Density Matrix

Assignment 6

Gabriel Amorosetti[1*]

[1]MSc Physics of Data, Università degli Studi di Padova

Abstract

This report investigates the formalism and computational methods for describing composite quantum systems, with a focus on separable and entangled states. The study highlights the importance of density matrices in representing pure and mixed states, especially for systems with complex subsystems. Key concepts, such as the reduced density matrix obtained via partial trace, are explored to understand correlations and entanglement. Theoretical frameworks are complemented with Python implementations for constructing separable and general quantum states, generating density matrices, and computing reduced density matrices. The scalability and efficiency of these algorithms are evaluated, providing insights into their computational demands as the number of subsystems increases. Concrete examples are presented to validate the methods and their practical applications.

Introduction

The quantum states of composite systems can be sorted into two groups : separable states, that can be decomposed as products of the sub-systems' states, and entangled (or non-separable) states, that cannot be expressed in a decomposed way. These entangled states are at the core of quantum technologies, and characterizing them is essential for understanding the properties of complex systems.

To describe quantum states, the density matrix formalism is indispensable. This representation generalizes state vectors to include mixed states (statistical mixtures of pure states) and enables the modelling of open systems interacting with their environment. In particular, reduced density matrices, obtained via partial trace, provide a description of the subsystems of a global system. Analysing these matrices allows for the exploration of concepts such as entanglement and correlations between subsystems. However, working with composite systems presents several challenges due to increasing complexity as the number of subsystems grows, and optimized algorithms are needed in order to ensure computational efficiency.

This report focuses on presenting the formalism used to describe composite systems (for separable and entangled states) and to construct density matrices, as well as their numerical implementations with Python. These algorithms will be evaluated in terms of efficiency and tested with concrete examples.

*gabriel.amorosetti@studenti.unipd.it

Due by : December 10, 2024

Methodology

Entangled and separable quantum states

Formalism We consider a quantum system composed of N subsystems, each described by a wave function $|\psi_i\rangle \in \mathscr{H}_i$ where $\mathscr{H}_i = \mathscr{H}^D$ is a D-dimensional Hilbert space (assuming that the Hilbert space of each subsystem has the same dimension D).

The total wave function of the system is $|\Psi_T(\psi_1, \psi_2, \dots, \psi_N)\rangle$ and $|\Psi_T\rangle \in \mathscr{H}_T$, Hilbert space of dimension D^N, where $\mathscr{H}_T = \mathscr{H}_1 \otimes \mathscr{H}_2 \otimes .. \otimes \mathscr{H}_N$.

The general expression of the quantum state $|\Psi_T\rangle$ is a superposition of tensor products of the basis states of each subsystem :

$$|\Psi_T\rangle = \sum_{j_1,j_2,\dots,j_N} c_{j_1,j_2,\dots,j_N} |j_1\rangle_1 \otimes |j_2\rangle_2 \otimes \cdots \otimes |j_N\rangle_N$$

with the basis vectors built from the tensor product of each subsystem basis vector :

$$\{|j\rangle_i\}_{i=1,\dots,N}^{j=1,\dots,D}$$

and $c_{j_1,j_2,\dots,j_N}$ the complex coefficients verifying the normalization condition :

$$\sum_{j_1,j_2,\dots,j_N} \left|c_{j_1,j_2,\dots,j_N}\right|^2 = 1$$

The quantum state is called non-separable or entangled if it can't be written as a tensor product of the wave functions relative to each subsystem and the coefficients $c_{j_1,j_2,\dots,j_N}$ cannot be factored into products of the subsystem coefficients. For a non-separable quantum state, all the D^N combinations of coefficients are described by 2 complex parameters. Moreover, the normalization condition and the global phase invariance allow us to remove two degrees of freedom in the description of the system. Therefore, the system

is described by $2(D^N - 1)$ parameters. According to that, a non-separable state should scale exponentially with N.

The quantum state $|\Psi_T\rangle$ is separable if it can be written as a tensor product of the wave functions relative to each subsystem, i.e. $|\Psi_T\rangle = |\Psi_T\rangle_S$:

$$|\Psi_T\rangle_S = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_N\rangle$$

or

$$|\Psi_T\rangle_S = \sum_{j_1} c_{j_1}|j_1\rangle \otimes \sum_{j_2} c_{j_2}|j_2\rangle \otimes ... \otimes \sum_{j_N} c_{j_N}|j_N\rangle$$

and the coefficients can be decomposed as a product of the coefficients of each subsystem :

$$c_{j_1,j_2,...,j_N} = c_{j_1}^{(1)} c_{j_2}^{(2)} \cdots c_{j_N}^{(N)}$$

In this case, each vector is described by D pairs of complex parameters, so $2D$ parameters. As explained previously, 2 degrees of freedom can be removed thanks to the normalization condition and the global phase invariance, meaning $2D - 2$ parameters are needed. Finally, there are N independent subsystems, so the quantum state is described by $N(2D - 2)$ parameters. Since each subsystem is treated independently, a separable state should scale linearly with N.

Implementation In order to describe the composite quantum system (considered at the beginning of the previous section) in the case of N-body non-interacting, separable pure states, we implement a function `separable_state(coeffs, dimensions)` that takes as inputs `coeffs`, a list of 1D numpy arrays of the coefficients of each subsystem's wave function, and `dimensions`, a list of dimensions (for each subsystem's Hilbert space). The function first initializes the variable `state` by assigning the first coefficient `state = coeffs[0]` and then iteratively computes the tensor product with the other subsystems `for coeff in coeffs[1:] : state = np.kron(state, coeff)` before returning `state`.

To describe the system in the case of a general N-body pure wave function in $\mathscr{H}^{D^N}$, the function `general_state(coeffs)` is implemented, where the input `coeffs` is a 1D numpy array of size D^N (Hilbert space dimension). It computes the norm of the coefficients vectors, `norm = np.linalg.norm(coeffs)` and returns `state = coeffs / norm`, the coefficients divided by the norm.

Considering the theoretical background introduced before and its implementation, we can state that both methods should scale exponentially with N. Indeed, in the general case, the function simply normalizes a vector of dimension D^N, so the scaling remains the same, but the iterative tensor product of N states in the function of the separable case leads to an exponential scaling.

Density matrices

Formalism Pure and mixed quantum states can be mathematically described by density matrices. The density matrix ρ is defined as

$$\rho = |\psi\rangle\langle\psi|$$

for a pure quantum state $|\psi\rangle$, and as

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$$

for a mixed quantum state $|\Psi(|\psi_i\rangle)\rangle$ where p_i are the probabilities associated with each pure state $|\psi_i\rangle$, satisfying $\sum_i p_i = 1$. The density matrix verifies $\forall|\psi\rangle$ $\langle\psi|\rho|\psi\rangle \geq 0$ and follows certain properties :

- $\rho^\dagger = \rho$,
- $Tr(\rho) = 1$,
- and $\rho^2 = \rho$ for a pure state only.

In order to extract the state of a subsystem from the global state of the quantum system, one might use the partial trace to compute the reduced density matrix. If we consider a bipartite system with subsystems A and B, with ρ_{AB} its density matrix, the reduced density matrix of the subsystem A is

$$\rho_A = Tr_B(\rho_{AB})$$

with Tr_B the partial trace over the degrees of freedom of the subsystem B, discarding the contributions of the subsystem B. ρ_A is expressed as

$$[\rho_A]_{ij} = \sum_k [\rho_{AB}]_{ik,jk}$$

with i, j the indices for subsystem A, and k runs over the basis states of the subsystem B. In the same way, the reduced density matrix of the subsystem B is $\rho_B = Tr_A(\rho_{AB})$ with Tr_A tracing over the subsystem A.

Implementation To write the density matrix of a general pure state $|\psi\rangle$ (given N=2), we implement the function `density_matrix(state)` that takes as input the output of the functions defined previously, and returns `matrix = np.outer(state, state.conj())`, the outer product of $|\psi\rangle$ and $|\psi^*\rangle$.

Finally, to compute the reduced density matrix given a generic matrix of dimension $D^N \times D^N$, the function `reduced_density_matrix(rho, dim, trace_over = "left"))` is implemented. It takes as input the full density matrix of size $D^2 \times D^2$ `rho`, the dimension of each subsystem `dim`, and `trace_over`, the

subsystem to trace (`left` or `right`). `rho` is reshaped into `rho_4D`, a 4D array of shape (D, D, D, D) by `rho_4d = rho.reshape(dim, dim, dim, dim)`.
If the `left` system (the first one) is chosen, the partial trace is computed over it `reduced_rho = np.trace(rho_4d, axis1 = 0, axis2 = 1)`. If the `right` system is chosen, the partial trace is computed over the second subsystem (indices 2 and 3) : `reduced_rho = np.trace(rho_4d, axis1 = 2, axis2 = 3)`.

Results

Functions tests

The functions implemented were tested with two spin-$\frac{1}{2}$ particles (qubits) in different states :

- **Separated state 1**

$$|0\rangle \otimes |1\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}$$

Outputs :

$$\text{Density matrix : } \begin{bmatrix}0 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\end{bmatrix}$$

$$\text{Reduced density matrix (left) : } \begin{bmatrix}0 & 0\\0 & 0\end{bmatrix}$$

$$\text{Reduced density matrix (right) : } \begin{bmatrix}0 & 0\\0 & 0\end{bmatrix}$$

- **Separated state 2**

$$(|0\rangle + |1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}} \left(\begin{pmatrix}1\\1\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} \right) = \frac{1}{\sqrt{2}} \begin{pmatrix}1\\1\\0\\0\end{pmatrix}$$

Outputs :

$$\text{Density matrix : } \begin{bmatrix}0.5 & 0 & 0.5 & 0\\0 & 0 & 0 & 0\\0.5 & 0 & 0.5 & 0\\0 & 0 & 0 & 0\end{bmatrix}$$

$$\text{Reduced density matrix (left) : } \begin{bmatrix}0.5 & 0\\0.5 & 0\end{bmatrix}$$

$$\text{Reduced density matrix (right) : } \begin{bmatrix}0.5 & 0\\0.5 & 0\end{bmatrix}$$

- **Entangled state 1 (Bell state $|\Phi^+\rangle$)**

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}} (|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle)$$

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}} \left(\begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} \right)$$

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}} \left(\begin{pmatrix}1\\0\\0\\0\end{pmatrix} + \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \right) = \frac{1}{\sqrt{2}} \begin{pmatrix}1\\0\\0\\1\end{pmatrix}$$

Outputs :

$$\text{Density matrix : } \begin{bmatrix}0.5 & 0 & 0 & 0.5\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0.5 & 0 & 0 & 0.5\end{bmatrix}$$

$$\text{Reduced density matrix (left) : } \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$$

$$\text{Reduced density matrix (right) : } \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$$

- **Entangled state 2 (Bell state $|\Psi^+\rangle$)**

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}} (|0\rangle \otimes |1\rangle + |1\rangle \otimes |0\rangle)$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}} \left(\begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} \right)$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}} \left(\begin{pmatrix}0\\1\\0\\0\end{pmatrix} + \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \right) = \frac{1}{\sqrt{2}} \begin{pmatrix}0\\1\\1\\0\end{pmatrix}$$

Outputs :

$$\text{Density matrix : } \begin{bmatrix}0 & 0 & 0 & 0\\0 & 0.5 & 0.5 & 0\\0 & 0.5 & 0.5 & 0\\0 & 0 & 0 & 0\end{bmatrix}$$

$$\text{Reduced density matrix (left) : } \begin{bmatrix}0 & 0\\0 & 0\end{bmatrix}$$

$$\text{Reduced density matrix (right) : } \begin{bmatrix}0 & 0\\0 & 0\end{bmatrix}$$

Efficiency evaluation

Finally, we measured the time needed to execute the functions `separable_state` and `general_state`, for different numbers of subsystems ($N = 2, 3, 4$ and 5) depending on the dimension D. These results are represented in the figure 1. All the results were obtained using libraries allowing the parallelisation of the computations on all the CPU cores in order to speed up the running times. In order to study the behaviour of the runtimes depending on the dimension,

the parameters of the power law $a \cdot b^x$ are fitted to the data and one should expect b to be close to N. However, the parallelisation of the computations reduced the runtimes and makes it difficult to evaluate the scaling according to N. Moreover, for the case $N = 2$, the fit was computed on the second half of points only (as the computation for small dimensions may be faster than the time needed to retrieve the data). For $N = 3, 4$ and 5, b is lower for the separable case, meaning the separable method is more efficient than the general one. On the contrary, the intercept a is lower for the general case ($N = 3, 4$ and 5), certainly improved by the parallelised computations.

Conclusion

In this report, we have presented the theoretical framework and computational strategies for modelling composite quantum systems, distinguishing between separable and entangled states. The analysis demonstrated the exponential scaling of computational resources with the number of subsystems for general quantum states and separable states, highlighting the challenges of scalability in quantum computations. Python implementations for constructing states and density matrices, as well as for computing reduced density matrices, have been developed and tested, confirming their alignment with the theoretical predictions. These tools provide a solid foundation for further exploration of entanglement and quantum correlations, crucial for advancing quantum technologies. Future work could focus on optimizing these methods to handle larger systems more efficiently and applying them to practical quantum computing problems.

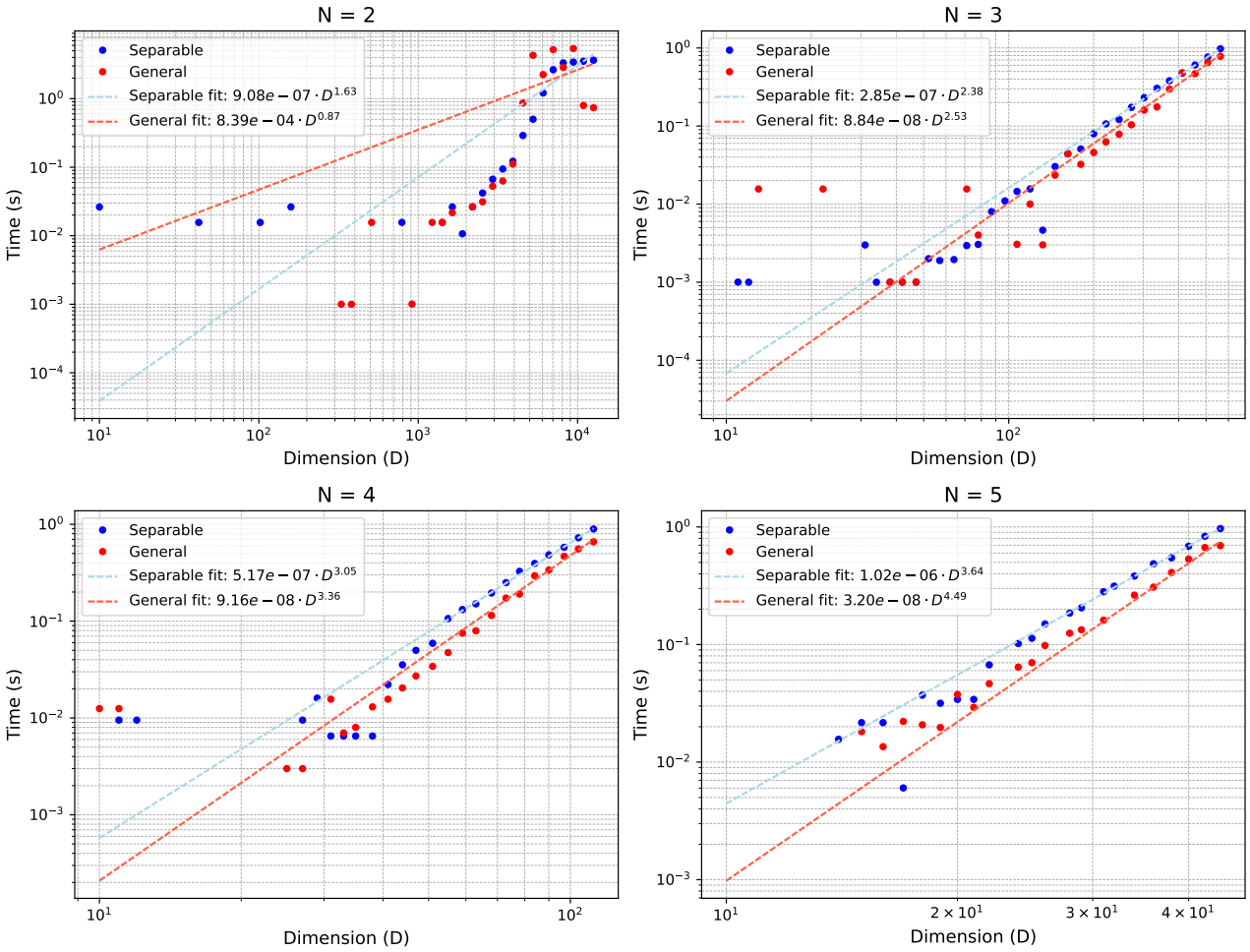

Figure 1: *Required times to build the vectors depending on the dimension, for different number of quantum subsystems*