Assignment 6 - Density matrices.

Calandra Buonaura Lorenzo
(Dated: December 5, 2024)

This paper focuses on understanding and representing composite quantum systems through the density matrix formalism, which provides a unified framework for describing both pure and mixed quantum states. Specifically, we examine the complexity of separable and general quantum states, highlight their differences in terms of parameter requirements, and explore the construction of reduced density matrices for subsystems. From this analysis it emerges how much more complex general quantum states, which require a lot of storage and computational time when the dimensions of the system increase.

1. INTRODUCTION

This paper presents a theoretical and computational framework for understanding composite quantum systems using the density matrix formalism. Composite quantum systems, which arise when multiple quantum subsystems are combined, are described within a Hilbert space which allows for the representation of a wide range of quantum states: separable states, which can be expressed as the product of subsystem states, and entangled states, which exhibit correlations that cannot be described classically. A key distinction between separable and general quantum states lies in the number of parameters required for their description: while separable states scale linearly in complexity with the number of subsystems, general entangled states exhibit an exponential growth in parameter space, reflecting the computational and theoretical challenges associated with analyzing large quantum systems.

The characterization and representation of these systems are often achieved through the density matrix formalism, which provides a flexible framework; for example, the partial trace operation on a composite system's density matrix enables the study of its subsystems, which is crucial for understanding entanglement and quantum correlations.

After outlining the fundamental properties and construction of density matrices, we explore the challenges and efficiencies of describing separable and general states. Complementing the theoretical analysis, we have developed a Python-based toolkit for generating, manipulating, and analyzing quantum states. This includes the capability to compute reduced density matrices and evaluate computational efficiency in terms of time and memory usage for various state representations.

2. THEORETICAL FRAMEWORK

This assignment delves into the representation of composite quantum systems; in order to study them, we need to understand how they are mathematically representable by the means of the density matrix formalism.

A. Quantum states

Consider a quantum system composed of N subsystems, where each subsystem resides in a Hilbert space of dimension D. The total Hilbert space of the system is given by the tensor product:

$$\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_N, \qquad \dim(\mathcal{H}) = D^N. \quad (2.1)$$

We can now define a pure state of the system, which is a state that is described by a single wavefunction $|\Psi\rangle \in \mathcal{H}$; we can have two different types of state [1]:

- **Separable:** the state can be expressed as a product of wavefunctions of the individual subsystems:
$$|\Psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_N\rangle, \quad (2.2)$$
where $|\psi_i\rangle \in \mathcal{H}_i$ for each subsystem i.
- **Non-separable (general):** the state cannot be expressed in the above form, indicating entanglement between the subsystems. A general entangled state for two subsystems can be written as:
$$|\Psi\rangle = \sum_i c_i |i\rangle_1 \otimes |j\rangle_2 \cdots \otimes |j\rangle_N, \quad (2.3)$$
where each $|i\rangle$ is a basis state of $\mathcal{H}_i$ and c_i are complex coefficients that do not factorize into a product of independent terms.

In order to understand better the whole discussion, we must highlight that there exist another type of states, other than pure, the mixed states; these represent a statistical mixture of several pure states rather than a single, well-defined pure state. Unlike a pure state, which is described by a single wavefunction $|\psi\rangle$, a mixed state describes a situation where the system has a probability p_i of being in each pure state $|\psi\rangle_i$. These states are not part of the assignment, but they are the reason why the density matrix formalism is introduced.

B. Number of parameters

The number of parameters required to fully describe a quantum state depends on whether the state is general or separable:

- **Separable state:** Given a separable state, for each subsystem, the number of real parameters required is $2D-2$ (similar to the single-system normalization and phase freedom). For N subsystems, the total number of real parameters is:

$$\text{Number of parameters} = N(2D-2). \tag{2.4}$$

- **General state:** For a general state, each complex coefficient has two real parameters (magnitude and phase), and one degree of freedom is fixed by normalization; so, the total number of independent real parameters is:

$$\#\text{ of parameters} = 2D^N - 2. \tag{2.5}$$

As we can see, the number of parameters for a general quantum state grows exponentially with the number of subsystems N, while, in contrast, for a separable state, the parameter count grows only linearly with N [1]. This difference reflects the complexity of describing entangled states compared to separable states, which has a strong impact also in terms of efficiency and storage.

C. Density matrix representation

The density matrix ρ provides a general description of quantum states, whether pure or mixed. For a pure state $|\Psi\rangle$, the density matrix is defined as:

$$\rho = |\Psi\rangle\langle\Psi|. \tag{2.6}$$

In the case of a mixed state, instead, it is:

$$\rho = \sum_i p_i |\Psi_i\rangle\langle\Psi_i|, \tag{2.7}$$

where $0 \leq p_i \leq 1$ and $\sum_i p_i = 1$ [1].

D. Desity matrix properties

The density matrix has some relevant properties:

- **Hermiticity**:

  The density matrix ρ is Hermitian, meaning $\rho = \rho^\dagger$.

- **Unit Trace**:

  The trace of the density matrix is 1, $\text{Tr}(\rho) = 1$.

- **Non-Negativity**:

  The eigenvalues of ρ are non-negative, $\lambda_i \geq 0$.

- **Idempotence for pure states**:

  For a pure state, $\rho^2 = \rho$. For mixed states, $\rho^2 \neq \rho$.

- **Dimensionality**:

  For a quantum system in a D-dimensional Hilbert space, ρ is a $D \times D$ matrix.

- **Reduced density matrices**:

  Partial tracing over subsystems produces reduced density matrices, which describe subsystems of the composite system.

E. Reduced density matrices

To study a subsystem of a composite quantum system, we compute the so-called reduced density matrix by tracing out the degrees of freedom of the other subsystems [1]. For a bipartite system with density matrix $\rho \in \mathcal{H}_A \otimes \mathcal{H}_B$, the reduced density matrix of subsystem A is obtained by tracing out subsystem B:

$$\rho_A = \text{Tr}_B(\rho). \tag{2.8}$$

Operationally, the partial trace Tr_B involves summing over the basis states of $\mathcal{H}_B$; thus, if ρ is expressed in a basis $\{|i\rangle_A \otimes |j\rangle_B\}$, the elements of ρ_A are given by:

$$(\rho_A)_{ij} = \sum_k (\rho)_{ik,jk}, \tag{2.9}$$

where the indices i, j refer to subsystem A and k refers to subsystem B.

3. CODE DEVELOPMENT

All the functions needed for the analysis and the plotting have been collected into a python script, `density_matrix_functions.py`, which is then imported in a jupyter notebook for better visualization.

- `separable_state`: Generates a separable pure state for an N-body system, with each subsystem having dimension D, either randomly or using predefined states.

- `general_state`: Creates a general pure state in the composite Hilbert space of dimension D^N, either randomly or using a predefined state.

- `test_efficiency`: Measures the efficiency (in terms of time and memory) of generating separable and general quantum states for varying N and D.

- `plot_efficiency`: Visualizes the efficiency of state generation as heatmaps, comparing separable and general states in terms of time and memory usage.

- `build_density_matrix`: Constructs the density matrix of a quantum state.

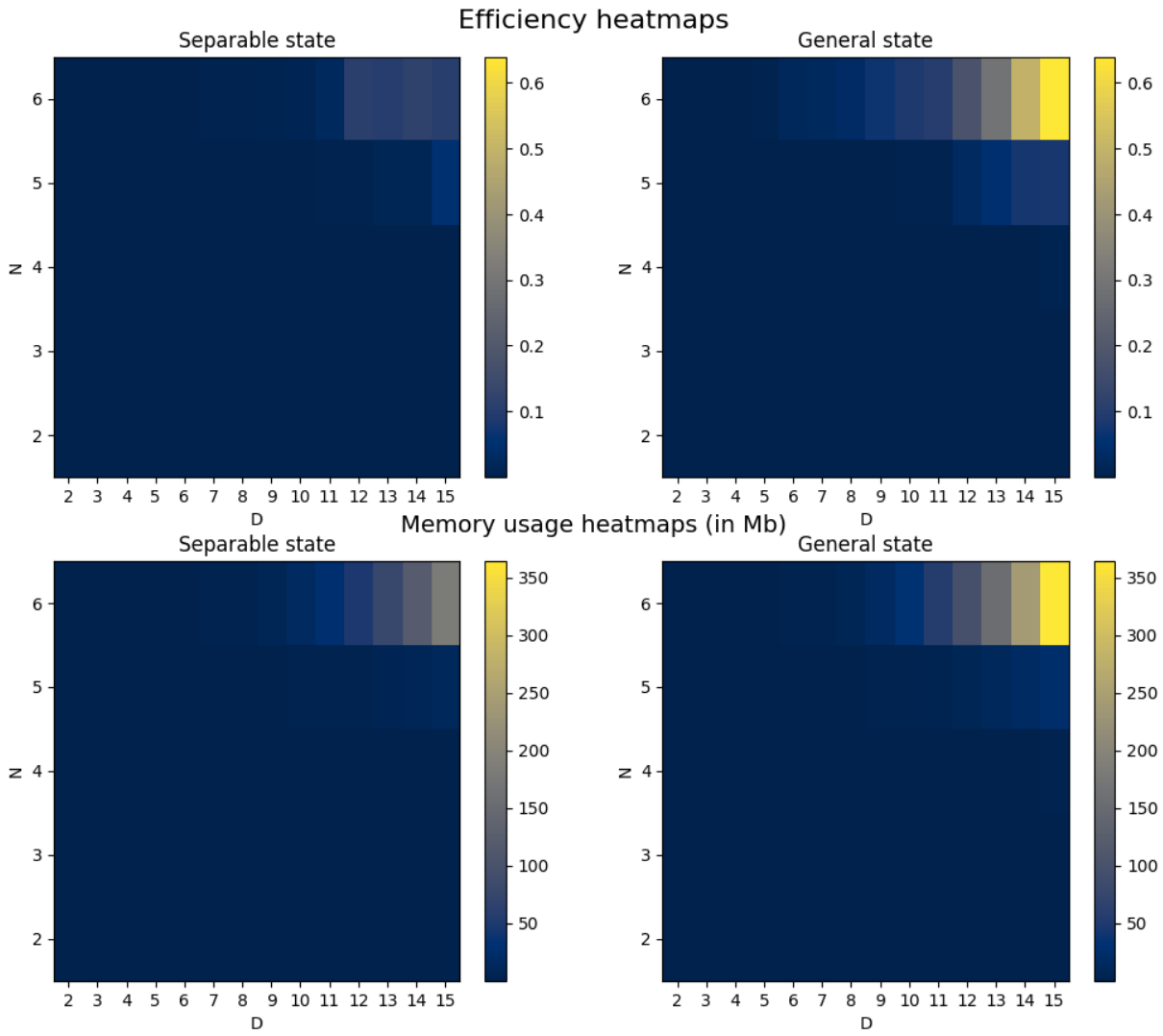

FIG. 1: Efficiency and memory usage comparison between separable states and general states ($N \leq 6$, $D \leq 15$).

- `rdm`: Computes the reduced density matrix by tracing out the degrees of freedom of the environment, keeping only the specified subsystems (indices).

- `trace_check`: Verifies whether the trace of a given density matrix is equal to 1, indicating proper normalization.

- `test_state`: Tests the accuracy of reduced density matrix calculations for a known quantum state by comparing numerical results to expected analytical results.

A. Code availability

All the data and the code for the data analysis can be found in this public Github repository: https://github.com/Kallo27/QIC/tree/main/Assignment6

4. RESULTS

The analysis has been structured in order to follow a specific logic order: we start from the correct implementation of quantum states, proceed to density matrices and reduced density matrices and we finish with a test of the correctness of the code on two-spin one-half (qubits).

Quantum states

First of all, we implement separable and general pure quantum states as described in Section 2 A. Both type of states will be vectors in an Hilbert space of dimension D^N, so when looking to the composite state we are not able to distinguish if it is separable or not (if not known). In order to highlight the correct implementation and also the structural difference between them, we can analyze

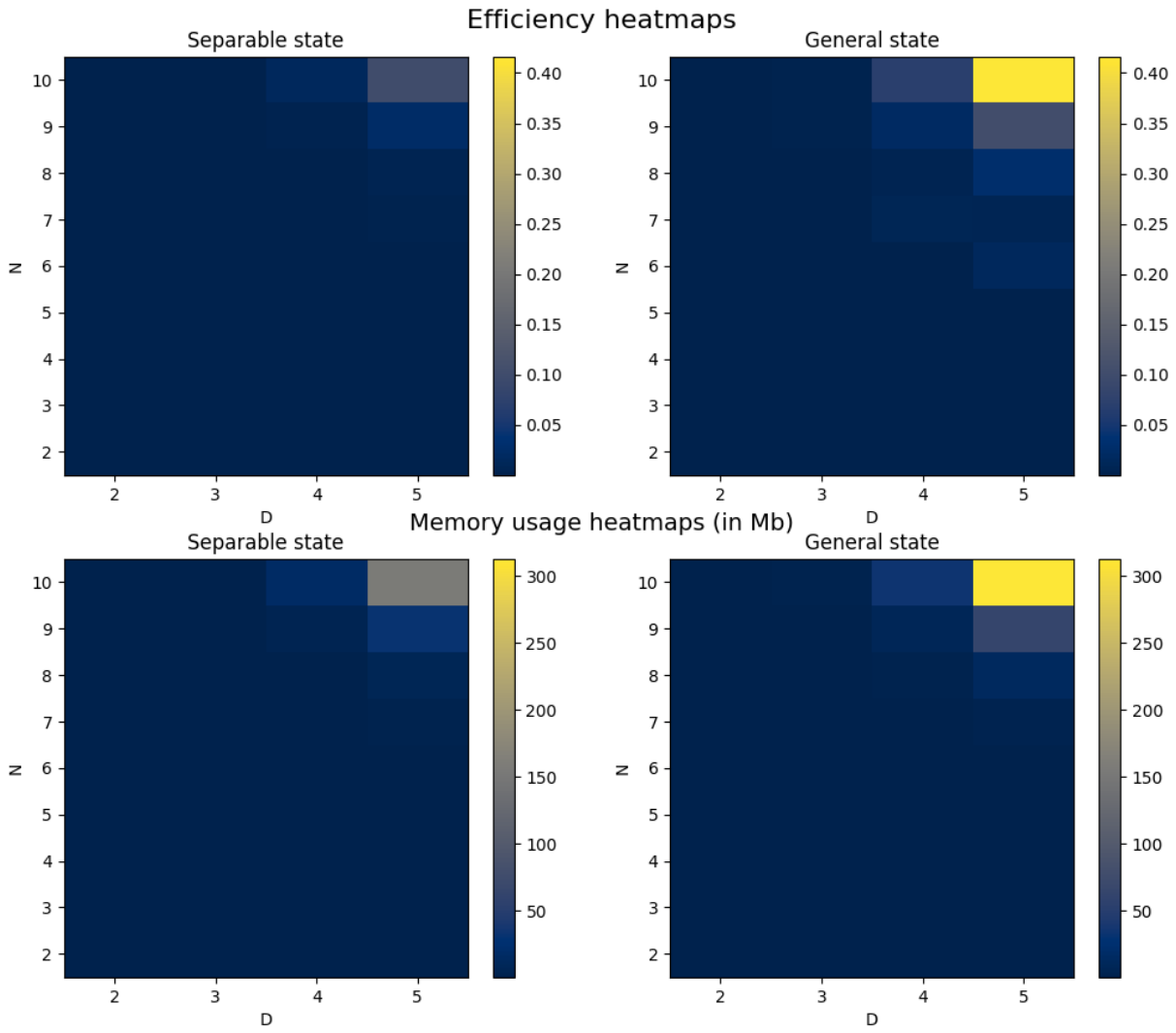

FIG. 2: Efficiency and memory usage comparison between separable states and general states ($N \leq 10$, $D \leq 5$).

them in terms of computational time needed to build them and storage needed to save them in memory. From this analysis, thanks to the fact that, as explained in Section 2 B, a different number of parameters is needed to encode them, we can highlight their difference. Figure 1 and Figure 2 show different heatmaps, where the efficiency and the memory usage are plotted for different values of N and D; we used two different heatmaps in order to explore more combination of these two parameters (high N and low D, high D and low N) without reaching states that would require much more computational power (high D and high N). From these, we can get some conclusions:

- **Efficiency:** In terms of computational time, we see that the building of a quantum state depends a lot on the values of D and N; as expected, for higher values we have an higher computational time, both for separable and general states. We must highlight that, when using the same scale for the colorbar, we can see that the separable state is nowhere near the values reached for the general state; this is in agreement with what expected, as the number of parameters scales differently for the type of states. We can also notice that we able to plot only up to $N = 10$, as the number of parameters scales as D^N for the general state, reaching very fast the computational limit.

- **Memory usage:** In terms of storage space, we must consider both the space needed to store the `np.ndarray` and the space to store the data. Similarly to the previous point, we see that the used memory scales as explained in Section 2 B, linearly for the separable state (for $N \times D = 50$ we have ~ 150 Mb and for $N \times D = 90$ we have ~ 270 Mb) and exponentially for the general state. Also in this case we must stop for a low number of both N and D in order not to reach the computational limit.

Density matrix

After the correct implementation of quantum states for composite systems, we can define the density matrix and the reduced density matrix as explained in Section 2 C and in Section 2 E. Once the pure states are correctly normalized, the density matrix is simply obtained through the outer product; in order to check if it is computed correctly, we can leverage the properties presented in Section 2 D: for all the constructed density matrices, we have that the trace is 1, they are hermitian and $\rho^2 = \rho$, which tells us the implementation is correct.

Considering the reduced density matrices, we must recover a new density matrix every time we compute the reduction; again we obtain that all the obtained density matrices have trace is 1, are hermitian and $\rho^2 = \rho$. This indicates that the implementation could be correct, but in order to have a complete proof we have to do the calculations and see whether the numerical and the analytical results are the same. Due to the fact that these calculations are not so fun to do, we test the implementation on known states of simple systems, two two-level systems.

Qubits

We used some known states to test the correct implementation of the functions in our code; we present here one separable state and one general state which have been used:

- For the separable state, we used the state:

$$|\psi\rangle = |0\rangle\,|+\rangle\,, \qquad \text{where } |+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}.$$

The density matrix of this state is:

$$\rho = |\psi\rangle\langle\psi| = (|0\rangle\langle 0|) \otimes (|+\rangle\langle +|)\,,$$

which explicitly expands to:

$$\rho = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

The reduced density matrices for the subsystems are:

$$\rho_1 = \mathrm{Tr}_2(\rho) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \qquad \rho_2 = \mathrm{Tr}_1(\rho) = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

- For the general state, instead, we used the Bell state:

$$|\psi_0\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

The density matrix of this state is:

$$\rho = |\psi_0\rangle\langle\psi_0| = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

The reduced density matrices for the Bell state are identical for both subsystems (maximally entangled state):

$$\rho_1 = \rho_2 = \mathrm{Tr}_2(\rho) = \mathrm{Tr}_1(\rho) = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}$$

In both cases (and also for the others tested), we obtained an agreement between the analytical and the numerical solution: this suggest that the implementation is correct.

5. CONCLUSION

In this report, we demonstrated the construction of quantum states, the computation of their density matrices, and the derivation of reduced density matrices. Through a structured analysis, we verified the correctness of our implementation by leveraging known theoretical properties and testing against well-known quantum states.

For separable and general quantum states, we highlighted the computational and memory efficiency differences, as expected from their parameter scaling. The results, shown in the form of heatmaps, confirmed that separable states require significantly fewer resources than general states, consistent with theoretical predictions.

The implementation of density matrices was validated by verifying their fundamental properties, such as normalization (trace equal to one), Hermiticity, and idempotence ($\rho^2 = \rho$). Similarly, for reduced density matrices, the same properties were observed, and their agreement with analytical results provided further confirmation of the code's accuracy.

Finally, we tested the entire implementation on some specific quantum states: in all cases, the numerical results for the density matrices and their reductions matched the analytical expectations. This agreement suggests that the developed code functions correctly across different scenarios.

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.