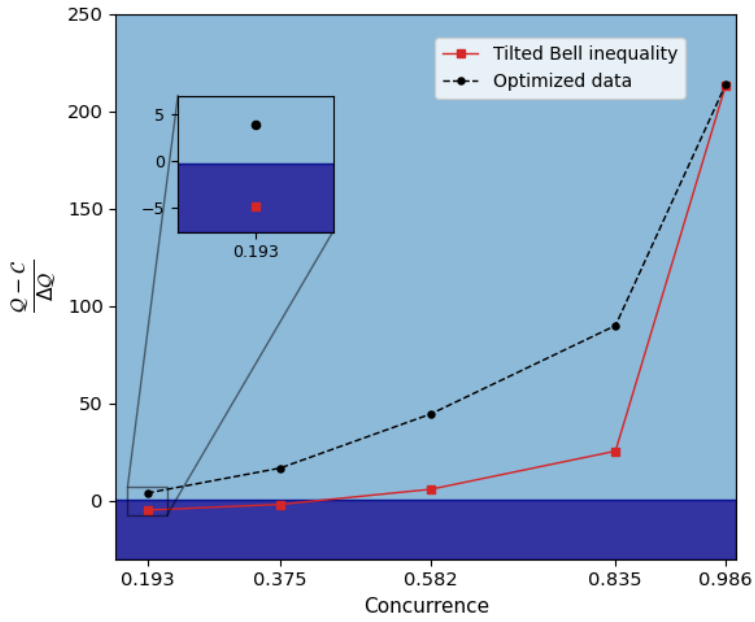


Figure 3.1: Number of standard deviation (SDN) for the gap between the LHV and quantum values, as a function of concurrence. For each value of concurrence, the optimization procedure (3.6) provides a Bell inequality that increases the number of standard deviations of the quantum-LHV value gap. SDN is calculated for two cases: the original tilted Bell inequality [red squares] and an inequality arising from optimization (3.6) [black circles]. In both cases, we consider experimental data, where quantum nonlocality can be certified in the light blue region. For concurrences $C = 0.375$ and $C = 0.193$, there is no quantum violation of the tilted Bell inequality, whereas inequalities arising from optimization (3.6) produce a violation for both concurrences. This result is obtained with the same statistical data that did not produce a violation with the tilted Bell inequality, i.e. a new experiment was not required to certify quantum nonlocality.

This result can lead to interesting practical applications. For instance, note that the amount of bits of randomness that can be extracted from the protocol [90] are identical to the number of bits that can be extracted by considering our optimized Bell inequality. This is so because both schemes consider the same experimental data, representing the same physical phenomena.

Inequalities

For each value of concurrence in table 3.1, a set of coefficients s_{xy}^{ab} was obtained by optimizing Eq. (3.6). Here we show those coefficients by writing the Bell inequalities:

Concurrence = 0.193

$$\begin{aligned}
&0.9424p(00|00) - 0.2156p(01|00) - 0.3479p(10|00) + 0.8025p(11|00)\\
&+ 0.5397p(00|01) - 0.3703p(01|01) - 0.9992p(10|01) + 0.3785p(11|01)\\
&+ 0.1495p(00|10) - 0.9801p(01|10) - 0.1586p(10|10) + 0.9992p(11|10)\\
&+ 0.0928p(00|11) + 0.9992p(01|11) + 0.4009p(10|11) - 0.9773p(11|11)\\
&- 0.3122p_A(0|0) + 0.2084p_A(1|0) \leq 1.4122.
\end{aligned}$$

Concurrence = 0.375

$$\begin{aligned}
&0.9784p(00|00) - 0.9989p(01|00) - 0.9993p(10|00) + 0.9970p(11|00)\\
&+ 0.9980p(00|01) - 0.9932p(01|01) - 0.9990p(10|01) + 0.9833p(11|01)\\
&+ 0.9996p(00|10) - 0.9967p(01|10) - 0.9807p(10|10) + 0.9965p(11|10)\\
&- 0.9976p(00|11) + 0.9936p(01|11) + 0.9827p(10|11) - 0.9997p(11|11)\\
&- 0.5965p_A(0|0) - 0.5953p_A(1|0) \leq 1.3819.
\end{aligned}$$

Concurrence = 0.582

$$\begin{aligned}
&p(00|00) - p(01|00) - p(10|00) + p(11|00) + p(00|01) - p(01|01)\\
&- p(10|01) + p(11|01) + p(00|10) - p(01|10) - p(10|10) + p(11|10)\\
&- p(00|11) + p(01|11) + p(10|11) - p(11|11) - p_A(0|0) - p_A(1|0) \leq 1
\end{aligned}$$

Concurrence = 0.835

$$\begin{aligned}
&p(00|00) - p(01|00) - p(10|00) + p(11|00) + p(00|01) - p(01|01)\\
&- p(10|01) + p(11|01) + p(00|10) - p(01|10) - p(10|10) + p(11|10)\\
&- p(00|11) + p(01|11) + p(10|11) - p(11|11) - p_A(0|0) - p_A(1|0) \leq 1.
\end{aligned}$$

Concurrence = 0.986

$$\begin{aligned}
&p(00|00) - p(01|00) - p(10|00) + p(11|00) + p(00|01) - p(01|01)\\
&- p(10|01) + p(11|01) + p(00|10) - p(01|10) - p(10|10) + p(11|10)\\
&- p(00|11) + p(01|11) + p(10|11) - p(11|11) - p_A(0|0) - p_A(1|0) \leq 1.
\end{aligned}$$

3.5 Closing the detection loophole

Since the work conducted by Aspect et al. in 1981 [103], which signified the experimental verification of Bell's work, physicists have been concerned about *loopholes* in the design of experiments to certify nonlocality. Free loopholes experiments are important, for example, in quantum key distribution protocols for secure communication [5]. Experiments to certify nonlocality has to guarantee that no messages are being interchanged between the parts involved; if they are not sufficiently distant apart, then signals can travel, at the speed of light, from one measurement location to the other and generate correlations. This is called the *locality loophole* [104].

Another source of loopholes is "missing" detections. If the *efficiency* of the detector is too low, then some of the particles, e.g. photons, might end up not being registered by the detector and the quantum correlations could be reproduced by a local hidden variables model [105]; this is called the *detection loophole*. Thus, one might wonder what is the *minimum* detection efficiency that an experiment can tolerate to consider that a Bell inequality is genuinely being violated. For example, for the CHSH inequality a detector needs to have an efficiency larger than 82.8% to close the detection loophole considering maximally entangled states [106]. A loophole-free Bell experiment was reported in 2015 [107]. In this section, we show that the optimization procedure (3.6), apart from increasing the quantum-LHV value gap, also reduces the detection efficiency required to close the detection loophole for a fixed set of statistical data.

Suppose that Alice and Bob have detector efficiencies η_A and η_B, respectively. The minimal efficiencies required to violate a Bell inequality are given by the following procedure [6]: first, a two-outcome Bell inequality has to be written in a canonical form. Namely, it has to consider only one of its outcomes per party (we choose $a = b = 0$), as we associate the other outcomes ($a = b = 1$) to the cases where detectors do not fire correctly.

To transform any bipartite Bell inequality with m settings and two outcomes to its canonical form, i.e. depending on outputs $a = b = 0$ only, the following identities have to be considered for local

$$\begin{aligned} p_A(1|x) &= 1 - p_A(0|x), \\ p_B(1|y) &= 1 - p_B(0|y), \end{aligned}$$

and joint probabilities

$$p(0,1|x,y) = p_A(0|x) - p(0,0|x,y),$$

$$\begin{aligned} p(1,0|x,y) &= p_B(0|y) - p(0,0|x,y), \\ p(1,1|x,y) &= 1 - p_A(0|x) - p_B(0|y) + p(0,0|x,y), \end{aligned}$$

for every $x, y = 0, m-1$. Applying these identities, it is simple to show that any bipartite Bell inequality (3.1) with m settings per side and two outcomes can be written as:

$$\sum_{x,y=0}^{m-1} \tilde{s}_{xy}^{00} p(0,0|x,y) + \sum_{x=0}^{m-1} \tilde{s}_{Ax}{}^{0} p_A(0|x) + \sum_{y=0}^{m-1} \tilde{s}_{By}{}^{0} p_B(0|y) \leq C, \tag{3.12}$$

where

$$\begin{aligned} \tilde{s}_{xy}^{00} &= \sum_{a,b=0}^{1} (-1)^{a+b} s_{xy}^{ab}, \\ \tilde{s}_{Ax}{}^{0} &= s_{Ax}{}^{0} - s_{Ax}{}^{1} + \sum_{y=0}^{m-1} \sum_{a=0}^{1} (-1)^{a} s_{xy}^{a1}, \\ \tilde{s}_{By}{}^{0} &= s_{By}{}^{0} - s_{By}{}^{1} + \sum_{x=0}^{m-1} \sum_{b=0}^{1} (-1)^{b} s_{xy}^{1b}. \end{aligned}$$

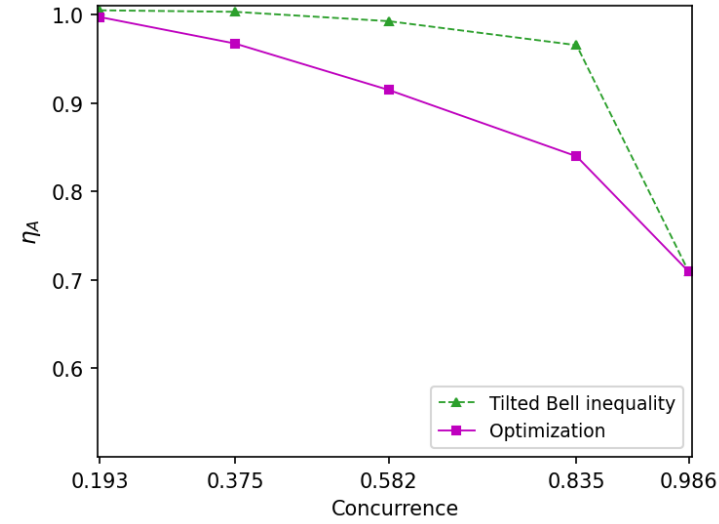
For instance, for the tilted Bell inequality (3.11), we have the following canonical form:

$$(\alpha/2-1)p_A(0|0) - p_B(0|0) + p(0,0|0,0) + p(0,0|0,1) + p(0,0|1,0) - p(0,0|1,1) \leq \tilde{C}_\alpha, \tag{3.13}$$

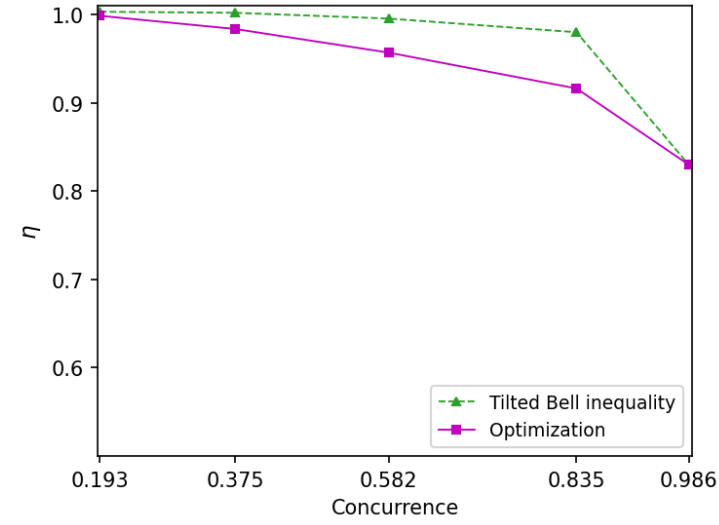
where $\tilde{C}_\alpha = (C_\alpha + \alpha - 2)/4$. Second, due to imperfect detectors, probabilities have to be transformed according to the the rule $p(0,0|x,y) \longrightarrow \eta_A \eta_B p(0,0|x,y)$, $p(0|x) \longrightarrow \eta_A\, p(0|x)$ and $p(0|y) \longrightarrow \eta_B\, p(0|y)$, for every setting x, y. Therefore, the lower bound for the minimal efficiencies required to detect a quantum violation satisfies:

$$\eta_A \eta_B \sum_{x,y=0}^{1} \tilde{s}_{xy}^{00} p(0,0|x,y) + \eta_A \sum_{x=0}^{1} \tilde{s}_{Ax}{}^{0} p_A(0|x) + \eta_B \sum_{y=0}^{1} \tilde{s}_{By}{}^{0} p_B(0|y) = C(s). \tag{3.14}$$

We observe that optimization (3.6) reduces the detection efficiency required to close the detection loophole, with respect to a target Bell inequality, for a fixed set of probability distributions $p(a,b|x,y)$, $p_A(0|x)$ and $p_B(0|y)$. Here, we show evidence of this fact by improving detection efficiencies for the experimental



(a)



(b)

Figure 3.2: Minimum efficiencies required to close the loophole for the (a) Asymmetric case, where $\eta_B = 1$, and (b) Symmetric case, where $\eta_A = \eta_B = \eta$. We compare efficiencies associated with the tilted Bell inequality and the optimization proposed in (3.6). In both cases, we consider the optimal experimental data obtained for the tilted inequality (3.13).

statistical distributions associated to the tilted Bell inequality (3.13), see Figure 3.2.

3.6 Conclusions

Violation of Bell inequalities is at the heart of quantum physics and defines a cornerstone for a wide range of quantum information protocols with real-world appeal. We have shown how an “experiment-inspired" optimization procedure can be applied to the search of Bell inequalities that increase the quantum-LHV gap with respect to a target Bell inequality, for a given set of experimental data (see Prop. 3.2.2). Furthermore, we demonstrated that the gap provided by our optimization procedure is the largest possible among the entire set of Bell inequalities having LHV equal to 1, something that can be assumed without loss of generality (see Prop. 3.2.2). When nonlocality certification from a given set of experimental data fails, our method provides a “second chance" to succeed, without requiring to perform any additional measurement. Furthermore, our method also provides a gain in the minimal detection efficiency required by a fixed statistical set, a crucial ingredient to maximize the chances to close the detection loophole. We illustrated our technique by considering experimental data associated to the maximal violation of the so-called *tilted Bell inequality*. Here, we considerably increased some previously obtained gaps, a fact that allowed us to certify nonlocality when considering weakly entangled quantum states. This certification was not possible to do with the tilted Bell inequality, i.e. the inequality that motivated the experiment, see Section 3.4. Furthermore, we also showed how the detection efficiencies required to close the detection loophole can be reduced after implementing our optimization procedure, see Section 3.5. Our technique finds application in device-independent protocols, random number generation, communication complexity and any practical application based in quantum nonlocality.

4 Quantum Marginal Problem

In this chapter, we study the quantum marginal problem. Here, we introduce an operator to tackle the problem of compatibility between parts of a quantum system and the whole. We also present an algorithm that applies iteratively this operator to find a global quantum state, if it exists, compatible with a prescribed set of quantum marginals and spectra.

4.1 Introduction

The Quantum Marginal Problem (QMP) is basically a problem of compatibility: given a set of marginals we want to find a *compatible* global density matrix, where compatible means that the marginals can be recovered from the global state using the partial trace. The fermionic version of the QMP is known as the *N-Representability problem*, terminology introduced in 1963 by A. J. Coleman [7], and represents one of the biggest challenges in quantum chemistry and quantum many body physics. Knowing the compatibility conditions would, for example, reduce the computational cost of calculating the ground state of many-body quantum systems [7]. Several works have also been devoted to establish the connections between the QMP and quantum entanglement [108, 109].

The QMP, in its most general form, is hard; it was shown in Ref. [110] that the N-Representability problem belongs to the Quantum Merlin-Arthur complete complexity class, which is the quantum analogous of the NP-complete class. Even the case of two-body marginals, of particular interest since most of Hamiltonians describing N-fermion systems, such as atoms and molecules, contain only two-body interactions [7], is believed to be intractable.

Despite the complexity of the QMP, tremendous progress has been made. Klyachko [111] completely solved the problem for the case of one-body marginals. He showed that the compatibility conditions are given by a set of linear inequalities involving the spectra of the one-body marginals. For the case of symmetric states (states describing bosons), necessary and sufficient conditions are given in Ref. [112], where the compatibility conditions are expressed as an SDP problem. Other works have focused on establishing whether a global state is uniquely determined by a given set of marginals among the pure states or

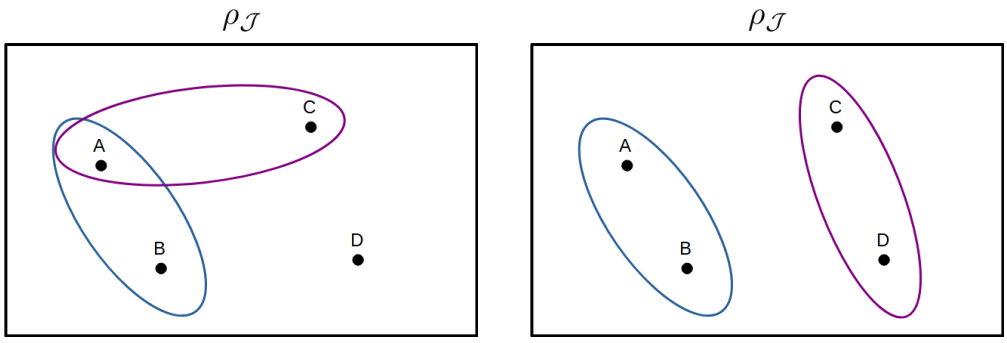


Figure 4.1: Overlapping and non-overlapping cases for a 4-body quantum system with 2-body marginals (enclosed by the coloured islands). a) *Overlapping:* $\mathcal{I}_1 \cap \mathcal{I}_2 \neq \emptyset$ with $\mathcal{I}_1 = \{A, B\}$ and $\mathcal{I}_2 = \{A, C\}$ b) *Non-overlapping:* $\mathcal{I}_1 \cap \mathcal{I}_2 = \emptyset$ with $\mathcal{I}_1 = \{A, B\}$ and $\mathcal{I}_2 = \{C, D\}$.

among all the states. For example, only two 2-body marginals are necessary to uniquely determine, in most of cases, 3-body pure states [113]. Wyderka and co-authors [114] showed that most of n-body pure states are uniquely determined, among pure states, by only three of their $(n-2)$-body marginals.

4.2 Imposing Marginals

In this section, we introduce the main tool of our study and describe its properties. We first need to distinguish two cases of the QMP: overlapping and non-overlapping. Let us consider a N-body quantum system formed by parties labeled $A, B, C, \ldots$, which form the ordered set $\mathcal{I} = \{A, B, C, \ldots\}$. Let us also consider sets $\mathcal{J}_i \subset \mathcal{I}$. We say that $\mathcal{J}_i$ and $\mathcal{J}_j$ are *overlapped* if $\mathcal{J}_i \cap \mathcal{J}_j \neq \emptyset$ for $i \neq j$. Similarly, $\mathcal{J}_i$ and $\mathcal{J}_j$ are *non-overlapped* if $\mathcal{J}_i \cap \mathcal{J}_j = \emptyset$ for $i \neq j$. In Fig. 4.1 we illustrate the overlapping and non-overlapping cases for a 4-body system.

To simplify the notation, we denote along this chapter the maximally mixed state of the system $\mathcal{J}$ by $\mathbb{I}_{\mathcal{J}}$. Now, we define the following operator:

DEFINITION 4.2.1. *Let $\rho_{\mathcal{I}}$ and $\sigma_{\mathcal{J}}$ be multipartite quantum states associated to a set of parties $\mathcal{I}$ and $\mathcal{J} \subset \mathcal{I}$, respectively. We define the marginal imposition operator as:*

$$Q_{\sigma_{\mathcal{J}}}(\rho_{\mathcal{I}}) := \rho_{\mathcal{I}} - \rho_{\mathcal{J}} + \sigma_{\mathcal{J}}, \tag{4.1}$$

where $\rho_{\mathcal{J}} = \mathrm{Tr}_{\mathcal{J}^c}[\rho_{\mathcal{I}}]$ and $\mathcal{J}^c$ denotes the complement of $\mathcal{J}$ with respect to $\mathcal{I}$.

Here, reductions are implicitly tensored with maximally mixed states, e.g.

$\sigma_{\mathcal{J}}$ means $\sigma_{\mathcal{J}} \otimes \mathbb{I}_{\mathcal{J}^c}$. As its name suggests, the operator $Q_{\sigma_{\mathcal{J}}}(\rho_{\mathcal{I}})$ imposes the marginal $\sigma_{\mathcal{J}}$ into the density matrix $\rho_{\mathcal{I}}$. To illustrate this, let us consider the case with $\mathcal{I} = \{A, B, C\}$ and $\mathcal{J} = \{A, B\}$. Thus, Eq. (4.1) becomes:

$$Q_{\sigma_{AB}}(\rho_{ABC}) = \rho_{ABC} - \rho_{AB} \otimes \mathbb{I}_C + \sigma_{AB} \otimes \mathbb{I}_C, \tag{4.2}$$

where $\rho_{AB} = \mathrm{Tr}_C[\rho_{ABC}]$. By partial tracing Eq. (4.2) over C, one obtain σ_{AB}. Let us formalize this observation in the following proposition:

PROPOSITION 4.2.1. *Let $\rho_{\mathcal{I}}$ and $\sigma_{\mathcal{J}}$ be multipartite quantum states associated to a set of parties $\mathcal{I}$ and $\mathcal{J} \subset \mathcal{I}$, respectively. Then* $\mathrm{Tr}_{\mathcal{J}^c}\left[Q_{\sigma_{\mathcal{J}}}(\rho_{\mathcal{I}})\right] = \sigma_{\mathcal{J}}$.

Proof. Let us compute $\mathrm{Tr}_{\mathcal{J}^c}\left[Q_{\sigma_{\mathcal{J}}}(\rho_{\mathcal{I}})\right]$

$$\mathrm{Tr}_{\mathcal{J}^c}\left[Q_{\sigma_{\mathcal{J}}}(\rho_{\mathcal{I}})\right] = \mathrm{Tr}_{\mathcal{J}^c}[\rho_{\mathcal{I}}] - \mathrm{Tr}_{\mathcal{J}^c}\left[\rho_{\mathcal{J}}\right] + \mathrm{Tr}_{\mathcal{J}^c}\left[\sigma_{\mathcal{J}}\right], \tag{4.3}$$

from Def. 4.2.1 $\rho_{\mathcal{J}} = \mathrm{Tr}_{\mathcal{J}^c}[\rho_{\mathcal{I}}]$. Also, $\mathrm{Tr}_{\mathcal{J}^c}\left[\rho_{\mathcal{J}}\right] = \rho_{\mathcal{J}}$ and $\mathrm{Tr}_{\mathcal{J}^c}\left[\sigma_{\mathcal{J}}\right] = \sigma_{\mathcal{J}}$. Thus, Eq. (4.3) becomes $\mathrm{Tr}_{\mathcal{J}^c}\left[Q_{\sigma_{\mathcal{J}}}(\rho_{\mathcal{I}})\right] = \rho_{\mathcal{J}} - \rho_{\mathcal{J}} + \sigma_{\mathcal{J}} = \sigma_{\mathcal{J}}$. ■

The action of $Q_{\sigma_{\mathcal{J}}}$ over $\rho_{\mathcal{I}}$ removes all the information about $\mathcal{J}$ and outputs an hermitian matrix containing the marginal $\sigma_{\mathcal{J}}$. Furthermore, the action of $Q_{\sigma_{\mathcal{J}}}$ does not affect the information about the subsystems in $\mathcal{I} \setminus \mathcal{J}$. Let us formalize this last statement in the following proposition:

PROPOSITION 4.2.2. *Let $\rho_{\mathcal{I}}$ and $\sigma_{\mathcal{J}}$ be multipartite quantum states, with $\mathcal{J} \subset \mathcal{I}$ and $\mathcal{K} \subset \mathcal{I}$ such that $\mathcal{J} \cap \mathcal{K} = \emptyset$. Therefore,* $\mathrm{Tr}_{\mathcal{K}^c}[Q_{\sigma_{\mathcal{J}}}(\rho_{\mathcal{I}})] = \mathrm{Tr}_{\mathcal{K}^c}[\rho_{\mathcal{I}}] = \rho_{\mathcal{K}}$.

Proof. Let us compute $\mathrm{Tr}_{\mathcal{K}^c}\left[Q_{\sigma_{\mathcal{J}}}(\rho_{\mathcal{I}})\right]$

$$\mathrm{Tr}_{\mathcal{K}^c}\left[Q_{\sigma_{\mathcal{J}}}(\rho_{\mathcal{I}})\right] = \mathrm{Tr}_{\mathcal{K}^c}[\rho_{\mathcal{I}}] - \mathrm{Tr}_{\mathcal{K}^c}\left[\rho_{\mathcal{J}}\right] + \mathrm{Tr}_{\mathcal{K}^c}\left[\sigma_{\mathcal{J}}\right], \tag{4.4}$$

by definition (1.14) $\mathrm{Tr}_{\mathcal{K}^c}[\rho_{\mathcal{I}}] = \rho_{\mathcal{K}}$. Also, $\mathcal{J} \cap \mathcal{K} = \emptyset$ implies that $\mathcal{J} \subseteq \mathcal{K}^c$ and therefore $\mathrm{Tr}_{\mathcal{K}^c}\left[\rho_{\mathcal{J}}\right] = \mathrm{Tr}_{\mathcal{K}^c}\left[\sigma_{\mathcal{J}}\right] = \mathbb{I}_{\mathcal{K}^c}$. Thus, (4.4) becomes $\mathrm{Tr}_{\mathcal{K}^c}[Q_{\sigma_{\mathcal{J}}}(\rho_{\mathcal{I}})] = \mathrm{Tr}_{\mathcal{K}^c}[\rho_{\mathcal{I}}] = \rho_{\mathcal{K}}$. ■

We can verify Prop. 4.2.2 for Eq. (4.2) by choosing $\mathcal{K} = \{C\}$. We obtain $\mathrm{Tr}_{AB}[\rho_{ABC}] = \mathrm{Tr}_{AB}[Q_{\sigma_{AB}}(\rho_{ABC})] = \rho_C$.

In the following proposition we show that the output of Eq. (4.1) is not always a positive semidefinite hermitian matrix (PSD). Henceforth, to simplify the notation, we will use $Q_{\sigma_{\mathcal{J}}} = Q_{\mathcal{J}}$ and $\rho = \rho_{\mathcal{I}}$.

PROPOSITION 4.2.3. *The operator* (4.1) *is not always positive semidefinite.*

Proof. Let us consider a 2-qubit quantum system. Let us define the following density matrix

$$\rho_{AB} = \begin{pmatrix} \alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta \end{pmatrix}, \tag{4.5}$$

with $\alpha + \beta = 1$ and $0 \leq \alpha, \beta \leq 1$, and calculate ρ_A from (4.5)

$$\rho_A = \mathrm{Tr}_B[\rho_{AB}] = \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix}. \tag{4.6}$$

Let us also consider the following 1-body marginal

$$\sigma_A = \begin{pmatrix} \gamma & 0 \\ 0 & \delta \end{pmatrix}, \tag{4.7}$$

with $\gamma + \delta = 1$ and $0 \leq \gamma, \delta \leq 1$. Replacing ρ_{AB}, ρ_A and σ_A in (4.1) we obtain

$$\begin{aligned} Q_A(\rho_{AB}) &= \rho_{AB} - \rho_A \otimes \frac{\mathbb{I}}{2} + \sigma_A \otimes \frac{\mathbb{I}}{2} \\ &= \frac{1}{2}\begin{pmatrix} \alpha+\gamma & 0 & 0 & 0 \\ 0 & -\alpha+\gamma & 0 & 0 \\ 0 & 0 & -\beta+\delta & 0 \\ 0 & 0 & 0 & \beta+\delta \end{pmatrix}. \end{aligned} \tag{4.8}$$

Thus, (4.8) is not positive semidefinite when either $\alpha > \gamma$ or $\beta > \delta$ ■

We address the issue of positivity in the next sections. Density matrices are trace one hermitian matrices. It turns out that (4.1) preserves this property:

PROPOSITION 4.2.4. *Let ρ and $\sigma_{\mathcal{J}}$ be multipartite quantum states, with $\mathcal{I}$ and $\mathcal{J} \subset \mathcal{I}$. The operator* (4.1) *is a trace one hermitian matrix.*

Proof. $Q_{\mathcal{J}}(\rho)$ is the sum of hermitian matrices, therefore it is also hermitian. Let us calculate the trace of (4.1):

$$\mathrm{Tr}[Q_{\mathcal{J}}(\rho)] = \mathrm{Tr}[\rho] - \mathrm{Tr}[\rho_{\mathcal{J}}] + \mathrm{Tr}[\sigma_{\mathcal{J}}], \tag{4.9}$$

but $\mathrm{Tr}[\rho] = \mathrm{Tr}[\rho_{\mathcal{J}}] = \mathrm{Tr}[\sigma_{\mathcal{J}}] = 1$. Thus, $\mathrm{Tr}[Q_{\mathcal{J}}(\rho)] = 1$. ■

Another property of operator (4.1) is that the composition of two of them commute.

PROPOSITION 4.2.5. *$Q_{\mathcal{J}_2} \circ Q_{\mathcal{J}_1}(\rho) = Q_{\mathcal{J}_1} \circ Q_{\mathcal{J}_2}(\rho)$ for any subsets $\mathcal{J}_1, \mathcal{J}_2 \subset \mathcal{I}$ and any quantum states $\rho, \sigma_{\mathcal{J}_1}$ and $\sigma_{\mathcal{J}_2}$.*

Proof. Let us first consider that the given marginals are compatible with some global density matrix σ, this is $\sigma_{\mathcal{J}_1} = \mathrm{Tr}_{\mathcal{J}_1^c}[\sigma]$ and $\sigma_{\mathcal{J}_2} = \mathrm{Tr}_{\mathcal{J}_2^c}[\sigma]$. Let us compute $Q_{\mathcal{J}_2} \circ Q_{\mathcal{J}_1}(\rho) = Q_{\mathcal{J}_2}(Q_{\mathcal{J}_1}(\rho))$

$$Q_{\mathcal{J}_2} \circ Q_{\mathcal{J}_1}(\rho) = \rho - \mathrm{Tr}_{\mathcal{J}_2^c}[\rho] - \mathrm{Tr}_{\mathcal{J}_1^c}[\rho] + \mathrm{Tr}_{\mathcal{J}_2^c}\mathrm{Tr}_{\mathcal{J}_1^c}[\rho] + \sigma_{\mathcal{J}_1} + \sigma_{\mathcal{J}_2} + \mathrm{Tr}_{\mathcal{J}_2^c}\mathrm{Tr}_{\mathcal{J}_1^c}[\sigma]. \quad (4.10)$$

Note that (4.10) is invariant under the interchange $\mathcal{J}_1 \leftrightarrow \mathcal{J}_2$. Therefore,

$$Q_{\mathcal{J}_2} \circ Q_{\mathcal{J}_1}(\rho) = Q_{\mathcal{J}_1} \circ Q_{\mathcal{J}_2}(\rho). \quad (4.11)$$

■

We can use (4.11) to prove that the composition $Q_{\mathcal{J}_m} \circ \cdots \circ Q_{\mathcal{J}_1}$ contains the set $\{\sigma_{\mathcal{J}_i}\}_{i=1}^m$ of imposed quantum marginals. To do that, we use the notation $Q_{\mathcal{J}_1,\ldots,\mathcal{J}_m} = Q_{\mathcal{J}_m} \circ \cdots \circ Q_{\mathcal{J}_1}$. Here, we are considering overlapping and non-overlapping marginals.

PROPOSITION 4.2.6. *Given a set $\{\sigma_{\mathcal{J}_i}\}_{i=1}^m$ of quantum marginals such that $\mathrm{Tr}_{\mathcal{J}_i^c}[\sigma] = \sigma_{\mathcal{J}_i}$ for a density matrix σ, then the composition*

$$Q_{\mathcal{J}_1,\ldots,\mathcal{J}_m}(\rho), \quad (4.12)$$

satisfy $\mathrm{Tr}_{\mathcal{J}_i^c}\left[Q_{\mathcal{J}_1,\ldots,\mathcal{J}_m}(\rho)\right] = \sigma_{\mathcal{J}_i}$, for $i = 1 \ldots m$.

Proof. By taking $\mathcal{J} = \mathcal{J}_2$ and $\rho_{\mathcal{I}} = Q_{\mathcal{J}_1}(\rho)$ from Prop. 4.2.1, we know that $Q_{\mathcal{J}_2}(Q_{\mathcal{J}_1}(\rho))$ satisfies

$$\mathrm{Tr}_{\mathcal{J}_2^c}\left[Q_{\mathcal{J}_2}(Q_{\mathcal{J}_1}(\rho))\right] = \sigma_{\mathcal{J}_2}.$$

Also, because of Prop. 4.2.5 $\mathrm{Tr}_{\mathcal{J}_1^c}\left[Q_{\mathcal{J}_2}(Q_{\mathcal{J}_1}(\rho))\right] = \mathrm{Tr}_{\mathcal{J}_1^c}\left[Q_{\mathcal{J}_1}(Q_{\mathcal{J}_2}(\rho))\right] = \sigma_{\mathcal{J}_1}$. In general,

$$Q_{\mathcal{J}_1,\ldots,\mathcal{J}_m} = Q_{\mathcal{J}_m}(Q_{\mathcal{J}_1,\ldots,\mathcal{J}_{m-1}}) = \ldots = Q_{\mathcal{J}_{m-1}}(Q_{\mathcal{J}_1,\ldots,\mathcal{J}_m}),$$

and therefore $\mathrm{Tr}_{\mathcal{J}_i^c}\left[Q_{\mathcal{J}_1,\ldots,\mathcal{J}_m}(\rho)\right] = \sigma_{\mathcal{J}_i}$, for $i = 1, \ldots, m$. ■

Derivation of expressions

Here, we show some expressions obtained from applying (4.12). We consider the set of all k-body marginals for a global state formed by N bodies. For example, for $N = 3$, with labels in $\mathcal{I} = \{A, B, C\}$, the possible reductions to 2 bodies are σ_{AB}, σ_{AC} and σ_{BC}. Thus, from (4.12)

$$\begin{aligned} Q_{AB,AC,BC}(\rho_{ABC}) = \rho_{ABC} - (\rho_{AB} + \rho_{AC} + \rho_{BC}) + (\rho_A + \rho_B + \rho_C) \\ + (\sigma_{AB} + \sigma_{AC} + \sigma_{BC}) - (\sigma_A + \sigma_B + \sigma_C), \end{aligned} \tag{4.13}$$

with $\rho_{ABC} = \rho$.

Expressions obtained in this way can be simplified by choosing $\rho = \mathbb{I}_{\mathcal{I}}$. Next we list some of them for different values of N and k:

- $N = 2, k = 1$

$$Q_{A,B}(\mathbb{I}_{\mathcal{I}}) = \sigma_A + \sigma_B - \mathbb{I}_{\mathcal{I}}. \tag{4.14}$$

- $N = 3, k = 1$

$$Q_{A,B,C}(\mathbb{I}_{\mathcal{I}}) = \sigma_A + \sigma_B + \sigma_C - 2\mathbb{I}_{\mathcal{I}}. \tag{4.15}$$

- $N = 4, k = 1$

$$Q_{A,B,C,D}(\mathbb{I}_{\mathcal{I}}) = \sigma_A + \sigma_B + \sigma_C + \sigma_D - 3\mathbb{I}_{\mathcal{I}}. \tag{4.16}$$

- $N = 3, k = 2$

$$Q_{AB,AC,BC}(\mathbb{I}_{\mathcal{I}}) = \mathbb{I}_{\mathcal{I}} + (\sigma_{AB} + \sigma_{AC} + \sigma_{BC}) - (\sigma_A + \sigma_B + \sigma_C). \tag{4.17}$$

- $N = 4, k = 2$

$$\begin{aligned} Q_{AB,\ldots,CD}(\mathbb{I}_{\mathcal{I}}) = 3\mathbb{I}_{\mathcal{I}} + (\sigma_{AB} + \sigma_{AC} + \sigma_{AD} + \sigma_{BC} + \sigma_{BD} + \sigma_{CD}) \\ - 2(\sigma_A + \sigma_B + \sigma_C + \sigma_D). \end{aligned} \tag{4.18}$$

- $N = 5, k = 2$

$$\begin{aligned} Q_{AB,\ldots,DE}(\mathbb{I}_{\mathcal{I}}) = 6\mathbb{I}_{\mathcal{I}} + (\sigma_{AB} + \sigma_{AC} + \sigma_{AD} + \sigma_{AE} + \sigma_{BC} + \sigma_{BD} + \sigma_{BE} \\ + \sigma_{CD} + \sigma_{CE} + \sigma_{DE}) - 3(\sigma_A + \sigma_B + \sigma_C + \sigma_D + \sigma_E). \end{aligned} \tag{4.19}$$

- $N = 4, k = 3$

$$Q_{ABC,\ldots,BCD}(\mathbb{I}_{\mathcal{I}}) = (\sigma_{ABC} + \sigma_{ABD} + \sigma_{ACD} + \sigma_{BCD}) - (\sigma_{AB} + \sigma_{AC} + \sigma_{AD} + \sigma_{BC} + \sigma_{BD} + \sigma_{CD}) + (\sigma_A + \sigma_B + \sigma_C + \sigma_D) - \mathbb{I}_{\mathcal{I}} \tag{4.20}$$

Some of the expressions above suggest the following proposition:

PROPOSITION 4.2.7. *Let $\mathcal{I} = \{A, B, C, \ldots, \mathcal{I}_N\}$ and $\mathcal{J} = \{AB, AC, \ldots, \mathcal{J}_m\}$ be the sets of labels associated to all one and two party marginals, respectively, with $|\mathcal{I}| = N$ and $|\mathcal{J}| = 2^{-1}N!/(N-2)!$. Given the set $\{\sigma_j\}_{j=1}^{m}$ of all possible bipartite quantum marginals, the following formula holds*

$$Q_{AB,AC\ldots,\mathcal{J}_m}(\mathbb{I}_{\mathcal{I}}) = \sum_{j\in\mathcal{J}} \sigma_j - (N-2)\sum_{i\in\mathcal{I}} \sigma_i + \frac{(N-1)(N-2)}{2}\mathbb{I}_{\mathcal{I}}. \tag{4.21}$$

Proof. Let us show by induction this proposition. We saw above that (4.21) holds for $N = 3, 4$ and 5. Let us suppose that Eq. (4.21) is valid for any N and show that it is also true for $N + 1$. Let us consider an $(N + 1)$-qudit system and all its 2-party marginals and let $\mathcal{I}'$ and $\mathcal{J}'$ be the ordered sets of labels for the $N + 1$ qudits and the $2^{-1}(N + 1)!/(N - 1)!$ 2-body marginals, respectively. In general, we can write

$$Q_{AB,AC\ldots,\mathcal{J}'_\ell}(\mathbb{I}_{\mathcal{I}'}) = \sum_{j\in\mathcal{J}'} \alpha_j\sigma_j + \sum_{i\in\mathcal{I}'} \beta_i\sigma_i + \gamma\mathbb{I}_{\mathcal{I}'}. \tag{4.22}$$

Because of the commutation property (see Prop. 4.2.5), the 2-body marginals in (4.22) have to appear equally weighted, this is $\alpha = \alpha_j$. Also, since the 1-body marginals result from partial tracing an equal number of 2-body marginals, $\beta = \beta_i$. Thus, Eq. (4.22) becomes

$$Q_{AB,AC\ldots,\mathcal{J}'_\ell}(\mathbb{I}_{\mathcal{I}'}) = \alpha\sum_{j\in\mathcal{J}'} \sigma_j + \beta\sum_{i\in\mathcal{I}'} \sigma_i + \gamma\mathbb{I}_{\mathcal{I}'}. \tag{4.23}$$

Let us take partial trace of (4.23) over the last element in the set $\mathcal{I}'$, denoted $\mathcal{I}'_{N+1}$. We obtain

$$\begin{aligned} \mathrm{Tr}_{\mathcal{I}'_{N+1}}\left[Q_{AB,AC,\ldots,\mathcal{J}'_\ell}(\mathbb{I}_{\mathcal{I}'})\right] &= \alpha\sum_{j\in\mathcal{J}} \sigma_j + \alpha\sum_{i\in\mathcal{I}} \sigma_i + \beta\sum_{i\in\mathcal{I}} \sigma_i + \beta\mathbb{I}_{\mathcal{I}} + \gamma\mathbb{I}_{\mathcal{I}}, \\ &= \alpha\sum_{j\in\mathcal{J}} \sigma_j + (\alpha+\beta)\sum_{i\in\mathcal{I}} \sigma_i + (\beta+\gamma)\mathbb{I}_{\mathcal{I}}. \end{aligned} \tag{4.24}$$

Using the fact that $\mathrm{Tr}_{\mathcal{I}'_{N+1}}\left[Q_{AB,AC,\ldots,\mathcal{J}'_\ell}(\mathbb{I}_{\mathcal{I}'})\right] = Q_{AB,AC,\ldots,\mathcal{J}_m}(\mathbb{I}_{\mathcal{I}})$, we obtain $\alpha = 1$, $\beta = -(N+1-2)$ and $\gamma = (N+1-1)(N+1-2)/2$. Thus, Eq. (4.22) becomes

$$Q_{AB,AC\ldots,\mathcal{J}'_\ell}(\mathbb{I}_{\mathcal{I}'}) = \sum_{j\in\mathcal{J}'}\sigma_j - (N+1-2)\sum_{i\in\mathcal{I}'}\sigma_i + \frac{(N+1-1)(N+1-2)}{2}\mathbb{I}_{\mathcal{I}'}, \quad (4.25)$$

and therefore (4.21) holds.

■

4.3 Numerical Study

In the previous section we showed that $Q_{\mathcal{J}_1,\ldots,\mathcal{J}_m}(\rho_0)$ is a trace one hermitian matrix containing the imposed marginals. Numerically, we have found that for some set $\{\sigma_{\mathcal{J}_i}\}_{i=1}^m$ and initial state ρ_0, the composition $Q_{\mathcal{J}_1,\ldots,\mathcal{J}_m}(\rho_0)$ is PSD. However, finding the correct initial state ρ_0, or even more, determining whether it exists for a given set of marginals, is not easy. Here, we want to show the occurrence of PSD matrices when choosing $\rho_0 = \mathbb{I}_{\mathcal{I}}$ in Eq. (4.12) for different number of parties and local dimension d. First, we define the *generator state*:

DEFINITION 4.3.1 (Generator state). *Given a set $\{\mathcal{J}_i\}$ of m labels, such that $\mathcal{J}_i \subset \mathcal{I}$, with $\mathcal{I}$ an alphabet of N symbols, a quantum state ρ_{gen} is called a generator state if $\mathrm{Tr}_{\mathcal{J}_i^c}[\rho_{gen}] = \sigma_{\mathcal{J}_i}$, for all $i = 1,\ldots,m$.*

For this study we considered $|\mathcal{J}_i| = k$, for $i = 1,\ldots,m$. This is, all the quantum marginals are formed by k bodies, each of them with local dimension d. Thus, the number of possible k-body marginals is

$$M(N,k) = \frac{N!}{k!(N-k)!}. \quad (4.26)$$

In Fig. 4.2 we show the number of PSD matrices (denoted *NPM*) resulting from $Q_{\mathcal{J}_1,\ldots,\mathcal{J}_m}(\mathbb{I}_{\mathcal{I}})$ vs the number m of quantum marginals, obtained from numerical simulations for different values of N and k. In each of the plots in Fig. 4.2 we compare results for $d = 2$ (magenta solid line) and $d = 3$ (blue dotted line). We see that NPM grows with the local dimension d (see also Fig. 4.3, which shows NPM vs d). We found from these numerical simulations that for mixed states ρ_{gen} chosen at random according to the Hilbert-Schmidt measure, $Q_{\mathcal{J}_1,\ldots,\mathcal{J}_m}(\mathbb{I}_{\mathcal{I}})$ outputs a full rank density matrix containing the given marginals, which turned out to be different to the generator state (mixed states are not uniquely determined by their quantum marginals). Thus, if it is known in

advance that the marginals are compatible with a full rank global state, then we can easily find, with high probability, a global state by simply applying the operator $Q_{\mathcal{J}_1,\ldots,\mathcal{J}_m}$ to $\mathbb{I}_{\mathcal{I}}$. The chances are even greater for large local dimensions d, as numerical simulations suggest. The probability becomes smaller for quantum states close to the pure states, and for pure states ρ_{gen} randomly drawn from the Haar measure [115], the occurrence of PSD matrices is zero for most of cases.

4.4 Algorithm

In section 4.3, we saw that given a set of all possible k-body marginals calculated from a mixed generator state, the chances of finding a full rank PSD matrix compatible with those marginals from the composition $Q_{\mathcal{J}_1,\ldots,\mathcal{J}_m}(\mathbb{I}_{\mathcal{I}})$ are, in many cases, high. In this section, we study a more restrictive version of the same problem. Let us consider an N-body quantum system of dimensions d each. Given the set $\{\sigma_{\mathcal{J}_i}\}_{i=1}^{m}$ of quantum marginals describing m reduced part of the system, each of them formed by k_i bodies, with $k_i = |\mathcal{J}_i|$, and given $\vec{\lambda} = (\lambda_0, \ldots, \lambda_{d^N-1})$, with $\lambda_i \geq 0$ and $\sum_{i=0}^{d^N} \lambda_i = 1$, we want to study the following question:

> *Given a set $\{\sigma_{\mathcal{J}_i}\}_{i=1}^{m}$ of quantum marginals and spectra $\vec{\lambda}$, is there any density matrix ρ compatible with this information?*

Here we are considering both, overlapping and non-overlapping sets $\mathcal{J}_i$. Instead of the spectra, we may also consider rank constraint. Thus, the problem consists in finding a density matrix of rank equal or smaller than r, if it exists, compatible with the given marginals.

In Prop. 4.2.6 was shown that applying the composition $Q_{\mathcal{J}_1,\ldots,\mathcal{J}_m}(\rho)$ produces a trace one hermitian matrix ρ', compatible with the m marginals. Also, we saw that ρ' is not necessarily PSD, which can be checked by computing its *spectral decomposition* and looking for negative eigenvalues. The spectral decomposition of ρ' consists of the following factorization

$$\rho' = UDU^{\dagger}, \tag{4.27}$$

with U a unitary matrix and $D = diag(\mu_0, \ldots, \mu_{d^N-1})$ a diagonal matrix whose entries are the *eigenvalues* of ρ'. We can *impose* the prescribed eigenvalues $\vec{\lambda}$ into ρ' simply by substituting $\Lambda = diag(\lambda_0, \ldots, \lambda_{d^N-1})$ in place of D. This is

$$\rho' \longrightarrow \rho'' = U\Lambda U^{\dagger}. \tag{4.28}$$

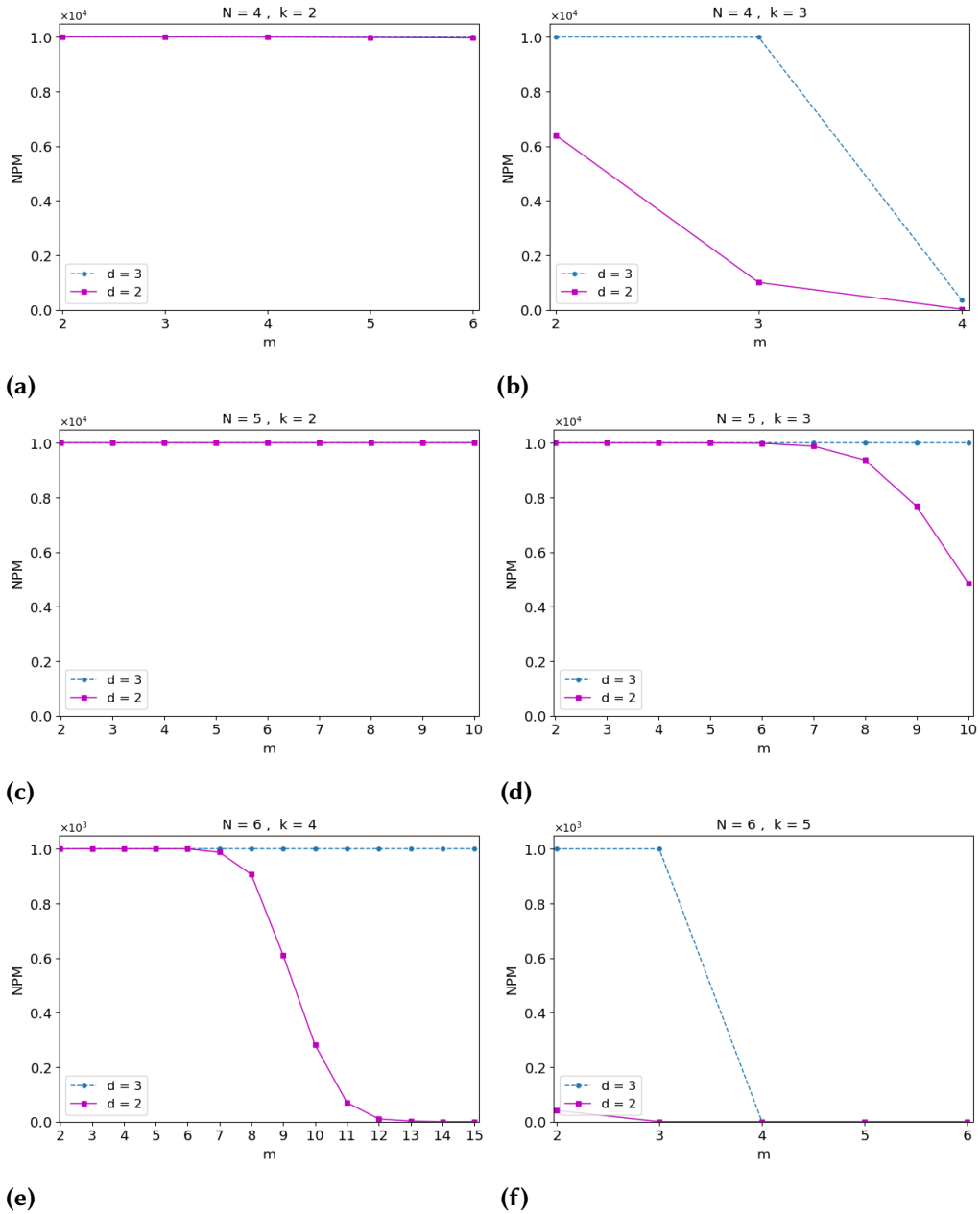


Figure 4.2: *Number of positive semidefinite matrices (NPM)* vs the number m of quantum marginals, for different values of N, k and d, obtained from the composition $Q_{\mathcal{J}_1,\ldots,\mathcal{J}_m}(\mathbb{I}_I)$. These results were obtained from a 10000 (1000 for the cases with $N = 6$) full rank generator states drawn at random under the Hilbert-Schmidt measure [116]. For each ρ_{gen}, m out of $M(N,k)$ possible quantum marginals are chosen at random.

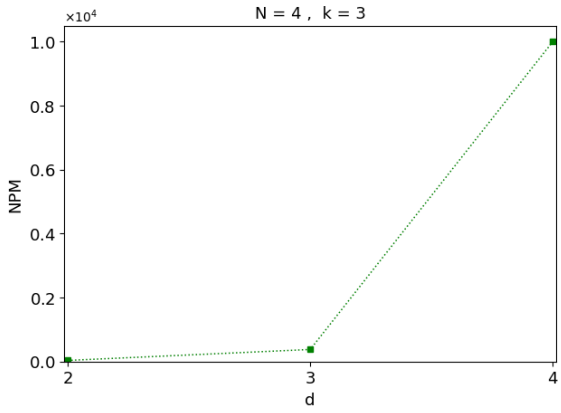


Figure 4.3: NMP vs d considering all the possible 3-body marginals. For $d = 2$, only 35 out of 10000 are PSD. For $d = 4$, all the output matrices are PSD.

For the rank constraint case, we only need to keep the r largest positive eigenvalues, set the rest of them to zero and then normalize the resulting matrix. Either way, *part* of the information that was previously imposed about the marginals is lost; this is, we will probably find that $\sigma_{\mathcal{J}_i} \neq \mathrm{Tr}_{\mathcal{J}_i^c}[\rho'']$. Then, we can apply $Q_{\mathcal{J}_1,\ldots,\mathcal{J}_m}(\rho'')$ to impose the marginals $\{\sigma_{\mathcal{J}_i}\}_{i=1}^m$ in ρ'', but it will cause the loss of part of the previously imposed eigenvalues. Intuitively, we realize that the loss of the marginals and spectra will reduce each time the whole process described above is repeated, until it finally converges to a matrix, if it exists, with the prescribed marginals and spectra. We summarize this procedure in algorithm 2.

Algorithm 2: QMP algorithm.

Input: $\{\sigma_{\mathcal{J}_i}\}_{i=1}^m$ and $\vec{\lambda}$ (or prescribed rank)

Output: $\rho'' \in B(\mathcal{H})$ with spectra $\vec{\lambda}$, satisfying $\sigma_{\mathcal{J}_i} = \mathrm{Tr}_{\mathcal{J}_i^c}[\rho'']$

ρ_0 (Random density matrix of dimension d^N)

$n = 0$

repeat

$\rho' \leftarrow Q_{\mathcal{J}_1,\ldots,\mathcal{J}_m}(\rho_0) = UDU^\dagger$

$\rho'' \leftarrow U\Lambda U^\dagger$

$\rho_0 \leftarrow \rho''$

$n \leftarrow n + 1$

until $\mathcal{D}_T \leq \epsilon$ or $n =$ Max number of iterarions

To study the convergence of algorithm 2 we consider the following metrics:

i) *Marginals Distance* $\mathcal{D}_M$*:*

$$\mathcal{D}_{\mathcal{M}} = \left(\frac{1}{m} \sum_{i=1}^{m} \mathfrak{D}(\sigma_{\mathcal{J}_i}, \sigma''_{\mathcal{J}_i})^2 \right)^{1/2}, \tag{4.29}$$

where $\mathfrak{D}(\sigma_{\mathcal{J}_i}, \sigma''_{\mathcal{J}_i}) = \sqrt{\mathrm{Tr}\left[(\sigma_{\mathcal{J}_i} - \sigma''_{\mathcal{J}_i})(\sigma_{\mathcal{J}_i} - \sigma''_{\mathcal{J}_i})^\dagger\right]}$ is the Hilbert-Schmidt distance between the given marginal $\sigma_{\mathcal{J}_i}$ and $\sigma''_{\mathcal{J}_i} = \mathrm{Tr}_{\mathcal{J}_j^c}(\rho'')$.

ii) *Spectral Distance* $\mathcal{D}_\lambda$*:*

$$\mathcal{D}_\lambda = \|\vec{\mu} - \vec{\lambda}\|, \tag{4.30}$$

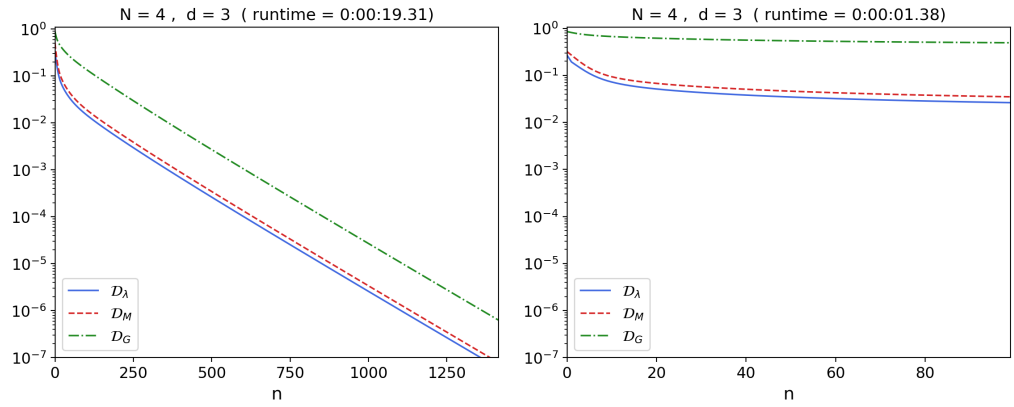
where $\mathcal{D}_\lambda$ is the euclidean distance between $\vec{\mu} = (\mu_0, \ldots, \mu_{d^N-1})$ and $\vec{\lambda} = (\lambda_0, \ldots, \lambda_{d^N-1})$.

iii) *Overall Distance* $\mathcal{D}_T$*:*

$$\mathcal{D}_T = \sqrt{\mathcal{D}_M^2 + \mathcal{D}_\lambda^2}. \tag{4.31}$$

$\mathcal{D}_M$ and $\mathcal{D}_\lambda$ tell us how close the algorithm is getting to a state with the prescribed marginals and spectra, respectively. Algorithm 2 will run until the overall distance $\mathcal{D}_T$ is smaller than the given tolerance ϵ, or until an specified number of iterations is reached.

In Fig. 4.4 (a) we show the convergence for a pure 4-party density matrix with local dimensions $d = 3$. The vertical axis corresponds to the metrics, given in log scale, whereas the horizontal axis shows the number n of iterations. In addition to $\mathcal{D}_M$ and $\mathcal{D}_\lambda$, the Hilbert-Schmidt distance $\mathcal{D}_G$ between ρ'' and ρ_{gen} is shown. We see that $\mathcal{D}_G$ tends to zero, which means that ρ'' approaches to ρ_{gen} in each iteration. Since, in this case, the marginals are compatible only with the generator state, ρ'' will always approach to ρ_{gen} for any initial seed ρ_0. Fig. 4.4 (a) shows the solution when $r = 1$ and Fig. 4.4 (b) for $r = 2$. Fig. 4.4 (b) reflects the fact that a rank-2 density matrix with the same marginals than a rank-1 ρ_{gen} does not exist. The nonexistence of this state produces instability in the algorithm, as the rank-2 restriction puts ρ'' away from a density matrix with the desired marginals. Despite the fact that a rank-1 solution exists, i.e. the generator state, it cannot be reached when considering a seed taken at random among the entire set of density matrices. Such a kind of instability is only observed when there is no quantum state of rank r with the imposed marginals.



(a) **(b)**

Figure 4.4: Convergence of a 4-party case, each party of dimension $d = 3$. The prescribed marginals consist of all possible reductions to 2 parties calculated from a pure generator state ρ_{gen}. Here, $\mathcal{D}_\lambda$, $\mathcal{D}_M$ and $\mathcal{D}_G$ vs n (the number of iterations) were plotted. We show the convergence when a) a rank one is prescribed and b) the behavior of algorithm 2 when rank two is prescribed. $\mathcal{D}_G$ is the Hilbert-Schmidt distance between ρ_{gen} and ρ''. The runtime is given in format *hh:mm:ss*.

The input marginals can have distinct number of parties. In Fig. 4.5 (a) we show the convergence of a 5-party density matrix ρ_{ABCDE} of rank 4, with inputs marginals σ_{ABD} and σ_{BC}, which were computed from a full rank generator state of 5 parties and local dimensions $d = 3$. The distance $\mathcal{D}_G$ experiences very small oscillations (unnoticeable in the plot) around some value, but the closer $\mathcal{D}_T$ gets to zero the smaller the amplitude of these oscillations.

In Fig. 4.5 (b) the convergence for a case with prescribed spectra is shown. The eigenvalues and the marginals $\sigma_{AC}, \sigma_{AD}, \sigma_{BC}$ and σ_{CD} were computed from a full rank mixed ρ_{gen}. Algorithm 2 converges to a state compatible with the inputs, but the state found is different to ρ_{gen}.

In general, there are some seeds ρ_0 for which algorithm 2 converges in fewer iterations than for other seeds. However, to determine the specific conditions on ρ_0 for faster convergence is not easy.

Absolutely Maximally Entangled States

A pure quantum state of N parties with local dimension d is called *Absolutely Maximally Entangled (AME)*, denoted $AME(N, d)$, if all its possible $\lfloor N/2 \rfloor$-body marginals are maximally mixed , with $\lfloor \ldots \rfloor$ the *floor* function. AME states have been used for quantum error-correcting codes [117], quantum secret sharing

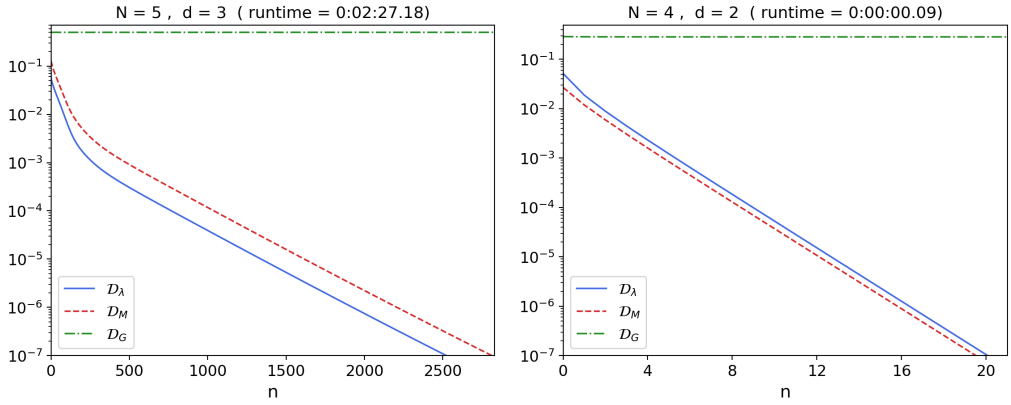


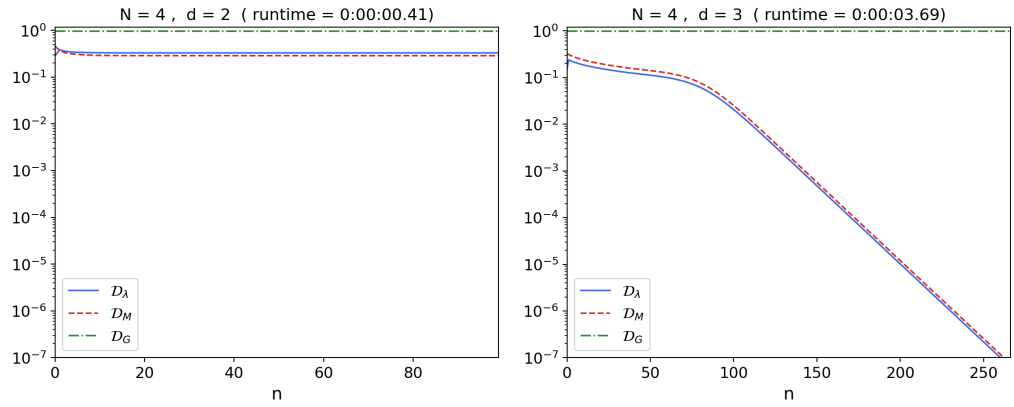
Figure 4.5: a) Convergence of a 5-body density matrix ρ_{ABCDE} of rank four, with input marginals σ_{ABD} and σ_{BC} calculated from a full rank generator state, and local dimension $d = 3$. b) Convergence of algorithm 2 for a prescribed spectra case. The prescribed eigenvalues are the same as those of the generator sate.

[118] and teleportation protocols [119]. The question of whether AME states exist for given N and d is still an open problem. A summary of the existence of AME states is given in [120].

Here, we test algorithm 2 for some AME states. In Fig. 4.6 (a) we show the case with $N = 4$ and $d = 2$. As expected, it cannot find an AME(4,2) state; Higuichi et al. showed in [121] that such a state does not exist. The convergence of the AME(4,3) is shown in 4.6 (b). The *plateau* seen in this last case is very common for AME states. For more parties and larger local dimensions, this plateau is much more prolonged, resulting in very large runtimes. For example, for cases such as the AME(4,5) and the AME(4,6), we allowed the algorithm to run for about two weeks, apparently going under the plateau stage, but it was unable to converge. In Fig. 4.7 (a) (solid line) the convergence for the AME(4,4) is shown; this case also exhibits a plateau. It took to the algorithm about 6 and half minutes to find the AME(4,4), whereas for other seeds it was unable to converge to this quantum state after 12 hours of running.

Acceleration of the algorithm

Algorithm 2 is a heuristic approach, which can be seen as a *fixed point algorithm.* A well-known algorithm for fixed point problems is the one developed by Halpern [74]. Let T be a nonexpansive mapping on a Hilbert space $\mathcal{H}$. For



(a) **(b)**

Figure 4.6: a) 100 iterations of algorithm 2 considering all the possible 2-body marginals, all of them equal to $\mathbb{I}/4$. In this case, the algorithm is trying to find a 4-parties pure quantum state, with local dimension $d = 2$, compatible with the given marginals; such a state does not exist. b) Convergence for the AME(4,3) state.

$x_0 \in \mathcal{H}$, the Halpern algorithm consists of the sequence

$$x_{n+1} := \alpha_n x_0 + (1 - \alpha_n) T(x_n), \quad n \in \mathbb{N}, \tag{4.32}$$

where $\alpha_n \in \mathbb{R}$ satisfies the conditions:

$$lim_{n\to\infty}\alpha_n = 0, \quad \sum_{n=0}^{\infty} \alpha_n = \infty \quad \text{and} \quad \sum_{n=0}^{\infty} |\alpha_{n+1} - \alpha_n| < \infty. \tag{4.33}$$

Halpern showed that (4.32) strongly converges to a fixed point. In Ref. [122] an strategy to accelerate the Halpern algorithm was introduced, which consists of the modified sequence:

$$\begin{aligned} z_{n+1} &:= \frac{1}{\alpha}(T(x_n) - x_n) + \beta_n z_n, \\ y_n &:= x_n + \alpha z_{n+1}, \\ x_n &:= \mu\alpha_n x_0 + (1 - \mu\alpha_n) y_n. \end{aligned} \tag{4.34}$$

with $z_0 = \alpha^{-1}(T(x_0) - x_0)$, $\mu \in (0, 1]$ and $\alpha > 0$. Besides conditions (4.33), $\beta_n \leq \alpha_n^2$ also has to be satisfied. When $\beta_n = 0$ and $\mu = 1$, (4.34) becomes (4.32).

We took some of the ideas from Ref. [122] and implemented them to accelerate algorithm 2; we show this changes in the algorithm 3. Any strategies to

update α_n and β_n are possible, as long as they satisfy the conditions above. For our numerical simulations we chose $\alpha_n = (n/10^5 + 1)^{-\alpha}$, which is a slightly modified version of α_n from a numerical example in Ref. [122], and $\beta_n = \alpha_n^2$. With $\beta_n = 0$, $\mu = 1$ and $\alpha_n = 1$, algorithm 3 reduces to algorithm 2.

In Fig. 4.7 we compare algorithms 2 and 3 when starting from the same ρ_0. The lines correspond to the overall distance $\mathcal{D}_T$. The dashed and dotted lines show the convergence of algorithm 3 for different choices of α and μ. Fig. 4.7 (a) corresponds to the AME(4,4); at first, the lines are converging at about the same rate, but suddenly algorithm 3 starts decaying faster. Fig. 4.7 (b) corresponds to a pure state, randomly drawn from the Haar measure.

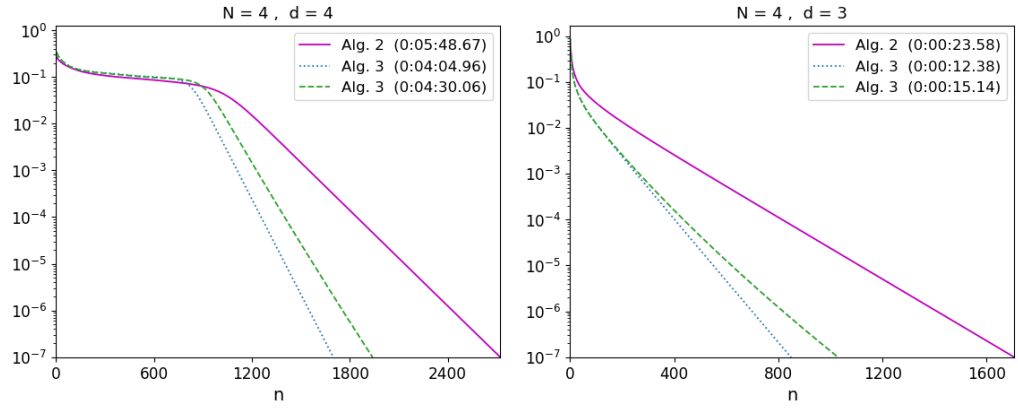
Algorithm 3: QMP algorithm.

Input: $\{\sigma_{\mathcal{J}_i}\}_{i=1}^m$, $\vec{\lambda}$ (or prescribed rank), α and μ.

Output: $\rho'' \in B(\mathcal{H})$ with spectra $\vec{\lambda}$, satisfying $\sigma_{\mathcal{J}_i} = \mathrm{Tr}_{\mathcal{J}_i^c}[\rho'']$.

ρ_0 (Random density matrix of dimension d^N)
$n = 0$
Repeat
  $\alpha_n \leftarrow (n/10^5 + 1)^{-\alpha}$
  $\beta_n \leftarrow \alpha_n^2$
  For $i = 1, \ldots, m$:
    $z_i \leftarrow \alpha^{-1}\left(Q_{\mathcal{J}_i}(\rho_0) - \rho_0\right)$
    $z_{i+1} \leftarrow z_i + \beta_n z_i$
    $y_i \leftarrow \rho_0 + \alpha z_{i+1}$
    $\rho_0 \leftarrow \mu\alpha_n\rho_0 + (1 - \mu\alpha_n)y_i$
  end for
  $\rho' \leftarrow \rho_0 = UDU^\dagger$
  $\rho'' \leftarrow U\Lambda U^\dagger$
  $\rho_0 \leftarrow \rho''$
  $n \leftarrow n + 1$
until $\mathcal{D}_T \leq \epsilon$ or $n =$ Max number of iterarions

The parameters α, β_n, μ and α_n can be sensitive to the local dimension, the number of bodies in the global system and the number of bodies in the marginals. Thus, one might have to tune these parameters for different values of N, d and k. Also, the performance for a set of parameters might depend on the initial seed ρ_0. Nonetheless, for given N, d and k, we have noticed improvement in the performance for a wide range of quantum states with the same choice of parameters. The algorithms were implemented in Python, see Appendix C, and



(a) **(b)**

Figure 4.7: Comparison between algorithm 2 (dashed line) and 3 (dashed and dotted lines). For the dotted line $\alpha = 1$ and $\mu = 10^{-10}$ and for the dashed line $\alpha = 50$ and $\mu = 5 \times 10^{-5}$. a) Convergence for the AME(4,4) and b) convergence for a pure random quantum state, with $N = 4$ and $d = 3$.

are available on Github [123]. All the calculations in this chapter were made on an AMD Ryzen 5 4500U laptop with 16GB of RAM.

4.5 Conclusions

In this chapter, we addressed the problem of compatibility in the Quantum Marginal Problem. We introduced an operator to impose a set of quantum marginals in a global matrix. We saw that, in many cases, for quantum marginals generated from a full rank mixed state, drawn from a uniform distribution, the composition of these operators outputs a full rank density matrix compatible with the marginals. We applied the operator to develop a heuristic algorithm for finding a global density matrix, if exists, compatible with a prescribed spectra and marginals. For quantum states randomly taken from uniform distributions, numerical simulations shows that the algorithm is able to find a solution for most of cases. It was also able to find many cases of AME states whose existence is known. However, for large number of parties and local dimension, the algorithm failed to find AME states in a reasonable amount of time, even though we showed that it is possible to accelerate it. For open AME cases (see Ref. [120]), such as the AME(8,7) and AME(11,3), it was not possible to run the algorithm due to the required memory resources.

Appendix

A: Codes for the quantum state estimation algorithm

Here we show the codes for the quantum state estimation algorithm presented in 2. The algorithm was implemented in Python and the code is available in Github [77].

```
import numpy as np
import time
from scipy.optimize import root
import enum
import qiskit.quantum_info as qi
import itertools as it
import matplotlib.pyplot as plt
import functools as ft
import random

def QSE(dimension, Measurements,Probabilities,maxNumIterations,min_hsDistance):

    """
    Estimate a density matrix for "Probabilities" obtained from "Measurements"

    """

    start_time_pio = time.time()
    rho_0 = np.eye(dimension)/dimension
    hsDistance = 2
    n = 0
    rho_estimate = 0
    while hsDistance > min_hsDistance and n < maxNumIterations:
        rho_1 = rho_0
        rho_0 = applyCompositionT(rho_1,Measurements,Probabilities)
        rho_0 = projectToDensityOperator(rho_0,dimension)
        rho_estimate += rho_0
        hsDistance = HS_distance(rho_0,rho_1)
        n += 1

    rho_estimate = rho_estimate/np.trace(rho_estimate)
    dt_pio = time.time() - start_time_pio

    return rho_estimate, dt_pio


def bootStrapping(rho_target,dimension,measurements,numOfSeeds,numOfCopies):

    Fidelities = []
    white_noise_level = 0.1
```

```
    rhoNoisy = (1 - white_noise_level)*rho_target + white_noise_level*np.identity(dimension)/dimension
    Probabilities = probability_distributions(measurements, dimension, rhoNoisy, numOfCopies)
    numOfbases = len(measurements)

    min_hsDistance = 1e-5
    maxNumIterations = 5

    for _ in range(numOfSeeds):
        Sprobabilities = boos_probs(Probabilities,numOfCopies,numOfbases, dimension)
        rho_estimate,_ = QSE(dimension, measurements,Sprobabilities, maxNumIterations,min_hsDistance)
        fidelity = qi.state_fidelity(qi.DensityMatrix(np.array(rho_target)),
                                     qi.DensityMatrix(np.array(rho_estimate)))
        Fidelities.append(fidelity)

    return Fidelities


def boos_probs(probs,countings,numOfbases,dimension):

    probabilities = np.zeros((numOfbases,dimension))
    for i in range(numOfbases):
        pdi  = np.array( [0] + list(probs[i]) )
        pdi, _ = np.histogram(list(np.random.random_sample(countings)), np.cumsum(pdi))
        probabilities[i] = pdi/np.sum(pdi)

    return probabilities


def born_rule_probabilities(bases, rho, numberOfBases,dimension):
    probabilities=np.zeros((numberOfBases,dimension))
    for i,basis in enumerate(bases):
        probabilities[i] = np.diag(np.conj(basis).T @ rho @ basis).real
    return probabilities


def probability_distributions(bases, dimension, rho, no_countings ):

    probabilities = np.zeros((len(bases),dimension))
    for i,basis in enumerate(bases):
        pdi = np.diag(np.conj(basis).T @ rho @ basis).real
        pdi  = np.array( [0] + list(pdi) )
        pdi, _ = np.histogram(list(np.random.random_sample(no_countings)), np.cumsum(pdi))

        probabilities[i] = pdi/np.sum(pdi)

    return probabilities


def pauli_bases(num_of_qubits):

    basisZ = np.array([[1,0],[0,1]])
    basisX = np.array([[1,1],[1,-1]])/np.sqrt(2)
    basisY = np.array([[1,1],[1j,-1j]])/np.sqrt(2)
```

```
94
95      qubitBasis = [basisZ,basisX,basisY]
96
97      if num_of_qubits == 1:
98          return qubitBasis
99      else:
100         pauliIndexes = list(it.product(range(3),repeat=num_of_qubits))
101         return [ft.reduce(lambda x, y: np.kron(x,y),[qubitBasis[i] for i in
        index]) for index in pauliIndexes]
102
103
104 class PauliBases:
105     basisZ = np.matrix([[1,0],[0,1]])
106     basisX = np.matrix([[1,1],[1,-1]])/np.sqrt(2)
107     basisY = np.matrix([[1,1],[1j,-1j]])/np.sqrt(2)
108
109     qubitBases = [basisZ,basisX,basisY]
110
111     no_qubits = 2
112
113     def __init__(self,numberOfQubits):
114         self.no_qubits=numberOfQubits
115
116     def _generatorRecursion(self,num_of_qubits):
117         if num_of_qubits == 1:
118             yield from self.qubitBases
119         else:
120             yield from (np.einsum('ik,jl', prod, aBasis).reshape(2**
        num_of_qubits,2**num_of_qubits)
121                         for aBasis in self.qubitBases
122                         for prod in self._generatorRecursion(num_of_qubits-1)
        )
123
124     def generator(self):
125         return self._generatorRecursion(self.no_qubits)
126
127
128 def T(rho_0, measurement , probabilities ):
129     aBasis=measurement
130     rotatedState = np.conj( aBasis ).T @ rho_0 @ aBasis
131     stateWithImposedInformation = rotatedState - np.diag( np.diag(
        rotatedState ) ) + np.real( np.diag( probabilities ) )
132     return aBasis @ stateWithImposedInformation @ np.conj( aBasis ).T
133
134
135 def applyCompositionT(rho_0,Measurements,Probabilities):
136     idxs = [i for i in range(len(Measurements))]
137     random.shuffle(idxs)
138     rho_1 = rho_0
139     for j in idxs:
140         rho_1 = T(rho_1, Measurements[j], Probabilities[j] )
141     return rho_1
142
143
144 def projectToDensityOperator(rho,dimension):
145
146     eivals,u = np.linalg.eigh(rho)
147     positiveEivals =  eivals.real[eivals.real>0]
148     fullSpectra = np.concatenate(( np.zeros(dimension - len(positiveEivals)),
        positiveEivals ))
```

```
    rho = u@np.diag(fullSpectra)@np.conj(u).T

    return rho/np.trace(rho)


def HS_distance(rho,sigma):
    return np.linalg.norm(rho - sigma)
```

B: Codes for certification of quantum nonlocality

The method described in chapter 3 for quantum nonlocality certification was implemented in Python. Here, we present the code of that implementation. The code is available in Github [101].

```
import numpy as np
from scipy.optimize import minimize
from scipy.optimize import NonlinearConstraint
import matplotlib.pyplot as plt
import os
import itertools

from datetime import datetime
cwd = os.getcwd()


def optimal_value(filename, m, num_of_outcomes, num_of_trials, F, marginals_A,
                  marginals_B, disp = False, save = False):

    """
    This routine optimizes R

    Arguments:
    m: number of settings.
    filename: Name of the file where the experimental data is stored.
    marginals_A: a list containing the marginals (expressions like A_i x I )
                 in the Alice's side that have to be considered for the
                 calculations.
    marginals_B: a list containing the marginals (expressions like  I x B_i )
                 in the Bob's side that have to be considered for the
                 calculations.
    disp: 'True' for displaying the partial results.
    save: 'True' to save the coefficients S, Sax and Sby as numpy tensors in
          the 'npy_data' folder.

    Returns:
    Tensors "S", "Sax" and "Sby" containing the coefficients
    """

    foundResults = {}
    SolutionFound = False
    dir1 = ( cwd + '\\npy_data\\' + 's_' +
        filename + datetime.today().strftime('_%Y-%m-%d') )
    dir2 = ( cwd + '\\npy_data\\' + 'sax_' +
        filename + datetime.today().strftime('_%Y-%m-%d') )
```

```
    dir3 = ( cwd + '\\npy_data\\' + 'sby_' +
        filename + datetime.today().strftime('_%Y-%m-%d') )
    dir4 = ( cwd + '\\npy_data\\' + 'p_' +
        filename + datetime.today().strftime('_%Y-%m-%d') )
    dir5 = ( cwd + '\\npy_data\\' + 'pax_' +
        filename + datetime.today().strftime('_%Y-%m-%d') )
    dir6 = ( cwd + '\\npy_data\\' + 'pby_' +
        filename + datetime.today().strftime('_%Y-%m-%d') )
    dir7 = ( cwd + '\\npy_data\\' + 'Q_C_Dq_Gap' +
        filename + datetime.today().strftime('_%Y-%m-%d') )

    print('n: number of iterations')
    print(60*'_')
    print(f'    n       R         Q         C         \u0394Q          Gap')
    print(60*'_')
    p, c, _ = load_data(filename, F, m, num_of_outcomes,
                        marginals_A, marginals_B)
    s0, bounds = initial_guess(m, num_of_outcomes,
                               marginals_A, marginals_B) # initial guess for
    the optimizer
    n, target = 1, -1
    nlc = NonlinearConstraint( lambda x: -target
                              + R( x, p, c, m, num_of_outcomes, marginals_A,
    marginals_B),
                                    -np.inf, -target ) # constraints

    while n <= num_of_trials:
        res = minimize( R , s0 ,  args=(p, c, m, num_of_outcomes, marginals_A,
     marginals_B),
                        method='SLSQP', tol = 1e-20,
                        options={ 'ftol':1e-20, 'maxiter':5000, 'disp': False
    },
                        bounds=bounds, constraints=[nlc])
        s0 = (res.x + s0)/2  # average between the target guess 's0' and the
    current solution
                             # R = -res.fun. Thus, 'res.fun < target' => 'R >
    -target'
        if  res.fun < target:
            SolutionFound = True
            solution = res # keeps the solution that satisfies the constraints
            target = res.fun  # updates the target to the best R value found
    so far
            nlc = NonlinearConstraint( lambda x: -target
                                      + R( x, p, c, m,
                                          num_of_outcomes, marginals_A,
                                          marginals_B),
                                      -np.inf, -target ) # update the
    constraints

            Q, C, Dq, gap = results( solution.x, p, c,
                                    m, num_of_outcomes,
                                    marginals_A, marginals_B)
            s, sax, sby = vector_to_tensor(solution.x, p,
                                           m, num_of_outcomes,
                                           marginals_A, marginals_B)
            # save solutions at the 'npy_data' folder
            pabxy, pax, pby = p
            if save:
                np.save(dir1, s)
                np.save(dir2, sax)
```

```
                np.save(dir3, sby)
                np.save(dir4, pabxy)
                np.save(dir5, pax)
                np.save(dir6, pby)
                np.save(dir7, np.array([Q, C, Dq, gap]) )

            # display results
            if disp:
                print( "%5d    %4.5f   %4.5f   %4.5f    %4.5f    %4.5f"
                      %(n, -solution.fun, Q, C, Dq, gap) )

        n += 1

    if SolutionFound:
        foundResults['Coefficients'] = (s, sax, sby)
        foundResults['values'] = (Q, C, Dq, gap)
        return foundResults
    else:
        print("\nNo solution has been found ...")



def probs_local_hidden_model(num_of_outcomes , m):

    outcomes = [1] + [0 for i in range(num_of_outcomes-1)]
    rows = ( list(i) for i in set( itertools.permutations(outcomes)) )

    for i in itertools.product(rows, repeat = m):
        yield np.array(i).T


def lhv_value(s, m, num_of_outcomes ):
    """
    Inputs:
    s: tuple containing the tensors (S, Sax, Sby)
    num_of_outcomes: number of outcomes
    m: number of settings

    Returns:
    local bound for a 2-party scenario, with "m" settings
    and "num_of_outcomes" possible outcomes
    """
    s, sax, sby = s
    Cmax = -1e10
    for pax in probs_local_hidden_model(num_of_outcomes , m):
        for pby in probs_local_hidden_model(num_of_outcomes , m):
            C =  ( np.sum(sax*pax) + np.sum(sby*pby)
                  + np.sum( s*np.einsum('ax,by->abxy', pax,pby) ) )
            if C > Cmax:
                Cmax = C
    return Cmax


def quantum_max_violation(s, p):
    """
    Inputs:
    s: tuple containing the tensors (S, Sax, Sby)
    p: tuple containing the tensors (p, pax, pby)

    Returns:
```

```
    Quantum Value
    """
    s, sax, sby = s
    p, pax, pby = p
    return np.sum(sax*pax) + np.sum(sby*pby) + np.sum(s*p)


def error_quantum_value( s, c, m, num_of_outcomes):

    c,cax,cby = c
    s,sax,sby = s
    Delta_Q, Delta_Q1, Delta_Q2 =  0, 0, 0
    for i in itertools.product(range(m), repeat = 2):
        for l in itertools.product(range(num_of_outcomes), repeat = 2):
            x,y = i
            a,b = l
            dQ = (s[a,b,x,y]*np.sum(c[:,:,x,y])
                  - np.sum(s[:,:,x,y]*c[:,:,x,y]))/(np.sum(c[:,:,x,y])**2)
            if False:
                dQ1 = (sax[a,x]*np.sum(cax[:,x])
                       - np.sum(sax[:,x]*cax[:,x]))/(np.sum(cax[:,x])**2)
                dQ2 = (sby[b,y]*np.sum( cby[:,y])
                       -np.sum(sby[:,y]*cby[:,y]))/(np.sum(cby[:,y])**2)
                Delta_Q1 += dQ1**2*cax[a,x]
                Delta_Q2 += dQ2**2*cby[b,y]
            else:
                dQ = dQ + (1/m)*(sax[a,x]*np.sum(c[:,:,y,x])
                         - np.sum(sax[:,x]*c[:,:,x,y]))/(np.sum(c[:,:,x,y])
    **2)
                dQ = dQ + (1/m)*(sby[b,y]*np.sum(c[:,:,y,x])
                         - np.sum(sby[:,y]*c[:,:,x,y]))/(np.sum(c[:,:,x, y])
    **2)
            Delta_Q += dQ**2*c[a,b,x,y]

    return np.sqrt( Delta_Q  + Delta_Q1 + Delta_Q2 )



def initial_guess(m, num_of_outcomes, marginals_A, marginals_B):

    nop = (num_of_outcomes**2)*(m**2) + num_of_outcomes**2*(len(marginals_A) +
     len(marginals_B))
    lb, ub = -1, 1
    bounds =  tuple( [(lb, ub) for i in range(nop)] )
    s0 = np.array(  [(ub - lb) * np.random.random() + lb   for i in range(nop)
    ] )

    return s0, bounds


def R( s, p, c, m, num_of_outcomes, marginals_A, marginals_B):

    k = (num_of_outcomes)**2*(m)**2
    s = vector_to_tensor(s, p, m, num_of_outcomes, marginals_A, marginals_B)
    Q = quantum_max_violation(s, p)
    C = lhv_value(s, m, num_of_outcomes )
    Dq = error_quantum_value( s, c, m, num_of_outcomes)

    return  -(Q - Dq + k)/(C + k)

```

```
209
210 def vector_to_tensor(s, p, m, num_of_outcomes, marginals_A, marginals_B):
211
212     o = num_of_outcomes
213     n = m**2*o**2
214     sax, sby = np.zeros((2,m)), np.zeros((2,m))
215     s, s_marginals = np.array( s[:n] ).reshape((o,o,m,m)),  s[n:]
216
217     j = 0
218     if len(marginals_A) > 0:
219         for x in marginals_A:
220             for a in range(o):
221                 sax[a,x] = s_marginals[j]
222                 j += 1
223     if len(marginals_B) > 0:
224         for y in marginals_B:
225             for b in range(o):
226                 sby[b,y] = s_marginals[j]
227                 j += 1
228     s[np.nonzero(p[0] == 0)] = np.zeros((s[np.nonzero(p[0] == 0)].shape[0]))
229
230     return (s, sax, sby)
231
232
233
234 def results( s, p, c, m, num_of_outcomes, marginals_A, marginals_B):
235
236     o = num_of_outcomes
237     n = m**2*o**2
238     sax, sby = np.zeros((2,m)), np.zeros((2,m))
239     s, s_marginals = np.array( s[:n] ).reshape((o,o,m,m)),s[n:]
240
241     j = 0
242     if len(marginals_A) > 0:
243         for x in marginals_A:
244             for a in range(o):
245                 sax[a,x] = s_marginals[j]
246                 j += 1
247     if len(marginals_B) > 0:
248         for y in marginals_B:
249             for b in range(o):
250                 sby[b,y] = s_marginals[j]
251                 j += 1
252
253     s[np.nonzero(p[0] == 0)] = np.zeros((s[np.nonzero(p[0] == 0)].shape[0]))
254     s = (s, sax, sby)
255     Q = quantum_max_violation(s, p)
256     C = lhv_value(s, m, o )
257     Dq = error_quantum_value( s, c, m, o)
258
259     return  Q, C, Dq, (Q - Dq - C)
260
261
262
263 def load_data(name_of_file, F, m, num_of_outcomes, marginals_A, marginals_B):
264
265
266     filename = cwd + '\\experimental_data\\' + name_of_file + '.txt'
267
268     with open(filename,'r') as f:
```

```
269         data = f.read()
270
271     data = data.split('\n')
272     labels_counts, counts = [] , []
273     for i in range(len(data)):
274         x,y = tuple( data[i].split())
275         labels_counts.append(x)
276         counts.append(float(y))
277
278     countings = dict(zip(labels_counts, counts))
279
280
281     shape = (num_of_outcomes,)*2 + (m,)*2
282     p = np.zeros( shape )
283     c = np.zeros( shape )
284     pax, cax =  np.zeros( (2,2) ), np.zeros( ( 2,2) )
285     pby, cby =  np.zeros( (2,2) ), np.zeros( ( 2,2) )
286
287     measured_marginals = False
288     for l in labels_counts:
289         a,b,x,y = l
290         if a == '-':
291             measured_marginals = True
292             cax[int(a),int(x)] = countings[l]
293         elif b == '-':
294             measured_marginals = True
295             cby[int(b),int(y)] = countings[l]
296         else:
297             a,b,x,y = list( map( int,l ) )
298             c[a,b,x,y] = countings[l]
299
300     for l in labels_counts:
301         a,b,x,y = l
302         a,b,x,y = list( map( int,l ) )
303         p[a,b,x,y] = c[a,b,x,y]/(np.sum(c[:,:,x,y]))
304
305     p1 = p
306     num_of_trials = 10
307     output = optimal_settings_KL_divergence(F, p, num_of_outcomes, m,
        num_of_trials)
308     p = get_probs_from_settings(output.x,m)
309
310     if measured_marginals:
311         for l in labels_counts:
312             a,b,x,y = l
313             a,b,x,y = list(map(int,l))
314             pax[int(a), int(x)] = cax[int(a),int(x)]/(np.sum(cax[:,int(x)]))
315             pby[int(b), int(y)] = cby[int(b), int(y)]/(np.sum(cby[:,int(y)]))
316
317     if len(marginals_A) > 0:
318         for x in marginals_A:
319             for a in range(2):
320                 pax[a,x] = np.sum(p[a,: ,x ,0])
321     if len(marginals_B) > 0:
322         for y in marginals_B:
323             for b in range(2):
324                 pby[b,y] = np.sum(p[:,b ,0 ,y])
325
326     return (p, pax, pby) , ( c, cax, cby ), measured_marginals
327
```

```
328
329
330
331 # Kullback-Leible divergence
332
333 def U(x):
334
335     beta, gamma, alpha= x[0], x[1], x[2]
336     return np.array( [ [ np.exp( -1j*gamma )*np.cos( beta/2 ) , - np.exp( 1j*
    alpha )*np.sin( beta/2 ) ],
337                        [ np.exp( -1j*alpha )*np.sin( beta/2 ), np.exp( 1j*gamma )
    *np.cos(beta/2) ] ] )
338
339
340 def W(theta):
341
342     ket0 = np.array([1,0])
343     ket1 = np.array([0,1])
344     psi = np.cos(theta)*np.kron(ket0,ket0) + np.sin(theta)*np.kron(ket1,ket1)
345
346     return np.outer( psi, np.conjugate( psi ) )
347
348
349 def get_probs_from_settings(x, m):
350
351     o = 2
352     theta, x = x[0],  x[1:]
353     rho = W(theta)
354     Us = np.array( list( map(U, x.reshape( int(x.shape[0]/3) ,3) ) ) )
355     Ua, Ub = Us[:int(Us.shape[0]/2)], Us[int(Us.shape[0]/2):]
356
357     pa, pb, p = np.zeros((o,m)), np.zeros((o,m)), np.zeros((o,o,m,m))
358     for i in itertools.product(range(m), repeat = 2):
359         for l in itertools.product(range(o), repeat = 2):
360             x,y = i
361             a,b = l
362             Pa = np.outer( Ua[x][:,a], np.conjugate( Ua[x][:,a] ) )
363             Pb = np.outer( Ub[y][:,b], np.conjugate( Ub[y][:,b] ) )
364             p[a,b,x,y] = np.real( np.trace( rho @ np.kron(Pa, Pb) ) )
365
366     return  p
367
368
369 def initial_data(m, num_of_parties):
370
371     lb , ub = 0, np.pi
372     x0 = np.array( [(ub - lb) * np.random.random_sample()
373                     + lb for i in range(3*num_of_parties*m)] )
374
375     lb , ub = 0, np.pi
376     theta = (ub - lb) * np.random.random_sample() + lb
377     x0 = np.insert(x0 , 0, theta )
378
379     ### Bounds
380     lb , ub = 0, np.pi
381     x_bounds = np.array( [(0, np.pi/4)]
382                         + [(lb, ub) for i in range(3*num_of_parties*m)] )
383
384     return x0, x_bounds
385
```

```
386
387 def kullback_leibe_divergence(x, F, f, num_of_outcomes, m):
388
389     """
390     f: relative frequencies (original probabilities)
391     """
392     p = get_probs_from_settings(x, m)
393
394     D = 0
395     for i in itertools.product(range(num_of_outcomes), repeat = m):
396         for j in itertools.product(range(m), repeat = m):
397             a,b = i
398             x,y = j
399             D += F(x,y)*f[a,b,x,y]*(np.log2(f[a,b,x,y])-np.log2(p[a,b,x,y]))
400
401     return D
402
403
404
405 def optimal_settings_KL_divergence(F, probs, num_of_outcomes,
406                                                 m, num_of_trials):
407
408     """
409     This routine minimize the Kullback-Leibe divergence.
410     It optimizes over the parameters in the settings.
411     """
412
413     # one can also use COBYLA in this optimization
414     # just comment the next line and uncomment the
415     # line corresponding to COBYLA.
416
417     OPTIMIZER = 'SLSQP'
418 #     OPTIMIZER = 'COBYLA'
419
420     # initial guess and bounds for the optimizer
421     x0, bounds = initial_data(num_of_outcomes, m)
422     threshold = 0
423     cons = ({'type': 'ineq', 'fun': lambda x: threshold
424              - kullback_leibe_divergence(x, F, probs, num_of_outcomes, m)} )
425
426     res = minimize( kullback_leibe_divergence , x0 ,
427                     args=(F,probs, num_of_outcomes, m),
428                     method=OPTIMIZER, tol = 1e-20,
429                     options={ 'maxiter':250, 'disp': False},
430                    constraints=cons
431                   )
432
433     previous = res.fun
434     result = res
435     for i in range(num_of_trials):
436
437         x0 = (res.x + x0)/2
438         res = minimize( kullback_leibe_divergence , x0 ,
439                     args=(F, probs, num_of_outcomes, m),
440                     method=OPTIMIZER, tol = 1e-20,
441                     options={ 'maxiter':250, 'disp': False},
442                     constraints=cons
443                   )
444
445         # save the best result
```

```
        if res.fun < previous:
            previous = res.fun
            result = res

    return result
```

C: Codes for the quantum marginal problem algorithm

The following codes correspond to the algorithms presented in chapter 4. The code was written in Python and is available in Github [123].

```
import qiskit.quantum_info as qi
import itertools
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import MaxNLocator
import time
import h5py
import scipy
from scipy import optimize
from datetime import datetime
from datetime import timedelta
import os
cwd = os.getcwd()



def spectra(self):
    return np.linalg.eigvalsh(self.data)

qi.DensityMatrix.spectra = spectra



class DensityMatrix(qi.DensityMatrix):
    def __init__(self, data, dims=None):
        self._labels = None
        super().__init__(data, dims)

    def get_labels(self):
        return self._labels

    def set_labels(self, given_labels):
        self._labels = given_labels

    def spectra(self):
        return np.linalg.eigvalsh(self.data)



def create_storing_folers():

```

```
    path_file_plots = cwd + '\\plots\\'
    path_file_data = cwd + '\\data\\'
    path_file_txtdata = cwd + '\\txtdata\\'
    directories = [path_file_plots, path_file_data,
                   path_file_txtdata]

    created = False
    for pathname in directories:
        try:
            os.mkdir(pathname)
            created = True
        except:
            pass
    if created:
        print("Folders to store data has just been created ...")


def partial_trace(rho, antisystem):

    n = len( rho.dims() )
    lista = [n-i-1 for i in antisystem]
    lista.sort()
    labels = set(range(n)) - set(antisystem)
    sigma = DensityMatrix( qi.partial_trace(rho, tuple(lista) ),
                           dims= rho.dims()[:len(labels)] )
    sigma.set_labels(list(labels))
    return sigma


def qmp(d, num_of_qudits, prescribed_marginals, rank = 1, prescribed_spectra =
     [], params = {}):

    start = time.time()

    if len(prescribed_spectra) > 0:
        print("prescribed spectra mode..")
    else:
        print("prescribed rank mode..")

    print(33*'_')
    print(f'      n      tdist      gdist')
    print(33*'_')


    rho_gen = params['rho_gen']
    txt_file_name = (cwd + '\\txtdata\\' + '_N' +
                     str(num_of_qudits) + 'd' + str(d) + 'data.txt')

    if params['save_iter_data']:
        try:
            open(txt_file_name,'w').close()
            f = open(txt_file_name,'a')
        except:
            f = open(txt_file_name,'w')

    dn, swapper_d = d**num_of_qudits, swapper(d)
    dims = tuple([d for i in range(num_of_qudits)])

    if params['x_0'] != None:
        x_0 = params['x_0']
```

```
    else:
        if params['seed'] == 'mixed':
            x_0 = qi.DensityMatrix( qi.random_density_matrix(dims))
        if params['seed'] == 'mms':
            x_0 = DensityMatrix(np.identity(dn)/dn, dims)
        if params['seed'] == 'pure':
            x_0 = qi.DensityMatrix( qi.random_statevector(dims))

    mdistance, edistance, gdistance, dtotal = [], [], [], []
    number_of_iterations, total_distance, previous_total_distance = 0, 10, 10
    marginal_hsd, eigenvals_distance = 10, 10

    runtime = 0

    while total_distance > params['dtol'] and number_of_iterations < params['
    max_iter']:

        x_0 = impose_marginals( d, num_of_qudits, x_0,
                                prescribed_marginals, swapper_d )

        if len(prescribed_spectra) > 0:
            eigenvals_distance, x_0 = impose_prescribed_eigenvals(x_0,
    prescribed_spectra)
        else:
            eigenvals_distance, x_0 = impose_rank_constraint(x_0, rank)

        x_0 = qi.DensityMatrix(x_0, dims)

        # metrics
        resulting_marginals, marginal_hsd = compute_marginals_distance( x_0,
    prescribed_marginals,

    num_of_qudits)
        total_distance = np.sqrt(marginal_hsd**2 + eigenvals_distance**2)
        dg = np.linalg.norm( x_0.data - rho_gen.data)
        gdistance.append( np.real( dg  ) )
        edistance.append( eigenvals_distance )
        mdistance.append( np.real( marginal_hsd ) )
        dtotal.append( total_distance )

        if number_of_iterations % params['iter_to_print'] == 0:
            print( "%7d  %4.5E  %4.5E" %( number_of_iterations + 1,
                                          total_distance, dg ) )
        # stores in h5 data
        if params['save_h5']:
            save_partial_data(x_0, mdistance, edistance,
                              gdistance, stype = params['h5_name'])
        # stores txt data
        end = time.time()
        runtime += end - start
        if params['save_iter_data']:
            f.write(f'%i  %1.10f  %s\n'%(number_of_iterations + 1,
    total_distance,
                                          time_format(timedelta(seconds =
    runtime)) ))
            f.flush()

        number_of_iterations += 1
        start = time.time()

```

```
155
156     if params['save_iter_data']:
157         f.close()
158
159     runtime += time.time() - start
160     data = {'rho_n':x_0, 'mdistance': mdistance,
161             'edistance':edistance, 'gdistance':gdistance,
162             'tdistance':dtotal,
163             'runtime':time_format(timedelta(seconds = runtime)) }
164
165     return data
166
167
168
169 def accelerated_qmp(d, num_of_qudits, prescribed_marginals, rank = 1,
170                     prescribed_spectra = [], params = {}):
171
172     start = time.time()
173
174     if len(prescribed_spectra) > 0:
175         print("prescribed spectra mode..")
176     else:
177         print("prescribed rank mode..")
178
179     print(33*'_')
180     print(f'      n      tdist      gdist')
181     print(33*'_')
182
183     rho_gen = params['rho_gen']
184     txt_file_name = (cwd + '\\txtdata\\' + '_N'
185                      + str(num_of_qudits) + 'd'
186                      + str(d) + 'data.txt')
187
188     if params['save_iter_data']:
189         try:
190             open(txt_file_name,'w').close()
191             f = open(txt_file_name,'a')
192         except:
193             f = open(txt_file_name,'w')
194
195     dn, swapper_d = d**num_of_qudits, swapper(d)
196     dims = tuple([d for i in range(num_of_qudits)])
197
198     if params['x_0'] != None:
199         x_0 = params['x_0']
200     else:
201         if params['seed'] == 'mixed':
202             x_0 = qi.DensityMatrix( qi.random_density_matrix(dims))
203         if params['seed'] == 'mms':
204             x_0 = DensityMatrix(np.identity(dn)/dn, dims)
205         if params['seed'] == 'pure':
206             x_0 = qi.DensityMatrix( qi.random_statevector(dims))
207
208     mdistance, edistance, gdistance, dtotal = [], [], [], []
209     number_of_iterations, total_distance, previous_total_distance = 0, 10, 10
210     marginal_hsd, eigenvals_distance, threshold = 10, 10, 1e-1
211
212     alfa, alfa_n = params['alfa'], params['alfa_n']
213     mu, bt = params['mu'], params['bt']
214     beta = alfa_n**2 + bt
```

```
215
216     runtime = 0
217     if not params['accelerated']:
218         alfa, beta, mu = 1, 0, 0
219
220     while total_distance > params['dtol'] and number_of_iterations < params['
        max_iter']:
221
222         x_0 = accelerated_impose_marginals( d, num_of_qudits, x_0,
223                                             prescribed_marginals, swapper_d,
224                                             alfa_n, alfa, mu, beta)
225         if len(prescribed_spectra) > 0:
226             eigenvals_distance, x_0 = impose_prescribed_eigenvals(x_0,
227                                                     prescribed_spectra)
228         else:
229             eigenvals_distance, x_0 = impose_rank_constraint(x_0, rank)
230
231         x_0 = qi.DensityMatrix(x_0, dims)
232         resulting_marginals, marginal_hsd = compute_marginals_distance( x_0,
233                                                 prescribed_marginals,
        num_of_qudits)
234         total_distance = np.sqrt(marginal_hsd**2 + eigenvals_distance**2)
235         dg = np.linalg.norm( x_0.data - rho_gen.data)
236         gdistance.append( np.real( dg  ) )
237         edistance.append( eigenvals_distance )
238         mdistance.append( np.real( marginal_hsd ) )
239         dtotal.append( total_distance )
240
241         if params['accelerated']:
242             alfa_n = 1/(1e-5*number_of_iterations + 1)**alfa
243             beta = alfa_n**2 + bt
244         if number_of_iterations % params['iter_to_print'] == 0:
245             print( "%7d  %4.5E  %4.5E" %( number_of_iterations + 1,
246                                           total_distance, dg ) )
247         # stores in h5 data
248         if params['save_h5']:
249             save_partial_data(x_0, mdistance, edistance,
250                               gdistance, stype = params['h5_name'])
251         # stores txt data
252         end = time.time()
253         runtime += end - start
254         if params['save_iter_data']:
255             f.write(f'%i  %1.10f  %s\n'%(number_of_iterations + 1,
        total_distance,
256                                           time_format(timedelta(seconds =
        runtime)) ))
257             f.flush()
258         number_of_iterations += 1
259         start = time.time()
260
261     if params['save_iter_data']:
262         f.close()
263
264     runtime += time.time() - start
265     data = {'rho_n':x_0, 'mdistance': mdistance,
266             'edistance':edistance, 'gdistance':gdistance,
267             'tdistance':dtotal,
268             'runtime':time_format(timedelta(seconds = runtime)) }
269
270     return data
```

```
271
272
273 def swapper(d):
274
275     p = 0
276     Id = np.identity(d)
277     for i in range(d):
278         for j in range(d):
279             v = np.outer(Id[:,i],Id[:,j])
280             u = np.transpose(v)
281             p += np.kron(v,u)
282     return p
283
284
285 def kron(*matrices):
286
287     m1, m2, *ms = matrices
288     m3 = np.kron(m1, m2)
289
290     for m in ms:
291         m3 = np.kron(m3, m)
292
293     return m3
294
295
296 def Pj( in_label, marginal, dl, num_of_qudits, swapper_d ):
297     """
298     dl: local dimension
299     marginal: reduced system with labels given in the tuple "in_label"
300
301     """
302
303     label = in_label
304     n = num_of_qudits - len( marginal.dims() )
305     dims = tuple( [ dl for i in range( num_of_qudits ) ] )
306     swapped_matrix = kron( marginal.data, np.identity( dl**n ) )
307
308     all_labels = [ i for i in range( num_of_qudits ) ]
309     right_labels = [ i for i in range( list( label )[-1] + 1, num_of_qudits )
    ]
310     left_labels = [ i for i in range( list( label )[0] ) ]
311
312     if left_labels + list( label ) + right_labels == all_labels:
313         nl  = list(label)[0]
314         nr = num_of_qudits - nl  - len(label)
315         Il, Ir = np.identity( dl**nl ), np.identity( dl**nr )
316         swapped_matrix = qi.DensityMatrix( kron( Il, marginal.data, Ir ) ,
    dims )
317         swapped_matrix = swapped_matrix/swapped_matrix.trace()
318         return swapped_matrix
319     else:
320         nl = list(label)[0]
321         nr = num_of_qudits - nl  - len(label)
322         Il, Ir = np.identity( dl**nl ), np.identity( dl**nr )
323         swapped_matrix = kron( Il, marginal.data, Ir )
324         label = tuple( left_labels + list( label ) )
325
326     length = len( label )
327     remaining = tuple( [ i for i in range( length ) ] )
328
```

```
    while length > 0 and label != remaining:

        last = label[-1]
        numOfswapps = np.abs( last  - length )
        l1, l2 = length - 1, num_of_qudits - ( length + 1 )
        I1, I2 = np.identity( dl**l1 ), np.identity( dl**l2 )
        gate = kron( I1, swapper_d, I2 )
        swapped_matrix = gate @ swapped_matrix @ gate

        for i in range( numOfswapps ):
            l1, l2 = l1 + 1, l2 - 1
            I1, I2 = np.identity( dl**l1 ), np.identity( dl**l2 )
            gate = kron( I1, swapper_d, I2 )
            swapped_matrix = gate @ swapped_matrix @ gate

        label = tuple( list( label[:-1] ) )
        length = len( label )
        remaining = tuple( [ i for i in range( length ) ] )

    swapped_matrix = qi.DensityMatrix( swapped_matrix, dims )
    swapped_matrix = swapped_matrix/swapped_matrix.trace()

    return swapped_matrix


def impose_prescribed_eigenvals(x_0, prescribed_spectra):

    dn, _ = x_0.data.shape
    rank = len(prescribed_spectra)
    prescribed_spectra = sorted(prescribed_spectra)
    prescribed_spectra = np.concatenate(( np.zeros((dn-rank)),
                                          prescribed_spectra ))
    eigenvalues, eigenvects = np.linalg.eigh( x_0.data )
    eigenvals_distance = np.linalg.norm( eigenvalues - prescribed_spectra )
    rhof = eigenvects @ np.diag( prescribed_spectra ) @ np.conjugate(
    eigenvects.T )

    return eigenvals_distance, rhof/np.trace(rhof)


def impose_rank_constraint(x_0, rank):

    dn, _ = x_0.data.shape
    eigenvalues, eigenvects = np.linalg.eigh( x_0.data )

    a, b = np.zeros((dn-rank)), eigenvalues[ dn - rank: ]
    if len(eigenvalues[ eigenvalues < 0]) <= rank:
        eigenvals_distance = np.linalg.norm( eigenvalues[ eigenvalues < 0] )
        eigenvalues[ eigenvalues < 0] =  - eigenvalues[ eigenvalues < 0]
    else:
        eigenvals_distance = np.linalg.norm( eigenvalues[:-rank] )

    rhof = eigenvects @ np.diag( np.concatenate((a,b)) ) @ np.conjugate(
    eigenvects.T )

    return eigenvals_distance, rhof/np.trace(rhof)


def impose_marginals(d, num_of_qudits, x_0, prescribed_marginals, swapper_d):

```

```
    dn = d**num_of_qudits
    all_systems = set( list( range(num_of_qudits)) )

    for l in list( prescribed_marginals.keys() ):
        antisys = tuple( all_systems - set( l ) )
        tr_rho0_I = partial_trace( x_0 , list( antisys ) )
        x_0 = (x_0  + Pj( l, prescribed_marginals[l], d, num_of_qudits,
    swapper_d ) -
                Pj( l, tr_rho0_I, d, num_of_qudits, swapper_d )  )

    return x_0


def accelerated_impose_marginals(d, num_of_qudits, x_0, prescribed_marginals,
                                 swapper_d, alfa_n, alfa, mu, beta):

    all_systems = set( list( range(num_of_qudits)) )
    dims = tuple( [ d for i in range( num_of_qudits ) ] )
    dn = d**num_of_qudits

    for l in list( prescribed_marginals.keys() ):
        antisys = tuple( all_systems - set( l ) )
        tr_rho0_I = partial_trace( x_0, list(antisys) )
        d_0 = ( Pj( l, prescribed_marginals[l], d, num_of_qudits, swapper_d ).
    data
                - Pj( l, tr_rho0_I, d, num_of_qudits, swapper_d ).data )/alfa
        d_1 = d_0 + beta*d_0
        y_0 = x_0.data + alfa*d_1
        x_0 = mu*alfa_n*x_0 + (1 - mu*alfa_n)*y_0
        x_0 = qi.DensityMatrix( x_0, dims )

    return x_0/x_0.trace()


def compute_marginals_distance(rho0, prescribed_marginals,  num_of_qudits):

    all_systems = set( list( range(num_of_qudits)) )
    marginal_hsd = 0
    projected_marginals = {}

    for l in list( prescribed_marginals.keys() ):
        antisys = tuple( all_systems - set(l) )
        projected_marginals[l] = partial_trace( rho0, list(antisys) )
        marginal_hsd += np.linalg.norm( projected_marginals[l].data
                                        - prescribed_marginals[l].data)**2
    norm = len( list( prescribed_marginals.keys() ) )
    marginal_hsd = np.sqrt(marginal_hsd/norm)

    return projected_marginals, marginal_hsd



def simul_data(d, num_of_qudits, labels_marginals, dtype = "pure" ):

    dn = d**num_of_qudits
    dims = tuple([d for i in range(num_of_qudits)])

    if dtype == "AME":
        rho_gen = np.identity(dn)/dn
        rho_gen = qi.DensityMatrix(rho_gen, dims)
```

```
    elif dtype == "mixed":
        rho_gen = qi.random_density_matrix(dims).data
        rho_gen = qi.DensityMatrix(rho_gen, dims)
    elif dtype == "pure":
        rho_gen = qi.DensityMatrix(qi.random_statevector(dn), dims)

    marginals = {}
    all_systems = set( list( range(num_of_qudits)) )

    for s in labels_marginals:
        tracedSystems = tuple( all_systems - set( s ) )
        if len(tracedSystems) > 0:
            marginals[s] = partial_trace(rho_gen, list(tracedSystems))
        else:
            marginals[s] = partial_trace(rho_gen, list(s))

    return marginals, rho_gen



def time_format(runtime):

    t = str(runtime)
    h,mi,s = t.split(':')
    s = str(np.round(np.float(s),2))

    f,dec = s.split('.')
    if int(f) < 10:
        s = '0' + s

    if int(dec) < 10:
        dec = dec + '0'

    return h + ':' + mi + ':' + s



def plot_data(d, num_of_qudits, edistance, mdistance,
              gdistance, runtime, params = {} ):

    cwd = os.getcwd()
    a, h = 7.024, 4.82

    plt.figure(figsize=(a,h), dpi = 100 )
    plt.figure().gca().xaxis.set_major_locator(MaxNLocator(integer=True))

    plt.plot( list( range( len(edistance) ) ), edistance,
             label = r'$\mathcal{D}_{\lambda}$',
             linestyle='-', linewidth = 1.3,
             markersize = 3.6, mfc='none',
             color='royalblue')
    plt.plot(  list( range( len(mdistance) ) ), mdistance,
             label = r'$\mathcal{D}_{M}$',
             linestyle='--', linewidth = 1.3,
             markersize = 3.6, mfc='none',
             color='tab:red')
    if params['plot_global']:
        plt.plot(  list( range( len(gdistance) ) ), gdistance,
                 label = r'$\mathcal{D}_{G}$',
                 linestyle='dashdot', linewidth = 1.3,
```

```
                markersize = 3.6, mfc='none',
                color='forestgreen')

    plt.title(f'N = {num_of_qudits} ,  d = {d}  ( runtime = {time_format(
    runtime)})')
    plt.yscale('log')
    plt.xlabel("n", fontsize = 10)

    plt.legend( loc= 'lower left', fontsize = 12)

    plt.xlim(0,  list( range( len(edistance) ) )[-1] )
    plt.tick_params(direction='out', axis='y',
                    which='minor', colors='black')

    if params['save_plot']:
        name = ( params['name'] + '_N' + str(num_of_qudits)
                + 'd' + str(d)  + datetime.today().strftime('_%Y-%m-%d')
                + '.png')
        path_file = cwd + '\\plots\\' + name
        plt.savefig(path_file, format='png',bbox_inches='tight')

    plt.show()



def save_partial_data(rho0, mdistance, edistance, gdistance, stype = 'pure'):

    Data = {'mhsd': np.array( mdistance), 'edist': np.array(edistance),
            'gdist': np.array(gdistance), 'rho_n': rho0.data}

    N = len( rho0.dims() )
    local_dim, *_ = rho0.dims()

    file_name = ( cwd + '\\data\\' + stype +
                 "_rhoN"+ str(N) + "d"
                 + str(local_dim) + ".h5" )
    try:
        with h5py.File(file_name,'w') as f:
            for l in list( Data.keys() ):
                dtset = f.create_dataset( l, data = Data[l],
                                         compression="gzip",
                                         compression_opts=9 )
                f.flush()
    except:
        with h5py.File(file_name,'x') as f:
            for l in list( Data.keys() ):
                dtset = f.create_dataset( l, data = Data[l],
                                         compression="gzip",
                                         compression_opts=9 )
                f.flush()
    else:
        with h5py.File(file_name,'a') as f:
            for l in list( Data.keys() ):
                try:
                    del f[l]
                    dtset = f.create_dataset( l, data = Data[l],
                                             compression="gzip",
                                             compression_opts=9 )
                except:
                    dtset = f.create_dataset( l, data = Data[l],
```

```
564                                                 compression="gzip",
565                                                 compression_opts=9 )
566                     f.flush()
```

Bibliography

[1] W. Pauli, *In quantentheorie, handbuch der physik*, Vol. 24 (Springer, Berlin, 1933), Pt 1, p. 28.

[2] A. Einstein, B. Podolsky, and N. Rosen, "Can quantum-mechanical description of physical reality be considered complete?", Phys. Rev. **47**, 777–780 (1935).

[3] D. W. Belousek, "Einstein's 1927 unpublished hidden-variable theory: its background, context and significance", Studies in History and Philosophy of Science Part B: Studies in History and Philosophy of Modern Physics **27**, 437–461 (1996).

[4] J. S. Bell, "On the Einstein Podolsky Rosen paradox", Physics Physique Fizika **1**, 195–200 (1964).

[5] A. Acín, N. Brunner, N. Gisin, S. Massar, S. Pironio, and V. Scarani, "Device-independent security of quantum cryptography against collective attacks", Phys. Rev. Lett. **98**, 230501 (2007).

[6] S. Pironio, A. Acín, S. Massar, A. Giroday, D. Matsukevich, P. Maunz, S. Olmschenk, D. Hayes, L. Luo, T. Manning, and C. Monroe, "Random numbers certified by bell's theorem", Nature **464**, 1021–4 (2010).

[7] A. J. Coleman, "Structure of fermion density matrices", Rev. Mod. Phys. **35**, 668–686 (1963).

[8] X.-D. Yu, T. Simnacher, N. Wyderka, H. C. Nguyen, and O. Gühne, "A complete hierarchy for the pure state marginal problem in quantum mechanics", Nature Communications **12**, 1012 (2021).

[9] F. Huber and O. Gühne, "Characterizing ground and thermal states of few-body hamiltonians", Phys. Rev. Lett. **117**, 010403 (2016).

[10] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information: 10th anniversary edition*, 10th (Cambridge University Press, USA, 2011).

[11] F. T. Hioe and J. H. Eberly, "N-level coherence vector and higher conservation laws in quantum optics and quantum mechanics", Phys. Rev. Lett. **47**, 838–841 (1981).

[12] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki, "Quantum entanglement", Rev. Mod. Phys. **81**, 865–942 (2009).

[13] R. F. Werner, "Quantum states with einstein-podolsky-rosen correlations admitting a hidden-variable model", Phys. Rev. A **40**, 4277–4281 (1989).

[14] A. Peres, "Separability criterion for density matrices", Phys. Rev. Lett. **77**, 1413–1415 (1996).

[15] M. Horodecki, P. Horodecki, and R. Horodecki, "Separability of mixed states: necessary and sufficient conditions", Physics Letters A **223**, 1–8 (1996).

[16] B. M. Terhal, "Detecting quantum entanglement", Theoretical Computer Science **287**, Natural Computing, 313–335 (2002).

[17] M. Lewenstein, B. Kraus, J. I. Cirac, and P. Horodecki, "Optimization of entanglement witnesses", Phys. Rev. A **62**, 052310 (2000).

[18] Born, Max, "Zur Quantenmechanik der Stoßvorgänge", Zeitschrift für Physik **37**, 863–867 (1926).

[19] J. A. Bergou, "Quantum state discrimination and selected applications", Journal of Physics: Conference Series **84**, 012001 (2007).

[20] J. M. Renes, R. Blume-Kohout, A. J. Scott, and C. M. Caves, "Symmetric informationally complete quantum measurements", Journal of Mathematical Physics **45**, 2171–2180 (2004).

[21] J. M. Renes, "Spherical-code key-distribution protocols for qubits", Phys. Rev. A **70**, 052314 (2004).

[22] A. Acín, S. Pironio, T. Vértesi, and P. Wittek, "Optimal randomness certification from one entangled bit", Phys. Rev. A **93**, 040102 (2016).

[23] I. D. Ivonovic, "Geometrical description of quantal state determination", Journal of Physics A: Mathematical and General **14**, 3241–3245 (1981).

[24] W. K. Wootters and B. D. Fields, "Optimal state-determination by mutually unbiased measurements", Annals of Physics **191**, 363–381 (1989).

[25] C. Spengler, M. Huber, S. Brierley, T. Adaktylos, and B. C. Hiesmayr, "Entanglement detection via mutually unbiased bases", Phys. Rev. A **86**, 022311 (2012).

[26] N. J. Cerf, M. Bourennane, A. Karlsson, and N. Gisin, "Security of quantum key distribution using d-level systems", Phys. Rev. Lett. **88**, 127902 (2002).

[27] S. Bandyopadhyay, P. Boykin, V. Roychowdhury, and F. Vatan, "A new proof for the existence of mutually unbiased bases", Algorithmica **34**, 512–528 (2002).

[28] W. K. Wootters and B. D. Fields, Annals of Physics **191**, 363–381 (1989).

[29] M. Grassl, *Computing numerical and exact sic-povms*, (2021) https://chaos.if.uj.edu.pl/ZOA/files/semianria/chaos/29.03.2021.pdf (visited on 10/22/2021).

[30] A. J. Scott and M. Grassl, "Symmetric informationally complete positive-operator-valued measures: a new computer study", Journal of Mathematical Physics **51**, 042203 (2010).

[31] A. J. Scott, *Sics: extending the list of solutions*, 2017.

[32] M. Grassl and A. J. Scott, "Fibonacci-lucas sic-povms", Journal of Mathematical Physics **58**, 122201 (2017).

[33] S. Hoggar, "t-Designs in Projective Spaces", European Journal of Combinatorics **3**, 233–254 (1982).

[34] A. Klappenecker and M. Rotteler, "Mutually unbiased bases are complex projective 2-designs", in Proceedings. international symposium on information theory, 2005. isit 2005. (2005), pp. 1740–1744.

[35] A. J. Scott, "Tight informationally complete quantum measurements", Journal of Physics A: Mathematical and General **39**, 13507–13530 (2006).

[36] B. Grünbaum, *Convex polytopes, 2n ed.* (Springer-Verlag, Heidelberg/New York, 2003).

[37] S. Popescu and D. Rohrlich, "Quantum nonlocality as an axiom", Foundations of Physics **24**, 379–385 (1994).

[38] J. F. Clauser and M. A. Horne, "Experimental consequences of objective local theories", Phys. Rev. D **10**, 526–535 (1974).

[39] N. Brunner, D. Cavalcanti, S. Pironio, V. Scarani, and S. Wehner, "Bell nonlocality", Rev. Mod. Phys. **86**, 419–478 (2014).

[40] J. Altepeter, E. Jeffrey, and P. Kwiat, "Photonic state tomography", in *Advances in atomic, molecular and optical physics*, Advances in Atomic, Molecular and Optical Physics (2005), pp. 105–159.

[41] G. Molina-Terriza, A. Vaziri, J. Řeháček, Z. Hradil, and A. Zeilinger, "Triggered Qutrits for Quantum Communication Protocols", Phys. Rev. Lett. **92**, 167903 (2004).

[42] E. Skovsen, H. Stapelfeldt, S. Juhl, and K. Mølmer, "Quantum state tomography of dissociating molecules", English, Physical Review Letters **91**, 090406 (2003).

[43] G. M. D'Ariano, M. D. Laurentis, M. G. A. Paris, A. Porzio, and S. Solimeno, "Quantum tomography as a tool for the characterization of optical devices", Journal of Optics B: Quantum and Semiclassical Optics **4**, S127–S132 (2002).

[44] J. O'Brien, G. Pryde, A. White, T. Ralph, and D. Branning, "Demonstration of an all-optical quantum controlled-not gate", Nature **426**, 264–7 (2003).

[45] H. Haeffner, W. Hänsel, C. Roos, J. Benhelm, D. Chek-al-Kar, M. Chwalla, T. Körber, U. Rapol, M. Riebe, P. Schmidt, C. Becher, O. Gühne, W. Dür, and R. Blatt, "Scalable multiparticle entanglement of trapped ions", Nature **438**, 643–6 (2006).

[46] C. Sayrin, I. Dotsenko, S. Gleyzes, M. Brune, J. Raimond, and S. Haroche, "Optimal time-resolved photon number distribution reconstruction of a cavity field by maximum likelihood", New Journal of Physics **14**, 115007 (2012).

[47] S. Christensen, J. Béguin, H. Sørensen, E. Bookjans, D. Oblak, J. Müller, J. Appel, and E. Polzik, "Toward quantum state tomography of a single polariton state of an atomic ensemble", New Journal of Physics **15**, 10 . 1088 / 1367 - 2630/15/1/015002 (2012).

[48] G. Mauro D'Ariano, M. G. Paris, and M. F. Sacchi, "Quantum tomography", in , Vol. 128, Advances in Imaging and Electron Physics (Elsevier, 2003), pp. 205–308.

[49] M. Paris and J. Rehacek, *Quantum state estimation*, Vol. 649 (Springer Science & Business Media, 2004).

[50] M. Guţă, J. Kahn, R. Kueng, and J. A. Tropp, "Fast state tomography with optimal error bounds", Journal of Physics A: Mathematical and Theoretical **53**, 204001 (2020).

[51] J. Shang, Z. Zhang, and H. K. Ng, "Superfast maximum-likelihood reconstruction for quantum tomography", Phys. Rev. A **95**, 062336 (2017).

[52] E. Bolduc, G. Knee, E. Gauger, and J. Leach, "Projected gradient descent algorithms for quantum state tomography", npj Quantum Information **3**, 10.1038/s41534-017-0043-1 (2017).

[53] M. S. Kaznady and D. F. V. James, "Numerical strategies for quantum tomography: alternatives to full optimization", Phys. Rev. A **79**, 022109 (2009).

[54] A. Acharya, T. Kypraios, and M. Guţă, "A comparative study of estimation methods in quantum tomography", Journal of Physics A: Mathematical and Theoretical **52**, 234001 (2019).

[55] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert, "Quantum state tomography via compressed sensing", Phys. Rev. Lett. **105**, 150401 (2010).

[56] M. Cramer, M. Plenio, S. Flammia, R. Somma, D. Gross, S. Bartlett, O. Landon-Cardinal, D. Poulin, and Y.-K. Liu, "Efficient quantum state tomography", Nature communications **1**, 149 (2010).

[57] A. Acharya and M. Guţă, "Statistical analysis of compressive low rank tomography with random measurements", Journal of Physics A: Mathematical and Theoretical **50**, 195301 (2017).

[58] D. Goyeneche, G. Cañas, S. Etcheverry, E. S. Gómez, G. B. Xavier, G. Lima, and A. Delgado, "Five measurement bases determine pure quantum states on any dimension", Phys. Rev. Lett. **115**, 090401 (2015).

[59] S. Boyd and L. Vandenberghe, *Convex optimization* (Cambridge university press, 2004).

[60] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming", J. ACM **42**, 1115–1145 (1995).

[61] Y. C. Eldar, "A semidefinite programming approach to optimal unambiguous discrimination of quantum states", IEEE Transactions on Information Theory **49**, 446–456 (2003).

[62] A. Aloy, M. Fadel, and J. Tura, "The quantum marginal problem for symmetric states: applications to variational optimization, nonlocality and self-testing", arXiv: Quantum Physics (2020).

[63] M. Navascués, S. Pironio, and A. Acín, "A convergent hierarchy of semidefinite programs characterizing the set of quantum correlations", New Journal of Physics **10**, 073013, 073013 (2008).

[64] A. Hald, "On the history of maximum likelihood in relation to inverse probability and least squares", Statistical Science **14**, 214–222 (1999).

[65] I. J. Myung, "Tutorial on maximum likelihood estimation", J. Math. Psychol. **47**, 90–100 (2003).

[66] P. Phillips and J. Yu, "Maximum likelihood and gaussian estimation of continuous time models in finance", Handbook of Financial Time Series, 10.1007/978-3-540-71297-8_22 (2007).

[67] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016).

[68] Z. Hradil, J. Řeháček, J. Fiurášek, and M. Ježek, "3 maximum-likelihood methodsin quantum mechanics", in *Quantum state estimation*, edited by M. Paris and J. Řeháček (Springer Berlin Heidelberg, Berlin, Heidelberg, 2004), pp. 59–112.

[69] S. H. Strogatz, *Nonlinear dynamics and chaos - with applications to physics, biology, chemistry and engineering* (Addison-Wesley, 1994).

[70] J. V. Neumann, "On rings of operators. reduction theory", Annals of Mathematics **50**, 401–485 (1949).

[71] I. Halperin, "The product of projection operators", Acta Sci. Math. **23**, 96–99 (1962).

[72] C. J. Pang, "Accelerating the alternating projection algorithm for the case of affine subspaces using supporting hyperplanes", Linear Algebra and its Applications **469**, 419–439 (2015).

[73] S. Kaczmarz, "Angenäherte auflösung von systemen linearer gleichungen", Bulletin International de l'Academie Polonaise des Sciences et des Lettres **35**, 355–357 (1937).

[74] B. Halpern, "Fixed points of nonexpanding maps", Bulletin of the American Mathematical Society **73**, 957–961 (1967).

[75] M. Kuczma, B. Choczewski, and R. Ger, *Iterative functional equations* (Cambridge University Press, 1990).

[76] https://github.com/qMLE/qMLE.

[77] https://github.com/gsenno/qStateEstimation.

[78] B. Efron and R. J. Tibshirani, *An introduction to the bootstrap*, Monographs on Statistics and Applied Probability 57 (Chapman & Hall/CRC, Boca Raton, Florida, USA, 1993).

[79] U. Leonhardt, "Quantum-state tomography and discrete wigner function", Phys. Rev. Lett. **74**, 4101–4105 (1995).

[80] R. Walser, J. I. Cirac, and P. Zoller, "Magnetic tomography of a cavity state", Phys. Rev. Lett. **77**, 2658–2661 (1996).

[81] S. WEIGERT, "Simple minimal informationally complete measurements for qudits", International Journal of Modern Physics B **20**, 1942–1955 (2006).

[82] J.-P. Amiet and S. Weigert, "Reconstructing the density matrix of a spinsthrough stern-gerlach measurements: II", Journal of Physics A: Mathematical and General **32**, L269–L274 (1999).

[83] G. M. D Ariano, L. Maccone, and M. Paini, "Spin tomography", Journal of Optics B: Quantum and Semiclassical Optics **5**, 77–84 (2003).

[84] D. Mayers and A. Yao, "Quantum cryptography with imperfect apparatus", in Proceedings 39th annual symposium on foundations of computer science (cat. no.98cb36280) (1998), pp. 503–509.

[85] M. Pawłowski and N. Brunner, "Semi-device-independent security of one-way quantum key distribution", Phys. Rev. A **84**, 010302 (2011).

[86] L. Zhou, Y.-B. Sheng, and G.-L. Long, "Device-independent quantum secure direct communication against collective attacks", Science Bulletin **65**, 12–20 (2020).

[87] A. Acín, S. Massar, and S. Pironio, "Randomness versus nonlocality and entanglement", Phys. Rev. Lett. **108**, 100402 (2012).

[88] D. Collins, N. Gisin, N. Linden, S. Massar, and S. Popescu, "Bell inequalities for arbitrarily high-dimensional systems", Phys. Rev. Lett. **88**, 040404 (2002).

[89] T. Vidick and S. Wehner, "More nonlocality with less entanglement", Phys. Rev. A **83**, 052310 (2011).

[90] S. Gómez, A. Mattar, I. Machuca, E. S. Gómez, D. Cavalcanti, O. J. Farías, A. Acín, and G. Lima, "Experimental investigation of partially entangled states for device-independent randomness generation and self-testing protocols", Phys. Rev. A **99**, 032108 (2019).

[91] M. Junge and C. Palazuelos, "Large violation of bell inequalities with low entanglement", Communications in Mathematical Physics **306**, 695–746 (2011).

[92] Y. Zhang, S. Glancy, and E. Knill, "Asymptotically optimal data analysis for rejecting local realism", Phys. Rev. A **84**, 062118 (2011).

[93] Y. Zhang, S. Glancy, and E. Knill, "Efficient quantification of experimental evidence against local realism", Phys. Rev. A **88**, 052119 (2013).

[94] Y.-C. Liang and Y. Zhang, "Bounding the plausibility of physical theories in a device-independent setting via hypothesis testing", Entropy **21**, 10.3390/e21020185 (2019).

[95] B. G. Christensen, Y.-C. Liang, N. Brunner, N. Gisin, and P. G. Kwiat, "Exploring the limits of quantum nonlocality with entangled photons", Phys. Rev. X **5**, 041052 (2015).

[96] C. Č. Brukner, M. Żukowski, J.-W. Pan, and A. Zeilinger, "Bell's inequalities and quantum communication complexity", Phys. Rev. Lett. **92**, 127901 (2004).

[97] A. Tavakoli, M. Żukowski, and Č. Brukner, "Does violation of a Bell inequality always imply quantum advantage in a communication complexity problem?", Quantum **4**, 316 (2020).

[98] P.-S. Lin, D. Rosset, Y. Zhang, J.-D. Bancal, and Y.-C. Liang, "Device-independent point estimation from finite data and its application to device-independent property estimation", Phys. Rev. A **97**, 032309 (2018).

[99] D. Kraft, "A software package for sequential quadratic programming", Tech. Rep. DFVLR-FB 88-28, DLR (1988).

[100] https://docs.scipy.org/doc/scipy/reference/optimize.minimize-slsqp.html.

[101] https://github.com/Danuzco/nonlocality_certification.

[102] S. Gómez, D. Uzcátegui, I. Machuca, E. S. Gómez, S. P. Walborn, G. Lima, and D. Goyeneche, "Optimal strategy to certify quantum nonlocality", Scientific Reports **11**, 20489 (2021).

[103] A. Aspect, P. Grangier, and G. Roger, "Experimental realization of einstein-podolsky-rosen-bohm gedankenexperiment: a new violation of bell's inequalities", Phys. Rev. Lett. **49**, 91–94 (1982).

[104] J. S. Bell and A. Aspect, *Speakable and unspeakable in quantum mechanics: collected papers on quantum philosophy*, 2nd ed. (Cambridge University Press, 2004).

[105] E. Santos, "Critical analysis of the empirical tests of local hidden-variable theories", Phys. Rev. A **46**, 3646–3656 (1992).

[106] N. Brunner, N. Gisin, V. Scarani, and C. Simon, "Detection loophole in asymmetric bell experiments", Phys. Rev. Lett. **98**, 220403 (2007).

[107] B. Hensen et al., "Loophole-free bell inequality violation using electron spins separated by 1.3 kilometres", Nature **526**, 10.1038/nature15759 (2015).

[108] X. Yu, T. Simnacher, N. Wyderka, H. C. Nguyen, and O. Gühne, "A complete hierarchy for the pure state marginal problem in quantum mechanics", Nature Communications **12**, https://doi.org/10.1038/s41467-020-20799-5 (2021).

[109] M. Navascués, F. Baccari, and A. Acín, "Entanglement marginal problems", arXiv: Quantum Physics (2020).

[110] Y.-K. Liu, M. Christandl, and F. Verstraete, "Quantum computational complexity of the N-representability problem: qma complete", Phys. Rev. Lett. **98**, 110503 (2007).

[111] A. A. Klyachko, "Quantum marginal problem and n-representability", Journal of Physics: Conference Series **36**, 72–86 (2006).

[112] A. Aloy, M. Fadel, and J. Tura, "The quantum marginal problem for symmetric states: applications to variational optimization, nonlocality and self-testing", New Journal of Physics **23**, 033026 (2021).

[113] L. Diósi, "Three-party pure quantum states are determined by two two-party reduced states", Phys. Rev. A **70**, 010302 (2004).

[114] N. Wyderka, F. Huber, and O. Gühne, "Almost all four-particle pure states are determined by their two-body marginals", Phys. Rev. A **96**, 010102 (2017).

[115] F. Mezzadri, "How to generate random matrices from the classical compact groups", English, Notices of the American Mathematical Society **54**, 592–604 (2007).

[116] K. Życzkowski, K. A. Penson, I. Nechita, and B. Collins, "Generating random density matrices", Journal of Mathematical Physics **52**, 062201 (2011).

[117] F. Pastawski, B. Yoshida, D. Harlow, and J. Preskill, "Holographic quantum error-correcting codes: toy models for the bulk/boundary correspondence", Journal of High Energy Physics **2015**, 10.1007/JHEP06(2015)149 (2015).

[118] W. Helwig, W. Cui, J. I. Latorre, A. Riera, and H.-K. Lo, "Absolute maximal entanglement and quantum secret sharing", Phys. Rev. A **86**, 052335 (2012).

[119] W. Helwig and W. Cui, *Absolutely maximally entangled states: existence and applications*, 2013.

[120] https://www.tp.nt.uni-siegen.de/+fhuber/ame.html.

[121] A. Higuchi and A. Sudbery, "How entangled can two couples get?", Physics Letters A **273**, 213–217 (2000).

[122] K. Sakurai and H. Iiduka, "Acceleration of the halpern algorithm to search for a fixed point of a nonexpansive mapping", Fixed Point Theory and Applications **2014**, 202 (2014).

[123] https://github.com/Danuzco/qmp_algorithm.