

The BraKet project offers, via an ergonomic and easy to use interface, the possibility to simulate the different probabilities associated to each component of each state of a final ket following the passage of an initial ket through several gates, each translated by an observable (hermitian operator).

1. Presentation of the software: the different graphic areas	2
1.1 Left field : path definition	2
1.2 Column in the middle: choice of gates	2
1.3 Right field: data management interface	2
1.3.1. Input: number of qubits and import of an initial ket	2
1.3.2. Output	2
1.3.3. File management	2
2. Getting started with the software	2
2.1. Some rules to follow	2
2.1.1. Rule 1	2
2.1.2. Rule 2	2
2.1.3. Rule 3	3
2.2 Manipulation steps	3
2.2.1. Importing the initial ket	3
2.2.2. Normalization	3
2.2.3. Calculation of the initial normalized ket	3
2.2.4. Calculation of the final ket	3
2.2.5. Saving	3
3. List of gates	3
3.1. Pauli-X/NOT gate (X)	3
3.2. Pauli-Y gate (Y) (not operational)	3
3.3. Pauli-Z gate (Z)	3
3.4. Hadamard gate (H)	4
3.5. Swap gate (S)	4
3.6. CNOT gate (NOT)	4
3.7. Toffoli gate (T) (not applicable here, because used for kets coded on 3 qubits)	4
3.8. Control gate (-) (not coded here)	4

1. Presentation of the software: the different graphic areas

1.1 Left field : path definition

The left field allows to define the path through which an initial ket will pass. It is a succession of gates acting on the input ket, and giving a modified ket as output.

1.2 Column in the middle: choice of gates

In the middle of the interface (between the right and left fields) there is a column with all the gate choices (within the limit of what has been coded beforehand, non-exhaustive list that can be modified by adding in the code). The list of available doors is given in part 3.

1.3 Right field: data management interface

1.3.1. Input: number of qubits and import of an initial ket

The right field offers in its upper part the opportunity to select the number of qubits for the kets to be studied by a field taking a natural number as argument (in our case, only the kets coded on one or two qubits are studied). The "Import" button allows to import an initial ket, either typed by hand or imported from a text file.

ATTENTION : the right number of qubits must be selected by the user. The kets coded on n qubits have 2^n coefficients.

1.3.2. Output

Below the input management part is the output management part. First of all, the button "Normalisation" gives in the form of a bar graph the normalized coefficients of the input ket. Then, the button "Probabilités Initiales" gives in the form of a bar chart the probabilities of the initial ket. Finally, the button "Probabilités Finales" gives in the form of a bar chart also the probabilities of the ket at the exit of the gates.

1.3.3. File management

A "Save" button is then made available to save in a text file :

- on a first line the coefficients of the initial ket
- on a second line the probabilities of the initial ket
- on a third line the probabilities of the output ket.

A "Aide" button is then provided to open the file "fr_README.pdf". An english version (this very one, "en_README.pdf") is also available. Finally, the button "Quitter" allows to quit the software.

2. Getting started with the software

2.1. Some rules to follow

2.1.1. Rule 1

To define the path in the left field, it is important to first click on one of the boxes in this field and then select one of the available doors in the central column.

2.1.2. Rule 2

Also, as warned previously in 1.3.1, it is imperative to select the right number of qubits for the desired study, and work accordingly with the right number of associated coefficients.

2.1.3. Rule 3

On the left window, only one control gate can be imposed on each column. However, several doors can be conditioned by the same control door.

2.2 Manipulation steps

2.2.1. Importing the initial ket

First, select the number of qubits on which the ket will be coded. Import a ket via one of the two methods, by hand or on a text file. The 2^n coefficients must be separated by a space " ", and they are ordered in ascending order of the associated ket (the first coefficient is the one corresponding to $|00\rangle$, the second to $|01\rangle$ etc...).

2.2.2. Normalization

Then normalize the initial ket using the "Normalisation" button. A window appears to visualize this normalization, and the data of the initial normalized ket are stored.

2.2.3. Calculation of the initial normalized ket

Then calculate the probabilities of the initial ket thanks to the button "Probabilités Initiales". The program then stores the probabilities associated with each of the coefficients composing the initial ket (square of each of the modules). A window then opens to represent the probabilities for the final ket to be in a state.

2.2.4. Calculation of the final ket

Then launch the process of calculating the final ket, by clicking on the button "Probabilités Finales". The program then stores the probabilities associated with each of the coefficients making up the final ket (square of each of the modules). A window then opens to represent the probabilities for the final ket to be in a state.

2.2.5. Saving

Then click on the "Save" button. A window will open. Enter in the field provided for this purpose the name of the output file, accompanied by the extension (".txt" is appropriate). The program then creates a text file containing on the first line the coefficients of the initial normalized ket, on the second line the coefficients of the final normalized ket, and on the third line the probabilities of each component of the final ket. The text document is located in the project folder.

3. List of gates

3.1. Pauli-X/NOT gate (X)

The Pauli-X gate, also called NOT gate (noted "X" on the interface) translates a "NOT" operator, transforming the $|0\rangle$ kets to change them to $|1\rangle$ and vice versa.

3.2. Pauli-Y gate (Y) (not operational)

The Pauli-Y gate (noted "Y" on the interface) acts on the kets $|0\rangle$ to transform them into $i|1\rangle$, and on the kets $|1\rangle$ to transform them into $-i|0\rangle$.

3.3. Pauli-Z gate (Z)

The Pauli-Z gate (noted "Z" on the interface) maintains the $|0\rangle$ kets, and transforms the $|1\rangle$ kets into their opposite $-|1\rangle$.

3.4. Hadamard gate (H)

The Hadamard gate (noted "H" on the interface) transforms the $|0\rangle$ kets into $(|0\rangle + |1\rangle)/\sqrt{2}$, and the $|1\rangle$ kets into $(|0\rangle - |1\rangle)/\sqrt{2}$.

3.5. Swap gate (S)

The Swap gate (noted "S" on the interface) swaps the qubits represented in the 2-qubit base $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. For example, the ket $|00\rangle$ remains $|00\rangle$, and the ket $|01\rangle$ is transformed into ket $|10\rangle$.

3.6. CNOT gate (NOT)

The CNOT gate (noted "NOT" on the interface) acts as a Pauli-X/NOT gate on the second qubit of the kets represented in the 2-qubit base $\{|ij\rangle \mid i,j \text{ belonging to } \{0,1\}\}$ under the following condition: if the first qubit i is 1, the second qubit is modified ($|0\rangle$ into $|1\rangle$, $|1\rangle$ into $|0\rangle$). For example, the ket $|01\rangle$ remains the ket $|01\rangle$, but the ket $|10\rangle$ is transformed into ket $|11\rangle$.

3.7. Toffoli gate (T) (not applicable here, because used for kets coded on 3 qubits)

The Toffoli gate (noted "T" on the interface) acts as a Pauli-X/NOT gate on the third qubit of the kets represented in the 3-qubit base $\{|ijk\rangle \mid i,j,k \text{ belonging to } \{0,1\}\}$ under the following condition: if the first two qubits i and j are 1, the third qubit is modified ($|0\rangle$ into $|1\rangle$, $|1\rangle$ into $|0\rangle$). For example, the ket $|101\rangle$ remains the ket $|101\rangle$, but the ket $|110\rangle$ is transformed into ket $|111\rangle$.

3.8. Control gate (-) (not coded here)

The control gate (noted "-" on the interface) combined with another gate (most often X, Y or Z, and then respectively abbreviated cX, cY or cZ) acts on the qubit where this second gate is applied in case the qubit of the control gate is $|1\rangle$. For example, the ket $|01\rangle$ passing in the gate cX (gate - combined with the gate Pauli-X) remains $|01\rangle$, the ket $|11\rangle$ becomes $|10\rangle$.

4. Brief description of the code

4.1. Global variables

All the global variables of the project are gathered in a structure, called "gDonnees". It contains, among other things, the names (one character) and the functions (matrices of sizes 2×2 or 3×3) of each gate. The number of qubits chosen is stored in the attribute "nbqubits" of the structure "gDonnees" and is used many times for the size of the data storage arrays. This structure also manages the initial and final coefficients of the kets involved.

4.2. Data

First of all, the data are initialized thanks to the void function "InitialiserDonnees()". This function allows the management of the dynamic allocation of the various tables (including those containing the matrices of each of the observables or those containing the initial data of the kets and after passage in the gates). The states are then defined for the kets brought into play during the simulation via the void function "etat_de_base()", which refers to the number of qubits chosen (via the attribute "nbqubits" of the "gDonnees" structure).

4.3. Calculation

The calculation of the ket at the output of the gates is then managed by the void function "calcul_etat_final()". The coefficients in output are then stored in "gDonnees", accessible via the attribute "coeff_fin".