Rapport de séminaire : L'algorithme quantique

RAETH Léandre, SANNA Thomas
L3 Sciences et Technologie P. Informatique

13 mars 2025

1 Introduction

1.1 Présentation du sujet

1.2 Exemple introductif

2 Informatique classique vs. Informatique quantique

2.1 Bits classiques

2.2 Qubits quantiques

3 Propriétés fondamentales de l'informatique quantique

3.1 Superposition

3.2 Intrication

3.3 Interférence

4 Génération de Nombre aléatoire

4.1 Problématique

Lors de la génération d'un nombre aléatoire en informatique traditionnelle, par exemple, en Python avec le module 'random()', le nombre généré est en réalité dit pseudo-aléatoire car l'algorithme de Mersenne Twister repose sur une graine pour initialiser le générateur de nombres aléatoires. Cette graine permet de prédire le prochain nombre pseudo-aléatoire, ce qui constitue un problème important pour de nombreux projets confidentiels, y compris la cryptographie.

Pour les projets impliquant une confidentialité sévère, comme la cryptographie, l'un des plus grands problèmes avec la génération de nombres pseudo-aléatoires est la capacité à produire des nombres élevés qui sont extrêmement difficiles à prédire. Par exemple, lors de la génération du sel du mot de passe afin de le hacher, un générateur de nombres aléatoires est utilisé. Si ce générateur est prévisible, un hacker compétent trouverait facilement le mot de passe en un rien de temps.

C'est pourquoi il est important de s'assurer que la graine choisie permettra un certain degré d'imprévisibilité.

En revanche, en informatique quantique, la génération de nombres aléatoires peut être véritablement aléatoire grâce aux propriétés quantiques expliquées il y a peu de temps. En effet, en utilisant un ordinateur quantique, on peut mesurer l'état d'un qubit en superposition pour obtenir un résultat aléatoire non-déterministe !

4.2 Fonctionnement

Pour comprendre comment fonctionne la génération de nombres aléatoires en informatique quantique, il faut se pencher sur les propriétés des qubits. Un qubit, comme on l'a vu plus tôt, peut être dans un état de superposition, ce qui signifie qu'il peut représenter simultanément les états 0 et 1. Lorsqu'on mesure un qubit en superposition, le résultat de la mesure est complètement aléatoire entre 0 et 1.

— **Préparation du qubit** : On commence par préparer un qubit dans un état de superposition. Cela peut être réalisé en appliquant une porte Hadamard (H) à un qubit initialement dans l'état $|0\rangle$. La porte Hadamard transforme l'état $|0\rangle$ en une superposition égale des états $|0\rangle$ et $|1\rangle$:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

La porte d'Hadamard est extrêmement utilisée en informatique quantique Ces superpositions ne sont pas des simulations de probabilités, mais des états réels qui peuvent être mesurés.

— **Mesure du qubit** : Une fois le qubit en superposition, on procède à sa mesure. La mesure d'un qubit en superposition donne un résultat aléatoire, soit 0 soit 1, avec une probabilité de 50% pour chaque état. Ce

processus est intrinsèquement aléatoire et ne peut pas être prédit, même si l'on connaît l'état initial du qubit.
— **Génération de séquences aléatoires** : En répétant ce processus de préparation et de mesure de qubits en superposition `n` fois, on peut générer des séquences de bits aléatoires.

Ainsi, la génération de nombres aléatoires en informatique quantique repose sur les principes fondamentaux de la mécanique quantique, offrant une source de véritable aléa, contrairement aux méthodes pseudo-aléatoires utilisées en informatique classique. Nous pourrons trouver le code de l'algorithme de génération de nombres aléatoires en informatique quantique dans la section suivante.

4.3 Comparaison des méthodes

Informatique classique	**Informatique quantique**
Pseudo-aléatoire	Non déterministe
Prédictible	Imprévisible
Basé sur des algorithmes	Basé sur des propriétés quantiques
Temps de calcul rapide	Temps de calcul plus lent

TABLE 1 – Comparaison de la génération de nombres aléatoires en informatique classique et quantique

5 Expérimentation : coder un algorithme quantique

5.1 Introduction

Il existe deux manières de coder un algorithme quantique : en utilisant un simulateur quantique ou un véritable ordinateur quantique. Pour coder sur un vrai ordinateur quantique, il est possible d'utiliser des services cloud comme IBM Quantum Experience, qui permettent d'accéder à des ordinateurs quantiques en ligne. WIKIPEDIA 2025a

Cependant, pour des raisons de simplicité, nous allons utiliser un simulateur quantique, qui permet de simuler un ordinateur quantique sur un ordinateur classique. Pour cela, nous allons utiliser le langage de programmation Qiskit.

En effet, Qiskit est un framework open-source développé par IBM en 2017 pour la programmation d'algorithme quantique en Python WIKIPEDIA 2025b. Qiskit permettra d'utiliser cette fameuse porte Hadamard pour préparer un qubit dans un état de superposition.

6 Limites et perspectives

6.1 Défis actuels

6.2 Futur de l'informatique quantique

7 Conclusion et Q&A

7.1 Récapitulatif des points clés

7.2 Session de questions-réponses

Références

Wikipedia (2025a). *IBM Quantum Platform - Wikipedia — en.wikipedia.org*. `https://en.wikipedia.org/wiki/IBM_Quantum_Platform`. [Accessed 13-03-2025].

— (2025b). *Qiskit - Wikipedia — en.wikipedia.org*. `https://en.wikipedia.org/wiki/Qiskit`. [Accessed 13-03-2025].