



collection

DataPro

Informatique quantique

De la physique quantique
à la programmation quantique en Q#

Benoît PRIEUR



eni

Avant-propos

Partie 1 : La physique quantique

Chapitre 1

De la théorie classique à la physique quantique

| | |
|--|----|
| 1. Du classique au quantique | 15 |
| 1.1 Lumière et sauts d'énergie | 15 |
| 1.2 Lumière et matière, même combat | 16 |
| 2. L'atome | 17 |
| 3. La classification périodique des éléments | 18 |
| 4. La lumière | 19 |
| 4.1 Lumière, onde-corpuscule | 19 |
| 4.2 Le photon | 20 |

Chapitre 2

Modèle de Bohr et quantification de l'énergie

| | |
|--|----|
| 1. Pourquoi le modèle de Bohr ? | 21 |
| 1.1 Introduction | 21 |
| 1.2 Spectre électromagnétique et raies spectrales | 22 |
| 1.3 Postulat de Bohr | 23 |
| 2. Modèle de Bohr | 24 |
| 2.1 Précisions concernant les orbites | 24 |
| 2.2 Modèle de Bohr et nombres quantiques | 25 |
| 2.2.1 Premier nombre quantique n | 25 |
| 2.2.2 Second nombre quantique l | 25 |
| 2.2.3 Troisième nombre quantique m | 25 |
| 2.2.4 Quatrième nombre quantique s | 26 |
| 2.3 Nombres quantiques et représentation de l'état de l'électron . . | 26 |

2 _____ Informatique quantique

De la physique quantique à la programmation quantique en Q#

Chapitre 3

L'état quantique

| | |
|--|----|
| 1. Vers une définition de l'état quantique. | 29 |
| 1.1 Le principe d'incertitude d'Heisenberg | 29 |
| 1.2 La mesure quantique | 29 |
| 1.3 Un monde probabiliste. | 30 |
| 1.4 L'approche systémique | 31 |
| 2. L'état quantique et les mathématiques | 31 |
| 2.1 Rapide retour sur les vecteurs | 31 |
| 2.2 Rapide retour sur le calcul matriciel | 33 |
| 2.3 La notation bra-ket | 35 |
| 3. Fonction d'onde | 37 |
| 3.1 La mise en évidence graphique | 37 |
| 3.2 La fonction d'onde continue | 38 |

Chapitre 4

La superposition quantique

| | |
|--|----|
| 1. Introduction à la superposition quantique | 39 |
| 1.1 En partant de l'état quantique. | 39 |
| 1.2 Outils mathématiques | 39 |
| 1.2.1 Premier exemple avec le spin de l'électron | 40 |
| 1.2.2 Rapide retour sur les nombres complexes | 40 |
| 1.2.3 Calcul relatif à la probabilité. | 41 |
| 1.3 Second exemple : le « dé quantique » | 42 |
| 2. Interprétation de la superposition quantique. | 42 |
| 2.1 Interprétation du « dé quantique » | 42 |
| 2.2 Le paradoxe du chat de Schrödinger | 43 |
| 2.2.1 Mise en équation | 43 |
| 2.3 Le congrès Solvay de 1927 | 44 |
| 2.4 Théorie d'Everett | 45 |
| 3. Vérification de la superposition quantique. | 46 |

Chapitre 5
L'intrication quantique

- 1. Première approche de l'intrication quantique 47
 - 1.1 Propos introductif 47
 - 1.2 Mise en évidence 48
- 2. Le paradoxe EPR 49
 - 2.1 Explications du paradoxe EPR 49
 - 2.2 Interprétations du paradoxe EPR 50
- 3. La théorie de Bell 50
 - 3.1 Explications de la théorie de Bell 50
 - 3.2 Intérêt des inégalités de Bell dans le champ expérimental 51
 - 3.3 « Dieu joue aux dés » (?) ou l'expérience d'Aspect 51

Partie 2 : L'informatique quantique

Chapitre 6
Le bit quantique

- 1. Vers le bit quantique 53
 - 1.1 Du bit au bit quantique 53
 - 1.2 Sphère de Bloch 54
 - 1.3 Résumé du cycle de vie d'un qubit 56
- 2. Définition vectorielle du bit quantique 57
 - 2.1 Exemple à un qubit 57
 - 2.2 Exemple à deux qubits 57

4 Informatique quantique

De la physique quantique à la programmation quantique en Q#

Chapitre 7

La porte quantique

| | |
|---|----|
| 1. Comparaison avec l'informatique quantique | 59 |
| 1.1 Introduction | 59 |
| 1.2 Exemple de la porte logique NOT | 60 |
| 1.3 Exemple de la porte logique XOR | 61 |
| 2. Les portes quantiques à un qubit en entrée | 62 |
| 2.1 Introduction | 62 |
| 2.2 Porte de Hadamard (H) | 63 |
| 2.2.1 Utilisation de la porte de Hadamard | 64 |
| 2.3 Porte de Pauli-X (NOT) | 65 |
| 2.3.1 Utilisation de la porte de Pauli-X | 66 |
| 2.4 Portes de Pauli-Y et de Pauli-Z | 66 |
| 2.4.1 Considérations sur la porte de Pauli-Y | 67 |
| 2.4.2 Considérations sur la porte de Pauli-Z | 67 |
| 2.5 Racine carrée de la porte NOT | 68 |
| 2.6 Porte de changement de phase | 68 |
| 3. Les portes quantiques à deux qubits en entrée | 69 |
| 3.1 Introduction | 69 |
| 3.2 Porte cNOT (cX) | 69 |
| 3.3 Porte SWAP (S) | 71 |
| 3.3.1 Présentation et description | 71 |
| 3.3.2 Conception de la porte SWAT à base de la porte cNOT | 72 |
| 4. Les portes quantiques à trois qubits en entrée | 72 |
| 4.1 Porte TOF (CCNOT) | 72 |
| 4.2 Porte Fredkin (CSWAP) | 74 |
| 4.3 La mesure dans les schémas quantiques | 75 |

Chapitre 8**Définition des circuits quantiques**

| | |
|--|----|
| 1. Introduction | 77 |
| 2. Produit tensoriel | 78 |
| 2.1 Montage en série | 78 |
| 2.2 Produit de Kronecker | 79 |
| 2.3 Montage en parallèle | 80 |
| 3. Les états de Bell | 81 |
| 3.1 Explications | 81 |
| 3.2 Initialisation | 82 |
| 3.3 Application de la porte de Hadamard | 82 |
| 3.4 Application de la porte CNOT | 82 |
| 3.5 Résultat et interprétation | 83 |
| 4. Téléportation quantique | 84 |
| 4.1 Explications | 84 |
| 4.2 Scénario de téléportation | 84 |
| 4.3 Schéma quantique de téléportation | 85 |
| 4.4 Interprétation | 85 |
| 5. Classification des problèmes à résoudre et complexité | 86 |
| 5.1 L'informatique classique face à l'informatique quantique | 86 |
| 5.1.1 Complexité des problèmes à résoudre | 86 |
| 5.1.2 Parallélisation quantique ou la puissance du registre quantique | 86 |
| 6. Typologie des problèmes | 88 |
| 6.1 Contexte | 88 |
| 6.2 La complexité | 88 |
| 6.2.1 Première approche | 88 |
| 6.2.2 Complexité de l'algorithme par dichotomie | 89 |
| 6.2.3 Complexité d'un algorithme appliqué au problème du voyageur de commerce | 90 |

6 **Informatique quantique**

De la physique quantique à la programmation quantique en Q#

| | | |
|-------|---------------------------------|----|
| 6.3 | Catégorisation des problèmes | 93 |
| 6.3.1 | Les classes P et NP | 93 |
| 6.3.2 | Inclusions ? | 93 |
| 6.3.3 | Les classes EXPTIME et NEXPTIME | 94 |
| 6.3.4 | La classe NP-Complet | 94 |
| 7. | Vers la suprématie quantique ? | 95 |
| 7.1 | Introduction | 95 |
| 7.2 | L'algorithme quantique de Shor. | 96 |

Chapitre 9

Construire un ordinateur quantique

| | | |
|-------|--|-----|
| 1. | Introduction | 97 |
| 2. | Architecture de l'ordinateur quantique | 98 |
| 2.1 | Premiers éléments. | 98 |
| 2.2 | Bit quantique logique et bit quantique physique. | 99 |
| 3. | Le bit quantique physique en pratique | 99 |
| 3.1 | Approche générale | 99 |
| 3.2 | Pistes de réalisation | 100 |
| 4. | Détails de quelques pistes de réalisation | 101 |
| 4.1 | La piste de la résonance magnétique nucléaire | 101 |
| 4.1.1 | Explication du phénomène | 101 |
| 4.1.2 | RMN et ordinateur quantique | 101 |
| 4.2 | La piste « Josephson » | 101 |
| 4.2.1 | L'effet Josephson | 101 |
| 4.2.2 | Bit quantique à supraconducteurs | 102 |
| 5. | Un bref état des lieux | 102 |
| 5.1 | Typologie des acteurs. | 102 |
| 5.2 | Un état des lieux, acteur par acteur | 103 |
| 5.2.1 | D-Wave Systems | 103 |
| 5.2.2 | Rigetti Computing | 104 |
| 5.2.3 | Fujitsu | 105 |

| | | |
|-------|-----------|-----|
| 5.2.4 | Atos | 105 |
| 5.2.5 | Google | 105 |
| 5.2.6 | Intel | 105 |
| 5.2.7 | IBM | 106 |
| 5.2.8 | Microsoft | 106 |

Partie 3 : Le développement quantique avec Q#

Chapitre 10

Quantum Development Kit et son installation

| | | |
|-------|---|-----|
| 1. | Présentation générale | 107 |
| 1.1 | Annonce de la sortie | 107 |
| 1.2 | Composition du kit quantique de Microsoft | 108 |
| 2. | Installation du Microsoft Quantum Development Kit | 109 |
| 2.1 | Environnements de développements ciblés | 109 |
| 2.2 | Installation avec Microsoft Visual Studio | 109 |
| 2.2.1 | Installation de Microsoft Visual Studio | 110 |
| 2.2.2 | Installation de Microsoft Quantum Development Kit | 111 |
| 2.3 | Installation avec Visual Studio Code | 111 |
| 2.3.1 | Installation de Visual Studio Code | 111 |
| 2.3.2 | Installation du .NET Core SDK | 112 |
| 2.3.3 | Installation annexe relative à C# (facultative) | 112 |
| 2.3.4 | Installation de Microsoft Quantum Development Kit | 115 |
| 2.3.5 | Configuration relative à la ligne de commande | 115 |
| 2.4 | Test de la validité de l'installation | 116 |
| 2.4.1 | Test avec Microsoft Visual Studio | 117 |
| 2.4.2 | Test avec Visual Studio Code | 118 |
| 2.5 | Examen du projet de test de téléportation | 120 |
| 2.5.1 | Le projet de téléportation | 120 |
| 2.5.2 | Le fichier TeleportationSample.qs | 122 |

8 Informatique quantique

De la physique quantique à la programmation quantique en Q#

Chapitre 11

Le langage Q# et son utilisation

| | |
|--|-----|
| 1. Présentation du langage Q# | 127 |
| 1.1 Introduction | 127 |
| 1.2 Un langage évolutif | 127 |
| 1.3 Un langage multiparadigme. | 128 |
| 2. Éléments relatifs aux signes de ponctuation. | 128 |
| 2.1 Contexte | 128 |
| 2.2 Nomenclature relative aux signes de ponctuation. | 129 |
| 3. Les types en Q# | 129 |
| 3.1 Les types primitifs du langage | 129 |
| 4. Les structures de données | 131 |
| 4.1 Introduction | 131 |
| 4.2 Le tableau | 131 |
| 4.3 Le tuple | 132 |
| 4.3.1 Le cas particulier du tuple singleton. | 132 |
| 5. Considérations sur les types et les structures de données en Q# | 133 |
| 5.1 Définir ses propres types en Q# | 133 |
| 5.2 Les types encapsulés en Q# | 134 |
| 5.2.1 Introduction du mot-clé let | 134 |
| 5.2.2 Utiliser les types encapsulés | 134 |
| 5.2.3 Opérateur de désencapsulation | 135 |
| 6. Opérations et fonctions Q# | 136 |
| 6.1 Distinction entre opération et fonction en Q# | 136 |
| 6.2 Utiliser des « callables » dans des « callables » | 136 |
| 6.3 L'opération en Q# | 137 |
| 6.3.1 Première approche | 137 |
| 6.3.2 Un exemple avec tuple non vide en sortie | 138 |
| 6.3.3 Les mots-clés adjoint et controlled. | 139 |
| 6.4 La fonction en Q# | 140 |

| | | |
|--------|---|-----|
| 6.5 | Conditions, boucles, etc. | 140 |
| 6.5.1 | Condition if...elif | 140 |
| 6.5.2 | Boucle for | 141 |
| 6.5.3 | Boucle repeat...until. | 141 |
| 7. | Les variables en Q# | 142 |
| 7.1 | Le mot-clé let et l'immutabilité | 142 |
| 7.2 | Les mots-clés mutable et set | 142 |
| 8. | Manipuler des bits quantiques en Q# | 143 |
| 8.1 | Utiliser un registre de bits quantiques | 143 |
| 8.2 | Première approche des portes quantiques en Q# | 143 |
| 8.2.1 | Les primitives en Q# | 143 |
| 8.2.2 | Une primitive particulière, la mesure | 145 |
| 8.3 | Inventaire des primitives en Q# | 145 |
| 8.3.1 | Opération Assert | 145 |
| 8.3.2 | Opération AssertProb | 146 |
| 8.3.3 | Opération CNOT | 146 |
| 8.3.4 | Opération CCNOT | 147 |
| 8.3.5 | Opération Exp | 148 |
| 8.3.6 | Opération ExpFrac | 148 |
| 8.3.7 | Opération H | 149 |
| 8.3.8 | Opération I | 149 |
| 8.3.9 | Opération M | 149 |
| 8.3.10 | Opération Measure | 150 |
| 8.3.11 | Opération MultiX | 150 |
| 8.3.12 | Opération R | 151 |
| 8.3.13 | Opération R1 | 152 |
| 8.3.14 | Opération R1Frac | 153 |
| 8.3.15 | Opération RFrac | 153 |
| 8.3.16 | Opération Rx | 153 |
| 8.3.17 | Opération Ry | 154 |
| 8.3.18 | Opération Rz | 154 |
| 8.3.19 | Opération Random | 155 |
| 8.3.20 | Opération Reset | 155 |

10 _____ Informatique quantique

De la physique quantique à la programmation quantique en Q#

| | | |
|--------|---|-----|
| 8.3.21 | Opération ResetAll | 155 |
| 8.3.22 | Opération S | 156 |
| 8.3.23 | Opération SWAP | 156 |
| 8.3.24 | Opération T | 157 |
| 8.3.25 | Opération X | 157 |
| 8.3.26 | Opération Y | 158 |
| 8.3.27 | Opération Z | 158 |
| 9. | Tester son code quantique et le déboguer | 158 |
| 9.1 | Les tests unitaires en Q# | 158 |
| 9.1.1 | Contexte | 158 |
| 9.1.2 | Création d'un projet de tests avec Microsoft Visual Studio | 159 |
| 9.1.3 | Création d'un projet de tests avec Visual Studio Code | 159 |
| 9.1.4 | Écrire des tests unitaires | 159 |
| 9.1.5 | Exécuter des tests unitaires | 160 |
| 9.2 | Déboguer un code quantique Q# | 162 |
| 9.2.1 | Les outils des environnements de développement | 162 |
| 9.2.2 | Utiliser les messages | 162 |
| 9.2.3 | Utiliser les logs | 163 |
| 9.2.4 | Utiliser les assertions | 163 |
| 9.2.5 | Les fonctions de dump | 164 |
| 10. | Inventaire des espaces de noms quantiques | 166 |
| 10.1 | Contexte | 166 |
| 10.2 | L'espace de noms Microsoft.Quantum.Core | 166 |
| 10.3 | Les espaces Microsoft.Extensions.* | 167 |
| 10.3.1 | Microsoft.Extensions.Bitwise | 167 |
| 10.3.2 | Microsoft.Extensions.Convert | 168 |
| 10.3.3 | Microsoft.Extensions.Diagnostics | 168 |
| 10.3.4 | Microsoft.Extensions.Math | 169 |
| 10.3.5 | Microsoft.Extensions.Testing | 169 |
| 10.4 | L'espace Microsoft.Quantum.Primitive | 169 |
| 10.5 | L'espace Microsoft.Quantum.Canon | 170 |
| 11. | C# en pilotage du simulateur quantique | 170 |

Chapitre 12
Simuler des programmes quantiques en Q#

- 1. Contexte 175
- 2. Exemple appliqué : les états de Bell 175
 - 2.1 Contexte 175
 - 2.2 Les états de Bell 176
 - 2.3 Création d'un projet quantique 177
 - 2.3.1 Création d'un projet quantique avec Microsoft Visual Studio 177
 - 2.3.2 Création d'un projet quantique avec Visual Studio Code 177
 - 2.4 Programmation du fichier Q# 178
 - 2.4.1 Renommage du fichier 178
 - 2.4.2 Codage du fichier Q# 178
 - 2.5 Programmation du fichier de pilotage C# 182
 - 2.5.1 Code automatiquement généré 182
 - 2.6 Exécuter le programme quantique 184
 - 2.6.1 Exécuter le programme quantique avec Microsoft Visual Studio 184
 - 2.6.2 Exécuter le programme quantique avec Visual Studio Code 184
 - 2.7 Simuler la superposition quantique 185
 - 2.8 Simuler les états de Bell avec l'intrication quantique 188
- 3. Plus loin dans le développement Q# 192
 - 3.1 Différentes situations de mesures en Q# 192
 - 3.1.1 Contexte 192
 - 3.1.2 Le projet quantique de mesure 192
 - 3.1.3 Le pilote C# 193
 - 3.1.4 Le code quantique Q# de la mesure d'un bit quantique 195
 - 3.1.5 Le code quantique Q# de la mesure d'un registre de bits quantiques 196

12 _____ Informatique quantique

De la physique quantique à la programmation quantique en Q#

| | | |
|-------|--|-----|
| 3.1.6 | Le code quantique Q# de la mesure dans la base de Bell | 197 |
| 3.2 | Algorithme quantique de Deutsch-Jozsa | 199 |
| 3.2.1 | Contexte | 199 |
| 3.2.2 | L'algorithme quantique | 200 |
| 3.2.3 | Le pilote C# | 200 |
| 3.2.4 | Le code quantique Q# | 202 |
| 4. | Simulateur quantique : en local ou dans le cloud ? | 204 |
| 4.1 | Contexte | 204 |
| 4.2 | Simulateur distant avec Azure | 205 |

Chapitre 13

Histoire de la physique quantique

| | | |
|-------|---|-----|
| 1. | Contexte | 207 |
| 2. | Histoire chronologique : le XIXe siècle | 208 |
| 2.1 | De 1800 aux années 1860 | 208 |
| 2.2 | Années 1870 | 208 |
| 2.3 | Années 1880 | 209 |
| 2.4 | Années 1890 | 209 |
| 3. | Histoire chronologique : le XXe siècle | 210 |
| 3.1 | Années 1900 | 210 |
| 3.2 | Années 1910 | 211 |
| 3.3 | Années 1920 | 211 |
| 3.3.1 | Année 1922 | 211 |
| 3.3.2 | Année 1923 | 212 |
| 3.3.3 | Année 1925 | 212 |
| 3.3.4 | Année 1926 | 212 |
| 3.3.5 | Année 1927 | 213 |
| 3.3.6 | Année 1928 | 213 |
| 3.3.7 | Année 1929 | 214 |
| 3.4 | Années 1930 | 214 |
| 3.5 | Années 1940 | 215 |

| | | |
|------|---|-----|
| 3.6 | Années 1950 | 215 |
| 3.7 | Années 1960 | 215 |
| 3.8 | Années 1970 | 216 |
| 3.9 | Années 1980 | 216 |
| 3.10 | Années 1990 | 217 |
| 4. | Histoire chronologique : le XXI ^e siècle | 218 |
| 4.1 | Années 2000 | 218 |
| 4.2 | Années 2010 | 218 |

Chapitre 14

Glossaire de la physique quantique

| | | |
|----|------------------------|-----|
| 1. | Contexte | 219 |
| 2. | Glossaire alphabétique | 220 |
| | Index | 231 |



Partie 2

L'informatique quantique

Chapitre 6

Le bit quantique

1. Vers le bit quantique

1.1 Du bit au bit quantique

Le bit quantique dont l'écriture est habituellement simplifiée en **qubit**, voire **qu-bit**, ou encore **qbit** est l'unité de base de stockage en informatique quantique. C'est l'équivalent du bit de l'informatique classique. En effet, dans un ordinateur, l'ensemble des informations qui circulent sont réductibles à des séquences de « 0 » et de « 1 ». Pour le codage de l'information d'abord, puis lors de son traitement qui utilise intensivement des **portes logiques**. Ces portes sont réalisées matériellement avec des dizaines voire des centaines de millions de transistors qui composent nos ordinateurs. De manière symétrique, nous étudierons par la suite les **portes quantiques** de l'informatique quantique.

Dorénavant, nous prendrons l'écriture qubit pour désigner un bit quantique. Dans la littérature ou dans la presse informatique, le qubit est parfois présenté comme « un bit qui peut valoir à la fois 0 et 1 ». On reconnaît ici le principe de superposition quantique largement étudié précédemment.

54 Informatique quantique

De la physique quantique à la programmation quantique en Q#

C'est effectivement ce principe qui sous-tend la définition d'un qubit qui a comme valeur éligible à la mesure (écrites ici en notation de Dirac) :

- $|0\rangle$ et $|1\rangle$ que l'on prononce à l'oral respectivement « ket zéro » et « ket un ».
- On a donc en première approche la mise en équation suivante d'un qubit :
$$|\psi\rangle = \alpha \cdot |0\rangle + \beta \cdot |1\rangle$$

1.2 Sphère de Bloch

La sphère de Bloch est au cœur de la définition du qubit. Elle porte le nom du physicien suisse **Félix Bloch** (1905 - 1983).

La sphère de Bloch ressemble à ceci :

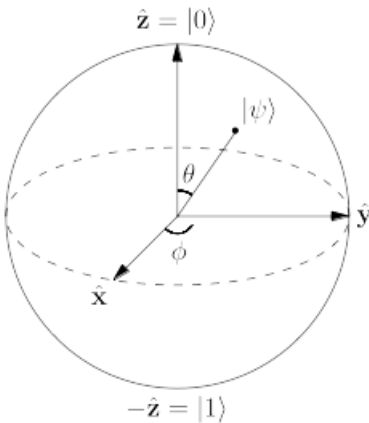


Illustration 2 : représentation de la sphère de Bloch

On a donc une sphère qui a les caractéristiques suivantes :

- La sphère est placée dans ce que l'on appelle un **espace de Hilbert**.
- L'axe z indique les deux valeurs $|0\rangle$ et $|1\rangle$.
- Le qubit consiste en un état quantique $|\psi\rangle$ qui est une combinaison linéaire des deux valeurs précitées.

- Cet état quantique (le qubit) est positionné sur la sphère et fait les angles suivants avec les différents axes du repère :
 - l'angle θ avec l'axe z.
 - l'angle ϕ avec l'axe x.

On a par ailleurs les égalités suivantes :

- Pour rappel, $|\psi\rangle = \alpha \cdot |0\rangle + \beta \cdot |1\rangle$.
- La somme des coefficients α et β , respectivement élevés au carré est égale à 1, $|\alpha|^2 + |\beta|^2 = 1$.

Signalons au passage que le qubit est en général initialisé à la grandeur $|0\rangle$.

À ce stade, rien de nouveau. En effet, on définit le qubit comme un état quantique un peu particulier, mais de la même façon que celle explicitée dans la première partie du livre. Toutefois, les différents angles définis dans la sphère de Bloch peuvent nous permettre d'explicitier d'autres équations mettant en jeu le qubit et justement ces angles. En l'occurrence, on utilise différentes projections puis la formule d'Euler pour établir l'équation suivante. Nul besoin de redémontrer l'équation pour comprendre la suite. Il est suffisant de savoir que la formule d'Euler se place dans le monde des complexes tel que nous l'avons étudié dans le chapitre La superposition quantique. Il faut juste se souvenir qu'elle est établie à partir des angles θ et ϕ de la sphère de Bloch :

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right) \cdot |0\rangle + e^{i\phi} \cdot \sin\left(\frac{\theta}{2}\right) \cdot |1\rangle$$

- Les coefficients α et β dans l'équation suivante $|\psi\rangle = \alpha \cdot |0\rangle + \beta \cdot |1\rangle$ se définissent donc ainsi :

$$\alpha = \cos\left(\frac{\theta}{2}\right)$$

$$\beta = e^{i\phi} \cdot \sin\left(\frac{\theta}{2}\right)$$

56 _____ Informatique quantique

De la physique quantique à la programmation quantique en Q#

Toujours grâce à des règles trigonométriques, on peut établir les coordonnées correspondant à ce qubit dans le repère tridimensionnel xyz associé à la sphère de Bloch :

$$- x = \sin \theta \cdot \cos \phi$$

$$- y = \sin \theta \cdot \sin \phi$$

$$- z = \cos \theta$$

1.3 Résumé du cycle de vie d'un qubit

À ce stade, nous pouvons faire un premier résumé avant de passer à la suite.

Un qubit est donc en général initialisé à $|0\rangle$.

Ensuite, l'algorithme quantique auquel il sera soumis fera varier sa valeur. On le soumet en général à ce qui est appelé des portes quantiques. Ces dernières correspondent à des opérateurs linéaires et donc réversibles. Au gré de ces opérations linéaires, le vecteur $(\alpha \beta)$ voit ses deux composantes varier, mais toujours avec une norme égale à 1. C'est-à-dire que l'on conserve forcément $|\alpha|^2 + |\beta|^2 = 1$.

Suite à la lecture du qubit, sa valeur est forcément $|0\rangle$ ou $|1\rangle$. En effet, la mesure rompt – comme nous l'avons vu précédemment - le principe de superposition. Exprimé autrement, l'observation de la valeur du qubit la fixe irrémédiablement à $|0\rangle$ ou à $|1\rangle$.

■ Remarque

Lors de l'initialisation ou lors de la lecture, on accède à une valeur du qubit égale à $|0\rangle$ ou $|1\rangle$. Entre l'initialisation et la lecture, le qubit est en état superposé et est donc à même de prendre une infinité de valeurs.

2. Définition vectorielle du bit quantique

Détaillons un peu plus encore les différentes valeurs possibles d'un qubit et ses possibles variations. Pour cela, nous allons utiliser une approche vectorielle.

2.1 Exemple à un qubit

Détaillons les valeurs $|0\rangle$ et $|1\rangle$ en explicitant les coefficients α et β dans leurs écritures vectorielles respectives.

– $|0\rangle = 1 \cdot |0\rangle + 0 \cdot |1\rangle$; ; correspondant donc au vecteur (1 0).

– $|1\rangle = 0 \cdot |0\rangle + 1 \cdot |1\rangle$; ; correspondant donc au vecteur (0 1).

On peut alors tenter d'écrire les formes vectorielles quand on a deux qubits.

2.2 Exemple à deux qubits

Quand on a deux qubits, on a évidemment quatre valeurs possibles mesurables qui sont les suivantes :

– $|00\rangle$

– $|01\rangle$

– $|10\rangle$

– $|11\rangle$

On a donc la combinaison linéaire habituelle suivante avec les coefficients α , β , γ et δ :

– $|\psi\rangle = \alpha \cdot |00\rangle + \beta \cdot |01\rangle + \gamma \cdot |10\rangle + \delta \cdot |11\rangle$

On peut donc alors et comme précédemment déterminer les vecteurs associés à chaque valeur mesurable :

– $|00\rangle = 1 \cdot |00\rangle + 0 \cdot |01\rangle + 0 \cdot |10\rangle + 0 \cdot |11\rangle$

– Correspondant donc au vecteur (1 0 0 0)

58 Informatique quantique

De la physique quantique à la programmation quantique en Q#

- $|01\rangle = 0 \cdot |00\rangle + 1 \cdot |01\rangle + 0 \cdot |10\rangle + 0 \cdot |11\rangle$
– Correspondant donc au vecteur $(0 \ 1 \ 0 \ 0)$
- $|10\rangle = 0 \cdot |00\rangle + 0 \cdot |01\rangle + 1 \cdot |10\rangle + 0 \cdot |11\rangle$
– Correspondant donc au vecteur $(0 \ 0 \ 1 \ 0)$
- $|11\rangle = 0 \cdot |00\rangle + 0 \cdot |01\rangle + 0 \cdot |10\rangle + 1 \cdot |11\rangle$
– Correspondant donc au vecteur $(0 \ 0 \ 0 \ 1)$

Les quatre vecteurs ainsi définis forment une **base vectorielle de dimension 4** nous permettant d'écrire la combinaison linéaire décrivant n'importe quel état quantique avant mesure.

■ Remarque

Pour des raisons pratiques, les vecteurs ont été écrits ici horizontalement. Rien n'empêche toutefois de les écrire verticalement.

Ainsi, on aurait pu écrire $(0 \ 0 \ 0 \ 1)$ sous cette forme $\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$

■ Remarque

On a vu dans cette partie que deux qubits assemblés deux à deux atteignent quatre états simultanés. Trois qubits assemblés permettent d'atteindre huit états, etc. selon un nombre d'états égal à $2^{\text{nombre de qubits}}$. Si le nombre de qubits dépasse 300, le nombre d'états coexistants est supérieur au nombre d'atomes dans l'univers.

Les fondements de l'informatique quantique se trouvent donc dans la définition de ce bit quantique. L'objectif à présent est d'aller en direction de l'algorithme quantique en développant et présentant la notion de porte quantique. Ensuite sera étudiée la conception même d'un ordinateur quantique et la fabrication concrète d'un bit quantique, par exemple.