Quantum Information and Computing (QIaC)

Prof. Simone Montangero

Tommaso Faorlin

Università degli Studi di Padova - Physics of Data

tommaso.faorlin@studenti.unipd.it

December 12, 2021

Theory - Quantum states

We consider a **quantum system** made of N subsystem, each of them represented by its own wave function $|\psi_i\rangle \in \mathcal{H}^D$. We call D **local** and N **system** dimensions respectively. By rewriting each wave function as a linear combination of basis vectors $|\alpha_j\rangle \quad j = 1, \dots, D-1$, we can define a **generic state** of the system $|\psi_G\rangle \in \mathcal{H}^{D^N}$ as

$$|\psi_G\rangle = \sum_{\alpha_1,\dots,\alpha_N} C_{\alpha_1,\dots,\alpha_N} |\alpha_1, \dots, \alpha_N\rangle \tag{1}$$

and a **separable state** with a simple form

$$|\psi_S\rangle = \bigotimes_{i=1}^{N} |\psi_i\rangle = \bigotimes_{i=1}^{N} \left(\sum_{\alpha_i} C_{\alpha_i} |\alpha_i\rangle \right) \tag{2}$$

The generic state seeks for D^N complex coefficients, while the separable only for $N \cdot D$ of them. For what concerns the code implementation, the **functions** `sep_wf_init` and `gen_wf_init` in the `qm_utils.f90` module are similar (only the dimension of the output state changes) and they **initialize** (and **normalize**) a state with `DOUBLE COMPLEX` in $[-1, 1]$. We are not going to report them in the slides.

Theory - Density matrix

Given a generic state, we also define its $D^N \times D^N$ **density matrix** as

$$\rho = |\psi_G\rangle \langle \psi_G| = \sum_{\alpha_1,\dots,\alpha_N} C_{\alpha_1,\dots,\alpha_N} C^*_{\alpha_1,\dots,\alpha_N} |\alpha_1,\dots,\alpha_N\rangle \langle \alpha_1,\dots,\alpha_N| \tag{3}$$

We then consider a system where $N = 2$ with local dimension D, where the generic state of the composite system is $|\psi_{12}\rangle \in \mathcal{H}^{D^2}$. We can compute the **reduce density matrices** for each one of the two subsystems by starting from $\rho_{12} = |\psi_{12}\rangle \langle \psi_{12}|$ and **tracing out** the system we are not interested in

$$\rho_1 = Tr_2[\rho_{12}] = \sum_{j=1}^{D} \langle j|_2\, \rho_{12}\, |j\rangle_2 \qquad \rho_2 = Tr_1[\rho_{12}] = \sum_{j=1}^{D} \langle j|_1\, \rho_{12}\, |j\rangle_1 \tag{4}$$

where we index by j the different basis elements. The reduced matrices have dimension $D^{N-1} \times D^{N-1}$. For this simplified situation we graphically show how to compute them for $D = 2$. The colors mean that for example, $a_1 = a + f$, or $d_2 = f + p$:

$$\rho_1 = \begin{pmatrix} a_1 & b_1 \\ c_1 & d_1 \end{pmatrix} \iff \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} \qquad \rho_2 = \begin{pmatrix} a_2 & b_2 \\ c_2 & d_2 \end{pmatrix} \iff \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix}$$

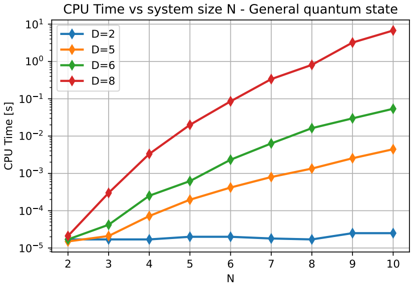
Partial trace

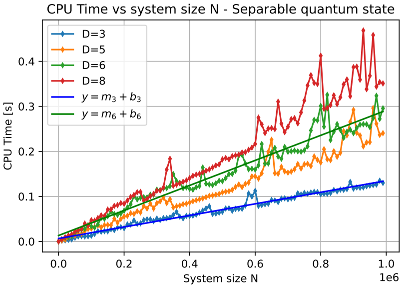
- We report only the function for the partial trace computation, since it is the most important one.

```
FUNCTION p_trace(local_dimension, dmatrix, i_subsystem) RESULT(red_dmatrix)
![...]
dN = SIZE(dmatrix,1)
d = local_dimension
![...]
IF (i_subsystem == 2) THEN
    DO aa=1, d
        row(aa) = 2*aa-1 !useful vector of indices
        col(aa) = 2*aa-1 !useful vector of indices
    ENDDO
    DO aa = 1, SIZE(row)
        DO bb = 1, SIZE(col)
            DO cc = 1, d
                !this computes the trace of each minor
                red_dmatrix(aa, bb) = red_dmatrix(aa, bb) + dmatrix(row(aa)+(cc-1), row(bb)+(cc-1))
    ELSE IF(i_subsystem == 1) THEN
        DO aa = 1, dN/d
            DO bb = 1, dN/d
                DO cc = 1, d
                    red_dmatrix(aa, bb) = red_dmatrix(aa, bb) + dmatrix(aa+(cc-1)*d, bb+(cc-1)*d)
    ELSE
        call checkpoint(.TRUE., 'The function is not built for more than two subsystems', 'e')
ENDIF
```

Results - CPU_TIME

For matters of convenience, we make use of a **Python script** named `script.py` to launch the Fortran code and save the results of the CPU_TIME required for the initialization of the states with different N and D.





The initialization requires D^N coefficients: the graph in fact shows an **exponential behavior** for the CPU_TIME. Already for $N = 10$, the time required is quite a lot, so it is **unfeasible** to initialize huge quantum system in this machine.

In the separable case things are different. The initialization time scales linearly, $m_6/m_3 \approx 2$. The results are also unstable and this may be due to the fact that the times involved are very small and can be influenced by the **instantaneous performance** of the laptop.

Results - Partial trace

We now show some results for the implementation of the partial trace operation in the case of a $N = 2$ system with local dimension $D = 2$. To obtain the partial trace one must **run the code** with `./w6.out -custom 2 2 -trace 2`. With the last line we initialize a density matrix for **two spins one half** ($N = 2$ the first, $D = 2$ the second) and we trace out (`-trace`) the subsystem 2.

$$\rho_{12} = \begin{pmatrix} 0.252025, -i0.000000 & 0.236731, +i0.043350 & 0.274330, +i0.142117 & 0.076515, +i0.171109 \\ 0.236731, -i0.043350 & 0.229821, -i0.000000 & 0.282127, +i0.086306 & 0.101304, +i0.147564 \\ 0.274330, -0.142117 & 0.282127, -i0.086306 & 0.378749, -i0.000000 & 0.179776, +i0.143106 \\ 0.076515, -i0.171109 & 0.101304, -i0.147564 & 0.179776, -i0.143106 & 0.139403, +i0.000000 \end{pmatrix}$$

The two **reduced density matrices** obtained with the partial trace are the following:

$$\rho_1 = \begin{pmatrix} 0.481846, -i0.000000 & 0.375634, +i0.289682 \\ 0.375634, -i0.289682 & 0.518153, +i0.000000 \end{pmatrix}$$

$$\rho_2 = \begin{pmatrix} 0.630774, -i0.000000 & 0.416507, +i0.186456 \\ 0.416507, -i0.186456 & 0.369225, -i0.000000 \end{pmatrix}$$

If we trace with respect to a system different from 1 or 2, the code **raises an error**. **Further checks** can be implemented, exploiting the properties of a density matrix to verify the correctness of this simple operations.