HAL open science

Informatique Quantique - Mathématique (Qubit suite et fin)

Michael Rollin

To cite this version:

Michael Rollin. Informatique Quantique - Mathématique (Qubit suite et fin). Master. France. 2021, pp.8. hal-03129624

HAL Id: hal-03129624

https://hal.science/hal-03129624v1

Submitted on 2 Feb 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Informatique Quantique - Mathématique (Qubit suite et fin)

Michaël Rollin

Altran, part of Capgemini

22 janvier 2021

Table des matières

1 Introduction **1**

2 Notation de Dirac **2**
2.1 States et mesures 2

3 Application logique **2**

4 Représentation graphique **4**
4.1 Bloch Sphere 4
4.2 Qsphere 5

5 Notation $O(N)$ **6**

6 En bref **7**

1 Introduction

Tout comme avec l'informatique classique, l'informatique quantique est régie par des lois mathématiques. La majorité des algorithmes ayant été écrite par des mathématiciens et/ou des physiciens, il est nécessaire de comprendre de quelle manière sont effectuées les altérations des qubits entre eux et avec les portes logiques. Pour cela, je commencerai par vous présenter dans cet article les représentations mathématiques de l'informatique quantique. Puis j'expliquerai comment fonctionnent les représentations plus visuelles que sont la Bloch sphere et la Qsphere. Enfin un chapitre En bref résumera les différents principes à l'aide de termes plus accessibles.

2 Notation de Dirac

Les notations de Dirac sont utilisées pour décrire le fonctionnement et le comportement de particules à spin demi-entiers dans le plan complexe ($\mathbb{C}^2$) :

- ket : $|a>= \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$
- bra : $<b| = |b>^+= \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}^+ = (b_1^* b_2^*)$
- bra-ket : $<b|a>= a_1 b_1^* + a_2 b_2^* =<a|b>^* \in \mathbb{C}$
- ket-bra : $|a><b| = \begin{pmatrix} a_1 b_1^* & a_1 b_2^* \\ a_2 b_1^* & a_2 b_2^* \end{pmatrix}$

Note :
- L'indice $^+$ –> indique que nous parlons de l'inverse de notre fonction.
  Ainsi si $|b>= +1$ alors $|b>^+= -1$
- l'indice * –> indique que nous adaptons une fonction du monde complexe à un autre, on appelle cela le vecteur conjugué complexe.
  Ainsi si $<c| = \begin{pmatrix} c_1^* \\ c_2^* \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}^* = |c>^+$

Ces différents éléments vont nous permettre d'additionner[1] les changements appliqués sur un ou plusieurs qubits et de mesurer l'interaction entre chacun d'eux.

2.1 States et mesures

Nous définissons l'axe Z (δ_z) comme base de mesure par défaut, comme le fait la librairie QISKIT, ce qui nous permet de mesurer $|0>$ ou $|1>$, c'est ce que l'on appelle les états maximaux de δ_z (appelé EIGENSTATES) . De cette manière on définit que :

- $|0>= \begin{pmatrix} 1 \\ 0 \end{pmatrix}$
- $|1>= \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
- $<0|1>= 1*0+0*1=0$
- $<1|1>= 1*1+1*1=1$
- Ecriture générique : $<\psi|\psi>= 1 \quad -> \quad |\psi>= \frac{1}{\sqrt{2}}(|0>+|1>) = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}$

Il y a une infinité d'axes de mesure mais les plus communs outres l'axe δ_z sont les bases δ_x et δ_y, qui représentent les états superposés maximaux appelés $|+>$ et $|->$ et se définissent ainsi :

- $\delta_x \left\{ |+>:= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0>+|1>) \quad ; \quad |->:= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0>-|1>) \right\}$
- $\delta_y \left\{ |+i>:= \frac{1}{\sqrt{2}}(|0>+i|1>) \quad ; \quad |-i>:= \frac{1}{\sqrt{2}}(|0>-i|1>) \right\}$

3 Application logique

Maintenant que nous connaissons les équations des différents états principaux d'un qubit, nous pouvons nous demander quelles sont celles des différentes portes logiques.
Dans la théorie quantique, les portes sont unitaires, ce qui signifie qu'une porte ne s'applique qu'à un ensemble fini de qubits. À partir de là on sait que nous pouvons les représenter par une matrice unitaire sous la forme $2^n * 2^n$ où

n représente l'ensemble de qubits impacté par la porte logique.

- Porte X = $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = |0><1| + |1><0|$ –> Applique une inversion sur l'état général du qubit (0 <–> 1).
- Porte Z = $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = |0><0| - |1><1|$ -> Applique une inversion sur la phase du qubit (+ <–> −).
- Porte Y = $\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = i * PorteX * PorteZ$ -> Applique une inversion sur l'état du qubit et de la phase en fonction de i ou i = la phase de l'état du qubit $* - 1$.
- Porte de Hadamard = $\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0><0| + |0><1| + |1><0| - |1><1|)$ –> Mise en superposition du qubit.

Exemple de ces portes sur un ou plusieurs qubit(s) :

- $P.X|0>= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1>$
- $P.Z|+>= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = |->$
- $P.Y|0>= i|1>$; ou encore $P.Y|1>= -i|0>$
- $P.H|0>= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = |+>$

Pour appliquer une porte logique sur plusieurs qubits, c'est le même principe, il suffit de multiplier les matrices de chacun des qubits ensemble puis d'y multiplier la matrice de la porte logique :

- $|a> \otimes |b>= \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \otimes \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1 & b_1 \\ a_1 & b_2 \\ a_2 & b_1 \\ a_2 & b_2 \end{pmatrix}$
- $|10>= |1> \otimes |0>= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$
- P.Control NOT = $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
- $CNOT * |00>= CNOT * \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = |00>$
- $CNOT * |10>= CNOT * \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = |11>$

Dans le cas des portes de contrôle il faut que le qubit source soit à 1 afin d'activer l'effet de la porte.

Attention Les cas de figure vu ci-dessus ne concernent que les qubits dans leurs états de bases (0 ou 1) ! En cas de qubits intriqués en superposition les transformations se passent ainsi :

— $|ab> \text{–>} (H_a \otimes \mathbb{1}_b)|ab> \text{–>} CNOT \text{–>} |\psi^{ab}>$

— $|00> \text{–>} \frac{(|00>+|10>)}{\sqrt{2}} \text{–>} \frac{(|00>+|11>)}{\sqrt{2}} \text{–>} |\psi^{00}>$

— $|01> \text{–>} \frac{(|01>+|11>)}{\sqrt{2}} \text{–>} \frac{(|00>+|10>)}{\sqrt{2}} \text{–>} |\psi^{01}>$

— $|10> \text{–>} \frac{(|00>+|10>)}{\sqrt{2}} \text{–>} \frac{(|00>+|11>)}{\sqrt{2}} \text{–>} |\psi^{10}>$

— $|11> \text{–>} \frac{(|01>+|11>)}{\sqrt{2}} \text{–>} \frac{(|01>+|10>)}{\sqrt{2}} \text{–>} |\psi^{11}>$

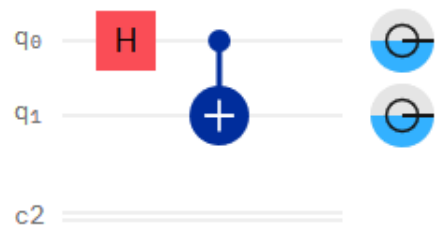

FIGURE 1 – Qubits superposés en intrication

Fort heureusement ces cas sont ce que l'on appelle les états de Bell, ce sont des identités définies que nous n'avons pas besoin de recalculer.

4 Représentation graphique

Il existe deux manières de représenter un qubit. La première, la BLOCH SPHERE donne la représentation d'un qubit et de son angle Ψ/ψ (appelé PSI) à l'aide d'un vecteur correspondant à son état (0, 1 ou intriqué) ; la seconde, la QSPHERE est une représentation des probabilités d'états d'un ou plusieurs qubits et de leur phase.

4.1 BLOCH SPHERE

Pour se représenter plus simplement un qubit et son état actuel, nous utilisons la BLOCH SPHERE. La BLOCH SPHERE est un objet en trois dimensions ayant un pôle nord, un pôle sud, une face avant et une face arrière, à l'instar de la Terre ou la Lune :

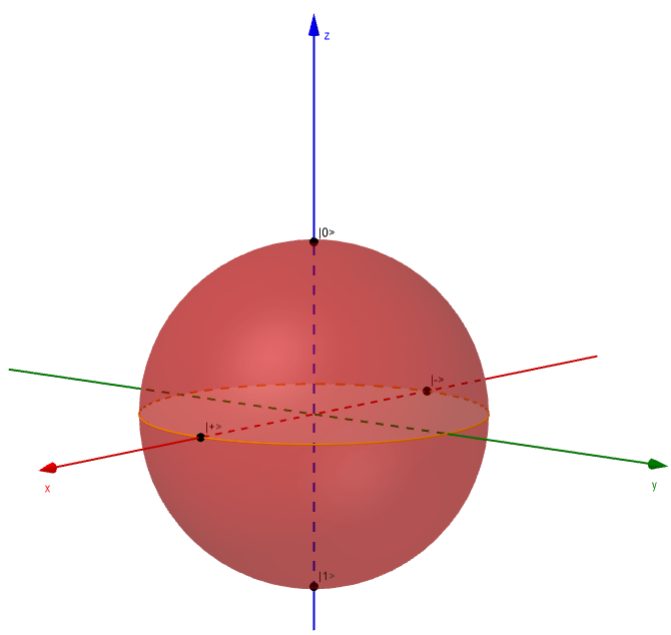

FIGURE 2 – La Bloch sphere, on peut voir les différents axes cités dans la partie précédente.

FIGURE 3 – Terre - Image par Arek Socha de Pixabay

Dans cette comparaison, on pourrait dire que nos axes δ_z, δ_x et δ_y correspondent chacun à une longitude à 360° de notre Terre et que les états maximaux $|+>$ et $|->$ correspondraient à leur croisement avec l'équateur tandis que les points $|0>$, $|1>$ seraient les pôles terrestres :

— les pôle nord et sud seraient les points $|0>$ et $|1>$.
— δ_z représenterait l'axe de l'équateur, séparant notre qubit en deux par son centre.
— δ_x représenterait la longitude à 0° aussi appelé le méridien de Greenwich à +180° l'antiméridien, ses intersections avec l'équateur seraient les points $|+>$ et $|->$.
— δ_y représenterait la longitude de 90° à - 90° ses intersections avec l'équateur seraient les points $|+i>$ et $|-i>$.

Dans la Bloch sphere, l'état d'un qubit nous est indiqué à l'aide d'un vecteur d'une longueur infinie. Seul l'angle de ce vecteur sur le plan δ_z et son déphasage sur le plan δ_x/δ_y nous importe pour connaitre l'état du qubit. Cet angle est appelé l'angle ψ (dire psi) et est calculé en fonction de deux autres angles θ (theta) et φ (phi) qui sont fonction du plan δ_z pour θ et δ_x pour φ.

— Bases principales :
  — $|0>: \theta = 0 \quad ; \quad \varphi = [0 -> \pi]$
  — $|1>: \theta = \pi \quad ; \quad \varphi = [0 -> \pi]$
  — $|+>: \theta = \frac{\pi}{2} \quad ; \quad \varphi = 0$
  — $|->: \theta = \frac{\pi}{2} \quad ; \quad \varphi = \pi$
  — $|+i>: \theta = \frac{\pi}{2} \quad ; \quad \varphi = \frac{\pi}{2}$
  — $|-i>: \theta = \frac{\pi}{2} \quad ; \quad \varphi = \frac{3\pi}{2}$
— Exemple concret :
  — Ici notre vecteur $|\psi>$ est égal à l'angle θ sur le déphasage de φ :
  — $|\psi>: \{\theta = 34,4 \quad ; \quad \varphi = 198,9\}$

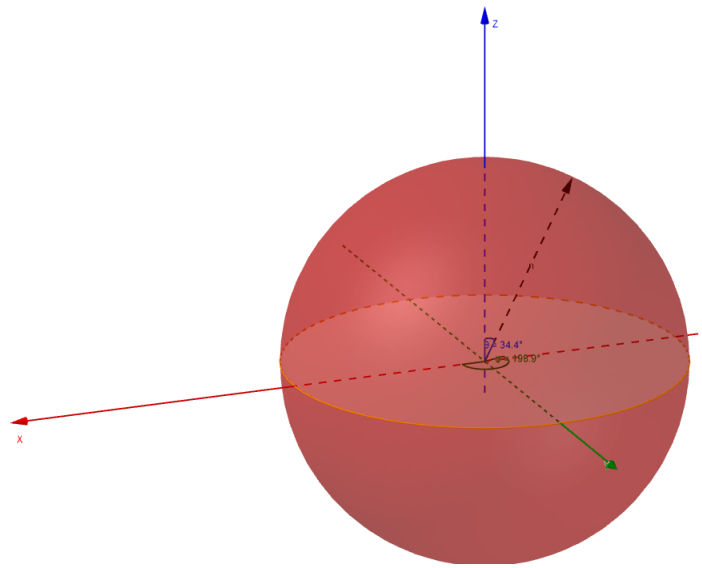

Figure 4 – La fleche noire est le vecteur pointant l'état de notre qubit, ici en intrication.

4.2 Qsphere

La Qsphere permet la représentation graphique de toutes les possibilités d'états d'un ou plusieurs qubits. La taille du point est proportionnelle à la probabilité d'obtenir ce point, et la couleur du lien correspond au déphasage du qubit dans le cas de cette possibilité.

Quelques exemples : Sur ces trois exemples, nos trois qubits possèdent la même phase sans aucun déphasage et

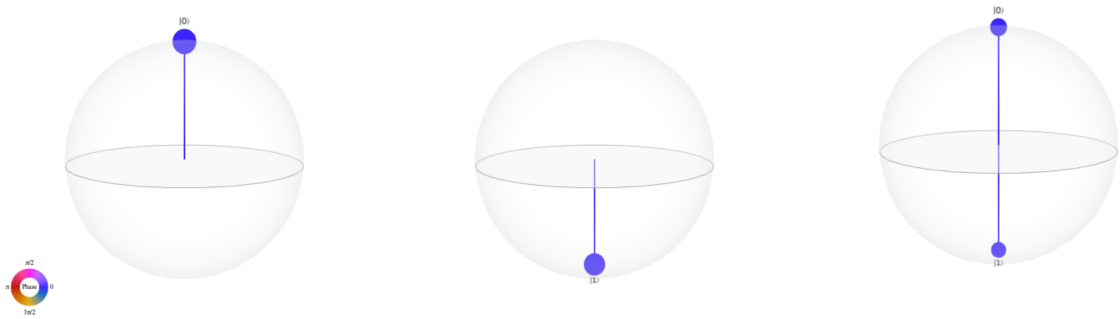

Figure 5 – De gauche à droite : un unique qubit à 0 avec 100% de chance ; un unique qubit à 1 avec 100% de chance ; un unique qubit en superposition avec 50% de chance de 0 et de 1.

avec les mêmes probabilités d'obtenir chacun des résultats possibles. Voici d'autre exemple avec un qubit ayant ses différentes possibilités sur des phases différentes :

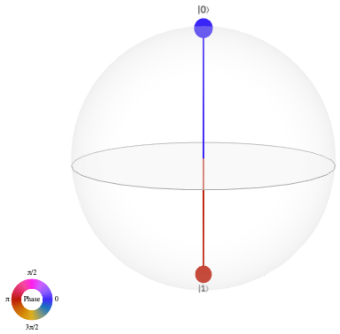

FIGURE 6 – Un unique qubit avec 50% d'obtenir chacun des résultats mais avec un déphasage de π sur $|1>$

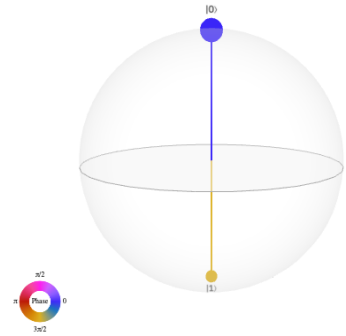

FIGURE 7 – Un unique qubit avec 75% de chance d'obtenir 0 et 25% de chances d'obtenir 1 et un déphasage de $\frac{3\pi}{2}$

Ici j'ai utilisé trois qubits, afin de représenter les quatres bases principales que l'on a vues ci-dessus :

- $|+>= (|bleu>)$
- $|->= -1 * (|rouge>)$
- $|+i>= \sqrt{2} * i * (|rose>)$
- $|-i>= \sqrt{2} * -i * (|jaune>)$
- On remarque qu'il y a plus de probabilité d'obtenir le troisième qubit (le plus à droite) à 0 plutôt qu'à 1.

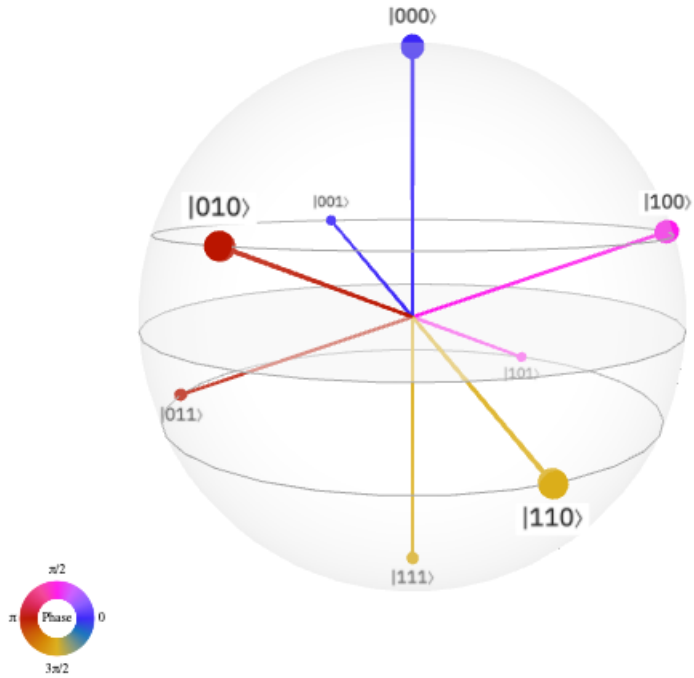

FIGURE 8 – Trois qubits, un sur chacune des phases principales et avec des probabilités différentes d'obtention de tel ou tel résultat.

5 Notation $O(N)$

Pour beaucoup d'algorithmes afin de faciliter la compréhension et de définir une valeur de temps utile ou de complexité moyenne, la notation[2] $O(N)$ ou N est utilisée pour représenter dans un cas le nombre de qubits ($N = 2^7$ dans le cas de Grover –> $O(\sqrt{N})$), le nombre d'opérations ($N = n^3$ ou n étant le nombre de digits dans le cas de Shor –> $O(N)$). Mais qu'en est-il du O ?
Pour définir ce grand O, voici un exemple :

- Nous avons deux fonctions f(x) et g(x), définies par f(x) = O(g(x)) cela signifie qu'il existe une étendue finie de nombres toujours supérieurs à 0 pour chacun des x_i de $g(x)$ tel que $M * g(x_i)$ où M serait égal à cette étendue de nombres pour tout $x > 0$. Ce qui donnerait $f(x) \leq M * g(x) \forall x > x_0$.

En clair, O est un facteur indépendant de l'algorithme et l'utiliser permet d'avoir une unique notation pour un même algorithme sans prendre en compte les spécifications de la plateforme sur lequel l'algorithme est lancé. Enfin,

celui-ci se base sur le facteur dominant où les autres sont rendus négligeables.
Exemple :

— $T(x) = 4x^2 - 2x + 2$ devient –> $T(x) = O(x^2)$.

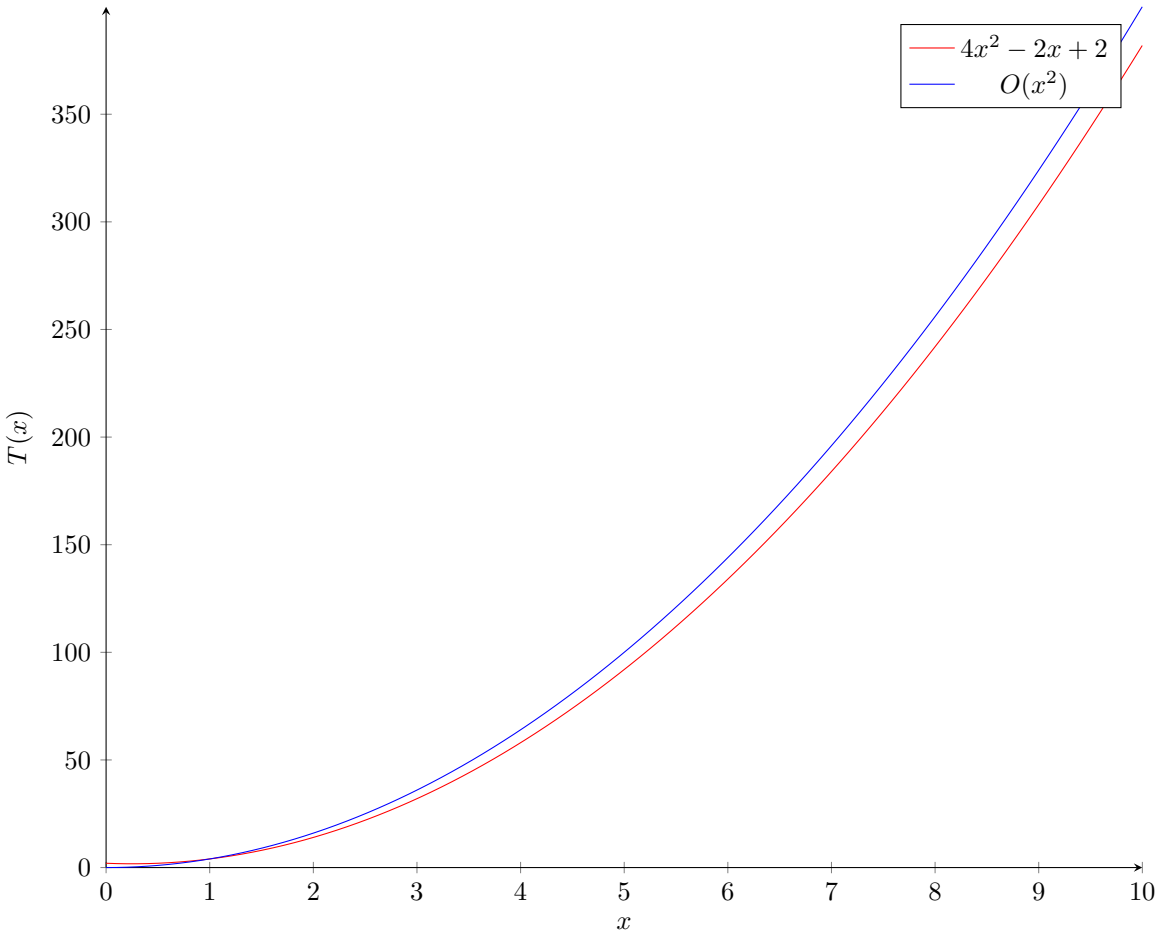

FIGURE 9 – Comparaison entre les deux fonctions, $T(x) = O(x^2)$ et $T(x) = 4x^2 - 2x + 2$

Comme on peut le voir sur cette figure, la différence entre les deux courbes peut être négligée. On conclue donc que $T(x) = 4x^2 - 2x + 2 = O(x^2)$.

6 En bref

Pour résumer, la représentation mathématique d'un qubit se nomme les équations de Dirac. Elles nous permettent de différencier un état pur d'un état superposé, ces notations se décomposant ensuite en matrices. En effet, un qubit est une sphere et son état est représenté par le vecteur qui se trouve à l'intérieur, ce qui nous permet de visualiser les interactions que les portes logiques apportent à un qubit. Lorsque différents qubits interagissent entre eux il faut les assembler pour en mesurer les interactions qu'ils subiront par la suite.
Enfin, la BLOCH SPHERE et la QSPHERE sont deux outils aidant à visualiser l'état ou les différents états de nos qubits. Ils apportent une mesure théorique de nos qubits à un instant t et ne sont donc pas soumis aux joies de la décohérence.

Disclaimer : Les informations qui m'ont servi de base à cet article proviennent d'un cours du *Qiskit Global Summer School[3]* organisé par IBM en juillet 2020.
L'ensemble des images utilisées sont opensource et on été générées soit via du code dans un environnement JUPYTER, soit avec GEOGEBRA, soit depuis l'ihm IBMQUANTUM EXPÉRIENCE.

Références

[1] Qiskit. Linear algebra, 2020. Disponible sur https://qiskit.org/textbook/ch-appendix/linear_algebra.html.

[2] Qiskit. The case for quantum, 2020. Disponible sur https://qiskit.org/textbook/ch-states/case-for-quantum.html.

[3] Qiskit. Introduction to quantum computing and quantum hardware, 2020. Disponible sur https://qiskit.org/learn/intro-qc-qh/.