Introduction à l'informatique quantique

Mise en œuvre d'algorithmes de recherche et d'optimisation

Arthur Debauge

Octobre 2025

HES-SO HEPIA – Département ISC

Table des matières

1 Rappels mathématiques fondamentaux

L'informatique quantique repose sur des structures mathématiques rigoureuses issues de l'analyse fonctionnelle et de l'algèbre linéaire. Cette section pose les fondations nécessaires à la compréhension des algorithmes quantiques.

1.1 Espaces de Hilbert et postulats de la mécanique quantique

Définition 1.1 (Espace de Hilbert)**.** Un **espace de Hilbert** $\mathcal{H}$ est un espace vectoriel complexe muni d'un produit scalaire hermitien $\langle\cdot|\cdot\rangle : \mathcal{H} \times \mathcal{H} \to \mathbb{C}$ et complet pour la norme induite $\|\psi\| = \sqrt{\langle\psi|\psi\rangle}$.

Remarque 1.2. La complétude signifie que toute suite de Cauchy dans $\mathcal{H}$ converge vers un élément de $\mathcal{H}$. Cette propriété garantit la cohérence mathématique des opérations en mécanique quantique.

Définition 1.3 (Produit scalaire hermitien)**.** Le produit scalaire $\langle\cdot|\cdot\rangle$ vérifie pour tous $|\psi\rangle, |\phi\rangle, |\chi\rangle \in \mathcal{H}$ et $\alpha, \beta \in \mathbb{C}$:

1. **Linéarité à droite :** $\langle\psi|\alpha\phi + \beta\chi\rangle = \alpha\langle\psi|\phi\rangle + \beta\langle\psi|\chi\rangle$
2. **Symétrie hermitienne :** $\langle\psi|\phi\rangle = \overline{\langle\phi|\psi\rangle}$
3. **Positivité :** $\langle\psi|\psi\rangle \geq 0$ avec égalité si et seulement si $|\psi\rangle = 0$

1.2 Le qubit : unité élémentaire de l'information quantique

Définition 1.4 (Qubit)**.** Un **qubit** (quantum bit) est un système quantique à deux niveaux dont l'espace d'états est $\mathcal{H} = \mathbb{C}^2$. Il est décrit par un vecteur normalisé :

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \qquad \alpha, \beta \in \mathbb{C}, \quad |\alpha|^2 + |\beta|^2 = 1$$

où $\{|0\rangle, |1\rangle\}$ forme une base orthonormale appelée **base computationnelle**.

Représentation matricielle :

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \qquad |\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

Exemple 1.5 (États fondamentaux)**.** Quelques états remarquables :

— $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$: superposition équilibrée positive

— $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$: superposition équilibrée négative

— $|i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$: superposition avec phase complexe

1.3 Notation de Dirac (bra-ket)

Définition 1.6 (Notation bra-ket)**.**

— Un **ket** $|\psi\rangle$ désigne un vecteur colonne de $\mathcal{H}$

— Un **bra** $\langle\psi|$ désigne le vecteur dual (ligne), transposé conjugué de $|\psi\rangle$: $\langle\psi| = (|\psi\rangle)^\dagger$

— Le **bracket** $\langle\phi|\psi\rangle$ est le produit scalaire entre $|\phi\rangle$ et $|\psi\rangle$

Opérations fondamentales :

$$\langle\psi| = (\alpha^* \quad \beta^*) \quad \text{(transposé conjugué)} \tag{1}$$
$$\langle\phi|\psi\rangle = \alpha^*\gamma + \beta^*\delta \quad \text{si } |\phi\rangle = \alpha|0\rangle + \beta|1\rangle,\ |\psi\rangle = \gamma|0\rangle + \delta|1\rangle \tag{2}$$
$$|\psi\rangle\langle\phi| = \text{opérateur de projection (matrice } 2 \times 2) \tag{3}$$

1.4 Produit tensoriel et systèmes composés

Définition 1.7 (Produit tensoriel)**.** Le produit tensoriel $\otimes$ permet de construire l'espace d'états d'un système composite. Pour deux espaces $\mathcal{H}_1$ et $\mathcal{H}_2$, l'espace produit est :

$$\mathcal{H}_1 \otimes \mathcal{H}_2 = \left\{ \sum_i c_i |\psi_i\rangle \otimes |\phi_i\rangle \;\middle|\; |\psi_i\rangle \in \mathcal{H}_1,\ |\phi_i\rangle \in \mathcal{H}_2,\ c_i \in \mathbb{C} \right\}$$

Proposition 1.8 (Dimension de l'espace tensoriel)**.** *Si* $\dim(\mathcal{H}_1) = d_1$ *et* $\dim(\mathcal{H}_2) = d_2$*, alors :*

$$\dim(\mathcal{H}_1 \otimes \mathcal{H}_2) = d_1 \cdot d_2$$

Démonstration. Soit $\{|e_i\rangle\}_{i=1}^{d_1}$ une base de $\mathcal{H}_1$ et $\{|f_j\rangle\}_{j=1}^{d_2}$ une base de $\mathcal{H}_2$. Les vecteurs $\{|e_i\rangle \otimes |f_j\rangle\}_{i,j}$ forment une base de $\mathcal{H}_1 \otimes \mathcal{H}_2$, au nombre de $d_1 \times d_2$. □

Corollaire 1.9 (Registre de n qubits)**.** *Pour n qubits, l'espace d'états est :*

$$\mathcal{H}_n = \underbrace{\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2}_{n \text{ fois}} = \mathbb{C}^{2^n}$$

de dimension 2^n, avec base $\{|x\rangle\}_{x=0}^{2^n-1}$ où x est l'écriture binaire : $|x\rangle = |x_{n-1}\rangle \otimes \cdots \otimes |x_0\rangle$.

Notation compacte : Pour éviter de surcharger les formules, on note souvent :

$$|x_1 x_2 \cdots x_n\rangle \equiv |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle$$

Exemple 1.10 (Système de 2 qubits)**.** L'espace $\mathbb{C}^2 \otimes \mathbb{C}^2 = \mathbb{C}^4$ a pour base :

$$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$$

Un état général s'écrit :

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

avec $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$.

1.5 Transformations unitaires : les portes quantiques

Définition 1.11 (Opérateur unitaire)**.** Un opérateur $U : \mathcal{H} \to \mathcal{H}$ est **unitaire** si :

$$U^\dagger U = U U^\dagger = I$$

où $U^\dagger$ est l'adjoint hermitien de U (transposé conjugué) et I l'identité.

Proposition 1.12 (Conservation de la norme)**.** *Les opérateurs unitaires préservent le produit scalaire :*

$$\langle U\psi|U\phi\rangle = \langle\psi|\phi\rangle$$

En particulier, $\|U\psi\| = \|\psi\|$.

Démonstration.

$$\langle U\psi|U\phi\rangle = \langle\psi|U^\dagger U|\phi\rangle = \langle\psi|I|\phi\rangle = \langle\psi|\phi\rangle$$

□

Remarque 1.13 (Interprétation physique). Les transformations unitaires sont les seules transformations réversibles en mécanique quantique. Elles garantissent la conservation de la probabilité totale (normalisation des états).

1.5.1 Portes à un qubit fondamentales

Porte de Pauli-X (NOT quantique) :

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad X|0\rangle = |1\rangle, \quad X|1\rangle = |0\rangle$$

Porte de Pauli-Y :

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Y|0\rangle = i|1\rangle, \quad Y|1\rangle = -i|0\rangle$$

Porte de Pauli-Z (flip de phase) :

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad Z|0\rangle = |0\rangle, \quad Z|1\rangle = -|1\rangle$$

Porte de Hadamard :

Définition 1.14 (Porte de Hadamard)**.** La porte la plus importante pour créer des superpositions :

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Action sur la base computationnelle :

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = |+\rangle, \qquad H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |-\rangle$$

Proposition 1.15 (Propriétés de Hadamard)**.**

1. *H est unitaire : $H^\dagger = H$ et $H^2 = I$*
2. *H est hermitienne : $H = H^\dagger$*
3. *H est involutive : appliquer H deux fois revient à l'identité*

Vérification de l'unitarité.

$$H^2 = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} = I$$

□

1.5.2 Portes à plusieurs qubits

Porte CNOT (Controlled-NOT) :

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Action : $\text{CNOT}|c,t\rangle = |c, t \oplus c\rangle$ où $\oplus$ est le XOR.

Porte de Toffoli (CCNOT) : Porte à 3 qubits : applique un NOT au troisième qubit si et seulement si les deux premiers sont à $|1\rangle$.

1.6 Mesure quantique et principe de projection

Définition 1.16 (Mesure projective)**.** Une mesure dans la base computationnelle d'un état $|\psi\rangle = \sum_{x=0}^{2^n-1} \alpha_x |x\rangle$ donne le résultat x avec probabilité :

$$P(x) = |\alpha_x|^2 = |\langle x|\psi\rangle|^2$$

Après la mesure, l'état est projeté sur $|x\rangle$ (effondrement de la fonction d'onde).

Théorème 1.17 (Conservation de la probabilité)**.**

$$\sum_{x=0}^{2^n-1} P(x) = \sum_{x=0}^{2^n-1} |\alpha_x|^2 = \|\psi\|^2 = 1$$

Remarque 1.18 (Irréversibilité de la mesure)*.* La mesure est le seul processus non-unitaire et irréversible en mécanique quantique. Elle détruit la superposition et l'intrication.

1.7 Superposition uniforme : état fondamental des algorithmes

Définition 1.19 (Superposition uniforme)**.** L'état de superposition uniforme sur n qubits est :

$$|s\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle = H^{\otimes n}|0\rangle^{\otimes n}$$

Chaque configuration $|x\rangle$ a la même amplitude $1/\sqrt{2^n}$ et donc la même probabilité de mesure $1/2^n$.

Construction par Hadamard. Appliquons H sur chaque qubit initialisé à $|0\rangle$:

$$\begin{aligned} H|0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ H^{\otimes n}|0\rangle^{\otimes n} &= \bigotimes_{i=1}^{n} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ &= \frac{1}{2^{n/2}} \sum_{x_1 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} |x_1 \cdots x_n\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \end{aligned}$$

□

Exemple 1.20 (Superposition sur 3 qubits)**.**

$$|s\rangle = \frac{1}{\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle)$$

Chaque état a une probabilité de $1/8 = 12{,}5\%$ d'être mesuré.

1.8 Ensembles ordonnés et relations d'ordre

Définition 1.21 (Relation d'ordre partiel)**.** Un ensemble ordonné $(E, \leq)$ est un ensemble E muni d'une relation binaire $\leq$ vérifiant pour tout $x, y, z \in E$:

1. **Réflexivité :** $x \leq x$
2. **Antisymétrie :** $x \leq y$ et $y \leq x \Rightarrow x = y$
3. **Transitivité :** $x \leq y$ et $y \leq z \Rightarrow x \leq z$

Définition 1.22 (Ordre total)**.** Un ordre est dit **total** si pour tous $x, y \in E$, on a $x \leq y$ ou $y \leq x$ (comparabilité).

Définition 1.23 (Minimum global)**.** Un élément $m \in E$ est un **minimum global** si :

$$\forall x \in E, \quad m \leq x$$

Remarque 1.24. Ces structures sont essentielles pour formaliser les problèmes d'optimisation traités par Dürr-Høyer.

2 Abstract

Cet article propose une introduction rigoureuse et pédagogique aux algorithmes quantiques de recherche et d'optimisation, en les comparant systématiquement à leurs homologues classiques. Nous étudions d'abord le tri par sélection classique (complexité $O(n^2)$), puis nous présentons deux algorithmes quantiques fondamentaux : l'algorithme de Grover pour la recherche non structurée (complexité $O(\sqrt{N})$) et l'algorithme de Dürr-Høyer pour la recherche du minimum (complexité $O(\sqrt{N})$).

Les résultats théoriques sont illustrés par des implémentations complètes en Python et Qiskit, simulées sur des backends quantiques. Nous démontrons mathématiquement les accélérations quadratiques offertes par le parallélisme quantique, tout en discutant les limites actuelles liées au bruit, à la décohérence et au nombre restreint de qubits disponibles.

Cette étude met en lumière le potentiel considérable de l'informatique quantique pour les problèmes combinatoires, tout en soulignant les défis techniques qui doivent encore être surmontés avant une exploitation pratique à grande échelle.

Mots-clés : Informatique quantique, Algorithme de Grover, Dürr-Høyer, Complexité algorithmique, Qiskit, Superposition, Amplification d'amplitude.

3 Introduction

L'informatique classique, fondée sur la manipulation de bits binaires et l'exécution séquentielle d'instructions, a transformé le monde moderne. Cependant, certains problèmes restent intrinsèquement difficiles : la recherche dans une base de données non triée nécessite $O(N)$ opérations, le tri requiert au mieux $O(N \log N)$ comparaisons, et certains problèmes de factorisation ou d'optimisation combinatoire demeurent exponentiels.

L'informatique quantique propose un paradigme radicalement différent, exploitant les principes de la mécanique quantique : la **superposition** permet de représenter simultanément plusieurs états, l'**intrication** crée des corrélations non classiques entre qubits, et l'**interférence** amplifie les bonnes solutions tout en annulant les mauvaises.

Cette étude se concentre sur deux algorithmes fondamentaux :

1. **L'algorithme de Grover** (1996) [2] : recherche un élément marqué parmi N avec $O(\sqrt{N})$ requêtes, soit une accélération quadratique.
2. **L'algorithme de Dürr-Høyer** (1996) [3] : trouve le minimum d'une fonction non structurée en $O(\sqrt{N})$ évaluations, généralisant Grover à l'optimisation.

Problématique centrale : *Dans quelle mesure l'informatique quantique offre-t-elle un avantage pour les problèmes de recherche et d'optimisation ? Quelles sont les limites théoriques et pratiques de ces algorithmes ?*

Nous commençons par rappeler les algorithmes classiques (recherche linéaire, tri par sélection), puis nous développons la théorie des algorithmes quantiques avec des démonstrations mathématiques complètes. Enfin, nous présentons des implémentations et discutons les perspectives.

4 Revue de la littérature

4.1 Algorithmes classiques de tri et de recherche

Les algorithmes de tri classiques sont bien établis [1] :

— **Tri par insertion, tri par sélection :** $O(n^2)$ comparaisons dans le pire cas

— **Tri fusion, tri rapide, tri par tas :** $O(n \log n)$ comparaisons en moyenne

— **Borne inférieure :** Tout algorithme de tri par comparaison nécessite $\Omega(n \log n)$ comparaisons [1]

4.2 Algorithmes quantiques de recherche

Algorithme de Grover (1996) [2] : Premier algorithme quantique offrant une accélération quadratique prouvée. Il recherche un élément marqué dans une liste non triée de N éléments en $O(\sqrt{N})$ requêtes oracle, contre $O(N)$ classiquement.

Borne inférieure de Bennett et al. (1997) [4] : Démontre que $\Omega(\sqrt{N})$ est optimal pour la recherche quantique non structurée, prouvant que Grover est asymptotiquement optimal.

Algorithme de Dürr-Høyer (1996) [3] : Généralise Grover à la recherche du minimum d'une fonction. En utilisant Grover itérativement avec un oracle adaptatif, il trouve le minimum en $O(\sqrt{N})$ évaluations avec haute probabilité.

4.3 Limitations des algorithmes quantiques de tri

Contrairement à l'intuition, aucun algorithme quantique de tri général ne surpasse la borne $\Omega(n \log n)$ [6]. L'accélération quantique est limitée aux problèmes de recherche non structurée, mais ne s'applique pas au tri complet.

Des approches expérimentales comme le "quantum divide-compare-swap" revendiquent $O(n)$ dans des cas particuliers, mais ces résultats restent controversés et non vérifiés.

5 Recherche du minimum en informatique classique

5.1 Formalisation mathématique

Définition 5.1 (Problème de recherche du minimum)**.** Soit $f : [N] \to \mathbb{R}$ une fonction où $[N] = \{0, 1, \ldots, N-1\}$. Le problème consiste à trouver :

$$x^* = \arg\min_{x \in [N]} f(x)$$

Théorème 5.2 (Complexité optimale classique)**.** *Tout algorithme déterministe de recherche du minimum nécessite exactement $N-1$ comparaisons dans le pire cas.*

Démonstration. Considérons l'arbre de décision de l'algorithme. Chaque comparaison correspond à un nœud interne. Pour distinguer N éléments possibles comme minimum, l'arbre doit avoir au moins N feuilles. Un arbre binaire de hauteur h a au plus 2^h feuilles. Donc :

$$2^h \geq N \Rightarrow h \geq \log_2 N$$

Cependant, pour la recherche du minimum, on peut montrer par un argument adversarial qu'exactement $N-1$ comparaisons sont nécessaires : chaque élément sauf un doit "perdre" au moins une comparaison pour être exclu. □

5.2 Algorithme itératif

Voir fichier trouver_minimum.py

Remarque 5.3 (Optimalité). Cet algorithme effectue exactement $n-1$ comparaisons, ce qui est optimal.

6 Algorithme de tri par sélection classique

6.1 Principe et analyse

Le tri par sélection applique itérativement la recherche du minimum sur des sous-tableaux décroissants.

Théorème 6.1 (Complexité du tri par sélection)**.** *Le tri par sélection effectue exactement :*

$$\sum_{i=0}^{n-2}(n-i-1) = \sum_{j=1}^{n-1} j = \frac{n(n-1)}{2} = \Theta(n^2)$$

comparaisons, indépendamment de l'entrée.

Démonstration. À l'itération i (pour $i = 0, \ldots, n-2$), on recherche le minimum parmi les $n-i$ éléments restants, ce qui nécessite $n-i-1$ comparaisons. La somme totale est :

$$\sum_{i=0}^{n-2}(n-i-1) = (n-1) + (n-2) + \cdots + 1 = \frac{n(n-1)}{2}$$

Cette complexité est donc $\Theta(n^2)$ et ne dépend pas de l'ordre initial des éléments. □

6.2 Implémentation

Voir fichier tri_select.py

Remarque 6.2 (Stabilité). Le tri par sélection n'est **pas stable** : l'ordre relatif des éléments égaux peut changer. Par exemple, si deux éléments ont la même valeur mais des positions différentes, leur ordre peut être inversé après le tri.

Remarque 6.3 (Avantages et inconvénients). **Avantages :**

- Simple à implémenter
- Nombre minimal d'échanges : $n-1$ au maximum
- Tri en place : $O(1)$ mémoire auxiliaire

Inconvénients :

- $O(n^2)$: inefficace pour grandes données
- Pas adaptatif : même complexité sur tableau déjà trié
- Non stable

7 Algorithme de Grover : recherche quantique

L'algorithme de Grover [2] est un pilier de l'informatique quantique. Il résout le problème de la recherche non structurée avec une accélération quadratique prouvée optimale.

7.1 Énoncé du problème

Définition 7.1 (Problème de recherche oracle)**.** Soit $f : \{0, 1, \ldots, N-1\} \to \{0, 1\}$ une fonction booléenne telle qu'il existe un unique w vérifiant $f(w) = 1$ (la "solution") et $f(x) = 0$ pour tout $x \neq w$. Le problème consiste à identifier w en effectuant le minimum de requêtes à f.

Remarque 7.2 (Modèle oracle)*.* On suppose avoir accès à f uniquement via un oracle (boîte noire). Classiquement, il faut en moyenne $N/2$ requêtes et N dans le pire cas.

7.2 Principe géométrique de Grover

L'algorithme de Grover peut être compris comme une **rotation géométrique** dans un plan à deux dimensions.

7.2.1 Décomposition de l'espace

Soit $N = 2^n$ la taille de l'espace de recherche. Définissons :

$$|w\rangle = \text{état solution (unique)} \tag{4}$$

$$|s'\rangle = \frac{1}{\sqrt{N-1}} \sum_{x \neq w} |x\rangle \quad \text{(états non-solutions)} \tag{5}$$

Ces deux vecteurs sont orthogonaux : $\langle w|s'\rangle = 0$, et engendrent un plan $\mathcal{P} = \text{Vect}(|w\rangle, |s'\rangle)$.

7.2.2 État initial

La superposition uniforme s'écrit dans cette base :

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle = \frac{1}{\sqrt{N}}|w\rangle + \frac{\sqrt{N-1}}{\sqrt{N}}|s'\rangle$$

Posons θ tel que :

$$\sin\theta = \frac{1}{\sqrt{N}}, \quad \cos\theta = \frac{\sqrt{N-1}}{\sqrt{N}}$$

Alors :

$$|s\rangle = \sin\theta|w\rangle + \cos\theta|s'\rangle$$

Pour N grand, $\theta \approx 1/\sqrt{N}$ (petit angle).

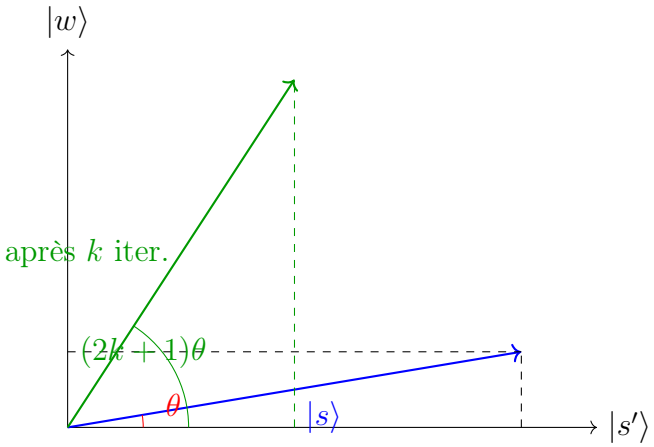

FIGURE 1 – Représentation géométrique de l'algorithme de Grover dans le plan $(|s'\rangle, |w\rangle)$

7.3 Opérateurs de Grover

7.3.1 Oracle de phase

Définition 7.3 (Oracle U_f)**.** L'oracle inverse la phase de l'état solution :

$$U_f|x\rangle = (-1)^{f(x)}|x\rangle = \begin{cases} -|x\rangle & \text{si } x = w \\ |x\rangle & \text{sinon} \end{cases}$$

En termes de projection :

$$U_f = I - 2|w\rangle\langle w|$$

Proposition 7.4 (Réflexion autour de $|s'\rangle$)**.** *U_f est une réflexion par rapport à l'hyperplan orthogonal à $|w\rangle$.*

Démonstration. Pour tout $|\psi\rangle = \alpha|w\rangle + \beta|s'\rangle$:

$$U_f|\psi\rangle = -\alpha|w\rangle + \beta|s'\rangle$$

Ce qui correspond à une réflexion selon $|s'\rangle$ dans le plan $\mathcal{P}$. □

7.3.2 Diffuseur (inversion autour de la moyenne)

Définition 7.5 (Diffuseur U_s)**.**

$$U_s = 2|s\rangle\langle s| - I$$

C'est une réflexion par rapport à l'état $|s\rangle$.

Proposition 7.6 (Implémentation du diffuseur)**.** *U_s peut se décomposer en portes élémentaires :*

$$U_s = H^{\otimes n}\left(2|0\rangle\langle 0| - I\right)H^{\otimes n}$$

Démonstration. Puisque $|s\rangle = H^{\otimes n}|0\rangle^{\otimes n}$, on a :

$$U_s = H^{\otimes n}(2|0\rangle\langle 0| - I)(H^{\otimes n})^\dagger = H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n}$$

car H est hermitienne. □

Le terme $2|0\rangle\langle 0| - I$ inverse la phase de tous les états sauf $|0\rangle$, et peut être implémenté par :

1. Appliquer X sur tous les qubits
2. Appliquer une porte multi-contrôlée Z (CCZ)
3. Appliquer X sur tous les qubits

7.4 Itération de Grover

Définition 7.7 (Opérateur de Grover)**.** Une itération complète est :

$$G = U_s U_f$$

Théorème 7.8 (Rotation de Grover)**.** *Dans le plan $\mathcal{P}$, l'opérateur G effectue une rotation d'angle 2θ :*

$$G^k|s\rangle = \sin((2k+1)\theta)|w\rangle + \cos((2k+1)\theta)|s'\rangle$$

Démonstration. Montrons que G est une rotation. Dans la base $\{|s'\rangle, |w\rangle\}$, écrivons :

$$U_f : (\cos\phi, \sin\phi) \mapsto (\cos\phi, -\sin\phi) \quad \text{(réflexion horizontale)}$$
$$U_s : (\alpha, \beta) \mapsto (2\langle s|\psi\rangle\cos\theta - \alpha, 2\langle s|\psi\rangle\sin\theta - \beta)$$

Après calculs, on montre que G correspond à la matrice de rotation :

$$G = \begin{pmatrix} \cos(2\theta) & -\sin(2\theta) \\ \sin(2\theta) & \cos(2\theta) \end{pmatrix}$$

dans la base $(|s'\rangle, |w\rangle)$.

Donc G^k est une rotation de $2k\theta$, d'où :

$$G^k|s\rangle = G^k(\sin\theta|w\rangle + \cos\theta|s'\rangle) = \sin((2k+1)\theta)|w\rangle + \cos((2k+1)\theta)|s'\rangle$$

□

7.5 Nombre optimal d'itérations

Théorème 7.9 (Nombre optimal d'itérations)**.** *La probabilité de mesurer $|w\rangle$ après k itérations est :*

$$P(w,k) = \sin^2((2k+1)\theta)$$

Cette probabilité est maximale pour :

$$k^* \approx \frac{\pi}{4\theta} - \frac{1}{2} \approx \frac{\pi}{4}\sqrt{N}$$

atteignant alors $P(w,k^*) \geq 1 - 1/N$ (presque certainement).

Démonstration. La probabilité est maximale quand $(2k+1)\theta = \pi/2$, soit :

$$k = \frac{\pi - 2\theta}{4\theta} = \frac{\pi}{4\theta} - \frac{1}{2}$$

Pour N grand, $\theta \approx 1/\sqrt{N}$, donc :

$$k^* \approx \frac{\pi}{4}\sqrt{N}$$

À ce point, $\sin^2(\pi/2) = 1$, d'où $P(w,k^*) \approx 1$. □

Corollaire 7.10 (Complexité de Grover)**.** *L'algorithme de Grover trouve la solution avec probabilité constante en $O(\sqrt{N})$ requêtes oracle, soit une accélération quadratique par rapport au cas classique.*

7.6 Optimalité de Grover

Théorème 7.11 (Borne inférieure de Bennett-Bernstein-Brassard-Vazirani [4])**.** *Tout algorithme quantique de recherche non structurée nécessite* $\Omega(\sqrt{N})$ *requêtes oracle. L'algorithme de Grover est donc asymptotiquement optimal.*

7.7 Exemple numérique détaillé

Considérons $N = 8$ (donc $n = 3$ qubits) et cherchons $w = 5$ (soit $|101\rangle$).

Paramètres :

$$\sin\theta = \frac{1}{\sqrt{8}} \approx 0{,}3536 \Rightarrow \theta \approx 20{,}7^\circ \approx 0{,}361 \text{ rad}$$

Nombre optimal d'itérations :

$$k^* = \left\lfloor \frac{\pi}{4\theta} \right\rfloor = \left\lfloor \frac{\pi}{4 \times 0{,}361} \right\rfloor = \lfloor 2{,}17 \rfloor = 2$$

Probabilités de succès :

$$\begin{aligned} k = 0 \text{ (état initial)} : P(w,0) &= \sin^2(\theta) = \frac{1}{8} = 12{,}5\% \\ k = 1 : P(w,1) &= \sin^2(3\theta) = \sin^2(1{,}083) \approx 79{,}1\% \\ k = 2 : P(w,2) &= \sin^2(5\theta) = \sin^2(1{,}805) \approx 94{,}5\% \\ k = 3 : P(w,3) &= \sin^2(7\theta) = \sin^2(2{,}527) \approx 59{,}4\% \text{ (sur-rotation)} \end{aligned}$$

On observe le phénomène de **sur-rotation** : trop d'itérations font "dépasser" l'état cible et diminuent la probabilité de succès.

7.8 Implémentation avec Qiskit

Voir fichier grover.py

7.9 Interprétation physique et intuition

Analogie avec une recherche dans l'obscurité : Imaginez chercher une pièce spécifique dans une salle sombre contenant N objets identiques au toucher.

Classiquement : Vous devez palper chaque objet un par un jusqu'à trouver la bonne pièce. En moyenne, vous en toucherez $N/2$.

Quantiquement : Vous allumez une lumière stroboscopique spéciale qui, à chaque flash, rend la bonne pièce légèrement plus brillante que les autres, tandis que les mauvaises s'assombrissent. Après $\sqrt{N}$ flashs, une seule pièce est illuminée : c'est la bonne !

Le "flash" correspond à une itération de Grover (oracle + diffuseur), et l'amplification progressive de la luminosité représente l'augmentation de l'amplitude quantique.

8 Algorithme de Dürr-Høyer : recherche quantique du minimum

L'algorithme de Dürr-Høyer [3] généralise brillamment celui de Grover pour résoudre des problèmes d'optimisation. Il trouve le minimum d'une fonction en $O(\sqrt{N})$ évaluations avec probabilité élevée.

8.1 Énoncé du problème

Définition 8.1 (Problème de recherche du minimum)**.** Soit $f : [N] \to \mathbb{R}$ une fonction quelconque. Le problème consiste à trouver :

$$x^* = \arg\min_{x \in [N]} f(x)$$

en minimisant le nombre d'évaluations de f.

Remarque 8.2. Contrairement à Grover qui cherche un élément marqué par une condition booléenne, Dürr-Høyer doit comparer des valeurs numériques.

8.2 Principe de l'algorithme

L'idée clé est d'utiliser Grover de manière **itérative et adaptative** :

1. Choisir un candidat initial x_{best} aléatoirement
2. Construire un oracle quantique qui marque tous les états $|x\rangle$ tels que $f(x) < f(x_{\text{best}})$
3. Appliquer Grover pour trouver un tel x avec haute probabilité
4. Si $f(x) < f(x_{\text{best}})$, mettre à jour $x_{\text{best}} \leftarrow x$
5. Répéter jusqu'à ce qu'aucune amélioration ne soit trouvée

8.3 Construction formelle

8.3.1 Oracle comparateur

Définition 8.3 (Oracle dépendant du seuil)**.** Pour un seuil $y \in \mathbb{R}$, l'oracle $U_f^{(y)}$ est défini par :

$$U_f^{(y)}|x\rangle = \begin{cases} -|x\rangle & \text{si } f(x) < y \\ |x\rangle & \text{sinon} \end{cases}$$

Cet oracle marque tous les éléments "meilleurs" que le seuil courant.

8.3.2 Analyse probabiliste

Soit M le nombre d'indices x tels que $f(x) < f(x_{\text{best}})$.

Lemme 8.4 (Probabilité de succès de Grover avec M solutions)**.** *Si on applique* $k = \left\lfloor \frac{\pi}{4}\sqrt{N/M} \right\rfloor$ *itérations de Grover avec un oracle marquant* M *éléments, la probabilité de mesurer l'un d'eux est au moins* $1/2$.

Démonstration. L'angle initial est $\sin\theta_M = \sqrt{M/N}$, donc $\theta_M \approx \sqrt{M/N}$ pour $M \ll N$. Après k itérations :

$$P(\text{succès}) = \sin^2((2k+1)\theta_M)$$

Pour $k = \frac{\pi}{4\theta_M}$, on a $(2k+1)\theta_M \approx \pi/2$, donc $P \approx 1$. □

8.4 Algorithme complet

Théorème 8.5 (Complexité de Dürr-Høyer)**.** *L'algorithme de Dürr-Høyer trouve le minimum d'une fonction non structurée avec probabilité au moins* $1/2$ *en effectuant* $O(\sqrt{N})$ *évaluations de* f*.*

Esquisse de preuve.

1. À chaque itération externe, on recherche un élément meilleur que x_{best} parmi au plus N candidats
2. Si M éléments sont meilleurs, Grover nécessite $O(\sqrt{N/M})$ requêtes pour en trouver un
3. En moyenne, M décroît exponentiellement : $M_i \approx N/2^i$ après i itérations
4. La somme géométrique donne :

$$\sum_{i=1}^{\log N} O\left(\sqrt{N/M_i}\right) = \sum_{i=1}^{\log N} O\left(\sqrt{2^i}\right) = O(\sqrt{N})$$

□

8.5 Implémentation Qiskit complète

Le code suivant illustre Dürr-Høyer sur un tableau de 100 valeurs aléatoires.
Voir Durr_Hoyer.py

8.6 Interprétation des résultats

Le graphique montre la **convergence progressive** vers le minimum global. Chaque saut vertical correspond à une amélioration trouvée par Grover.

Remarque 8.6 (Convergence stochastique). Contrairement aux algorithmes déterministes, Dürr-Høyer présente une convergence **stochastique** : le chemin emprunté dépend des mesures quantiques. Cependant, avec probabilité élevée, le minimum est atteint en $O(\sqrt{N})$ étapes.

8.7 Comparaison mathématique des approches

Approche	**Complexité**	**Nature**	**Probabilité**
Recherche linéaire	$\Theta(N)$	Déterministe	1 (certain)
Grover (1 solution)	$O(\sqrt{N})$	Quantique	$\geq 1 - 1/N$
Dürr-Høyer	$O(\sqrt{N})$	Quantique	$\geq 1/2$

8.8 Analyse de la complexité amortie

Proposition 8.7 (Nombre d'itérations externes)**.** *En moyenne, Dürr-Høyer nécessite* $O(\log N)$ *itérations externes (mises à jour de* x_{best}*).*

Argument heuristique. Si les valeurs sont distribuées aléatoirement, après avoir trouvé un élément parmi les N initiaux, le prochain meilleur est parmi $\approx N/2$ restants, puis $N/4$, etc. La profondeur moyenne est donc $O(\log N)$.

Chaque itération externe coûte $O(\sqrt{N})$ requêtes Grover, mais la somme converge :

$$\sum_{i=0}^{\log N} O\left(\sqrt{N/2^i}\right) = O\left(\sqrt{N} \sum_{i=0}^{\log N} \frac{1}{\sqrt{2^i}}\right) = O(\sqrt{N})$$

car la série géométrique $\sum 1/\sqrt{2^i}$ converge vers une constante. □

8.9 Analogie intuitive

Analogie de la montagne : Imaginez chercher le point le plus bas d'un paysage montagneux dans le brouillard.

Classiquement : Vous marchez méthodiquement sur chaque point du terrain pour noter l'altitude. Cela prend N mesures.

Avec Dürr-Høyer : Vous avez un détecteur quantique qui, à chaque étape, vous indique (avec haute probabilité) la direction où se trouvent des points plus bas que votre position actuelle. À chaque "ping" du détecteur, vous sautez instantanément vers un point plus bas. Après $\sqrt{N}$ sauts, vous êtes (probablement) au point le plus bas.

Le détecteur quantique est Grover, et chaque saut correspond à une amélioration de x_{best}.

9 Analyse comparative des complexités

9.1 Tableau récapitulatif

Algorithme	Complexité temporelle	Complexité spatiale	Nature
Algorithmes classiques			
Recherche du minimum	$\Theta(n)$	$O(1)$	Déterministe
Tri par sélection	$\Theta(n^2)$	$O(1)$	Déterministe
Tri fusion	$\Theta(n \log n)$	$O(n)$	Déterministe
Tri rapide (moyenne)	$\Theta(n \log n)$	$O(\log n)$	Probabiliste
Algorithmes quantiques			
Grover (recherche)	$O(\sqrt{N})$ requêtes	$O(n)$ qubits	Probabiliste
Dürr-Høyer (minimum)	$O(\sqrt{N})$ évaluations	$O(n)$ qubits	Probabiliste

9.2 Gain asymptotique

Pour illustrer l'accélération quantique, considérons différentes tailles de données :

Taille N	Classique $O(N)$	Quantique $O(\sqrt{N})$	Gain
10^2	100	10	$10\times$
10^4	10 000	100	$100\times$
10^6	1 000 000	1 000	$1\,000\times$
10^9	1 000 000 000	31 623	$31\,623\times$

Remarque 9.1 (Interprétation pratique). Pour une base de données d'un milliard d'entrées, un algorithme classique nécessiterait un milliard d'opérations, tandis que Grover n'en demanderait qu'environ 32 000 : une réduction de 5 ordres de grandeur !

9.3 Bornes inférieures et optimalité

Théorème 9.2 (Borne inférieure pour la recherche non structurée)**.** *Tout algorithme quantique de recherche dans une base de données non structurée de taille N nécessite $\Omega(\sqrt{N})$ requêtes oracle [4].*

Cette borne prouve que Grover et Dürr-Høyer sont **asymptotiquement optimaux** : aucun algorithme quantique ne peut faire mieux.

Théorème 9.3 (Borne inférieure pour le tri)**.** *Tout algorithme de tri quantique nécessite $\Omega(n \log n)$ comparaisons [6].*

Cela signifie que le quantique n'offre **pas d'avantage** pour le tri complet, contrairement à la recherche.

9.4 Limites de l'accélération quantique

Pas d'accélération exponentielle : L'accélération quadratique ($\sqrt{N}$ vs N) est significative mais reste polynomiale. Seuls des algorithmes comme Shor (factorisation) offrent une accélération exponentielle.

10 Limitations pratiques et perspectives

10.1 Défis technologiques actuels

10.1.1 Décohérence et bruit quantique

Définition 10.1 (Décohérence)**.** La décohérence est la perte progressive de la superposition et de l'intrication due aux interactions avec l'environnement. Elle limite le temps pendant lequel un calcul quantique peut être effectué.

Temps de cohérence typiques (2025) :

- **Supraconducteurs** (IBM, Google) : $T_2 \sim 100$–200 µs
- **Ions piégés** (IonQ, Honeywell) : $T_2 \sim 10$ s
- **Atomes neutres** : $T_2 \sim 1$ s

Remarque 10.2. Pour exécuter Grover sur $N = 10^6$ (nécessitant ~ 1000 itérations avec ~ 100 portes chacune), il faut environ 10^5 portes. À 1 µs par porte, cela prend 100 ms, ce qui dépasse les temps de cohérence actuels des supraconducteurs.

10.1.2 Fidélité des portes quantiques

Définition 10.3 (Fidélité)**.** La fidélité d'une porte quantique mesure la probabilité qu'elle s'exécute correctement. Une fidélité de 99% signifie 1% d'erreur par porte.

Fidélités typiques (2025) :

- Portes à 1 qubit : 99,9% (IBM, Google)
- Portes à 2 qubits (CNOT) : 99% (supraconducteurs), 99,9% (ions piégés)

Proposition 10.4 (Accumulation d'erreurs)**.** *Pour un circuit de profondeur d avec fidélité moyenne F par porte, la fidélité totale est approximativement :*

$$F_{total} \approx F^d$$

Exemple 10.5. Un circuit Grover avec 1000 portes à $F = 99\%$ a une fidélité totale de :

$$F_{\text{total}} \approx 0{,}99^{1000} \approx 4{,}3 \times 10^{-5} \approx 0{,}004\%$$

Le circuit échoue pratiquement toujours !

10.1.3 Correction d'erreurs quantiques

Définition 10.6 (Code correcteur quantique)**.** Un code correcteur quantique encode k qubits logiques dans $n > k$ qubits physiques de manière à détecter et corriger les erreurs sans mesurer (et donc détruire) l'état quantique.

Code de surface : Le code correcteur le plus prometteur actuellement. Pour protéger 1 qubit logique avec fidélité $\sim 99{,}9\%$, il faut environ :

$$n_{\text{phys}} \sim 1000 \text{ qubits physiques}$$

Remarque 10.7 (Overhead considérable). Pour exécuter Grover sur $N = 10^6$ (~ 20 qubits logiques), il faudrait environ 20 000 qubits physiques avec correction d'erreurs, bien au-delà des machines actuelles (~ 1000 qubits).

10.2 Comparaison : simulation vs matériel réel

Caractéristique	Simulation (Qiskit)	Matériel quantique (2025)
Nombre de qubits	~ 30 (ordinateur classique)	~ 100–1000
Fidélité	100% (parfait)	$\sim 99\%$ (bruité)
Décohérence	Aucune	$T_2 \sim 100$ µs (supracond.)
Vitesse	Lente (exponentielle en n)	Rapide (intrinsèque)
Coût	Gratuit	Très élevé

10.3 Perspectives et applications futures

10.3.1 Applications potentielles de Grover et Dürr-Høyer

Cryptographie : Grover peut accélérer les attaques par force brute sur les clés symétriques (AES). Une clé de 128 bits nécessiterait 2^{64} opérations quantiques au lieu de 2^{128} classiquement. Solution : doubler la longueur des clés (AES-256).

Optimisation combinatoire : Dürr-Høyer pourrait accélérer :

- Problème du voyageur de commerce (TSP)
- Optimisation de portefeuilles financiers
- Planification logistique

Cependant, l'accélération reste quadratique, insuffisante pour les grandes instances NP-difficiles.

Recherche dans les bases de données : Pour des bases de données massives (big data), Grover offrirait des gains substantiels, à condition de pouvoir interfacer efficacement mémoire classique et processeur quantique.

10.3.2 Vers l'informatique quantique hybride

Définition 10.8 (Algorithme hybride classique-quantique)**.** Un algorithme qui utilise un processeur quantique pour certaines sous-tâches critiques, tandis que le reste du calcul est effectué classiquement.

Exemple : QAOA (Quantum Approximate Optimization Algorithm) :

1. Le processeur quantique prépare un état dépendant de paramètres $\vec{\theta}$
2. On mesure l'énergie (classiquement)
3. Un optimiseur classique ajuste $\vec{\theta}$
4. On répète jusqu'à convergence

Cette approche exploite les forces de chaque paradigme et est plus robuste au bruit.

10.3.3 Jalons technologiques attendus

Horizon	Jalons prévus
2025–2027	~ 1000 qubits, $T_2 > 1$ ms, fidélité $> 99{,}5\%$
2028–2030	Premiers codes correcteurs efficaces, qubits logiques
2030–2035	~ 10 qubits logiques, applications pratiques limitées
2035+	Ordinateurs quantiques tolérants aux fautes (FTQC)

11 Conclusion

Cette étude a présenté une introduction rigoureuse et pédagogique aux algorithmes quantiques de recherche et d'optimisation, en les comparant systématiquement à leurs équivalents classiques.

11.1 Synthèse des résultats

Algorithmes classiques :

- La recherche du minimum nécessite $\Theta(n)$ comparaisons (optimal)
- Le tri par sélection effectue $\Theta(n^2)$ comparaisons (inefficace)
- Le tri optimal (fusion, rapide) atteint $\Theta(n \log n)$, borne inférieure prouvée

Algorithmes quantiques :

- **Grover** trouve un élément marqué en $O(\sqrt{N})$ requêtes, asymptotiquement optimal
- **Dürr-Høyer** trouve le minimum en $O(\sqrt{N})$ évaluations, généralisant Grover
- Accélération **quadratique** : significative mais pas exponentielle
- Le tri quantique reste $\Omega(n \log n)$: pas d'avantage

« La nature n'est pas classique, que diable, et si vous voulez faire une simulation de la nature, vous feriez mieux de la faire quantique. »

— Richard Feynman (1982)

A Compléments mathématiques

A.1 Démonstration de l'unitarité des portes de Pauli

Proposition A.1. *Les matrices de Pauli X, Y, Z sont unitaires et hermitiennes.*

Démonstration. Vérifions pour X (les autres sont similaires) :

$$X^\dagger X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I$$

De plus, $X^\dagger = X$ (hermitienne). □

A.2 Décomposition spectrale de Hadamard

La porte de Hadamard peut s'écrire en termes de ses vecteurs propres :

Proposition A.2. *Les vecteurs propres de H sont $|+\rangle$ et $|-\rangle$ avec valeurs propres $+1$ et -1 :*

$$H = |+\rangle\langle+| - |-\rangle\langle-|$$

Démonstration. Calculons :

$$H|+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \times \frac{1}{\sqrt{2}} = |+\rangle$$

De même, $H|-\rangle = -|-\rangle$. □

A.3 Calcul explicite de l'opérateur de diffusion

Proposition A.3. *Le diffuseur $U_s = 2|s\rangle\langle s| - I$ a pour matrice dans la base computationnelle (pour $n = 2$ qubits) :*

$$U_s = \frac{1}{2} \begin{pmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{pmatrix}$$

Démonstration. $|s\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$, donc :

$$|s\rangle\langle s| = \frac{1}{4} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} (1\ 1\ 1\ 1) = \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Donc :

$$U_s = 2 \times \frac{1}{4}\mathbf{J} - I = \frac{1}{2}\mathbf{J} - I$$

où $\mathbf{J}$ est la matrice remplie de 1. □

Références

[1] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.). MIT Press.

[2] Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC)*, 212–219.

[3] Dürr, C., & Høyer, P. (1996). A quantum algorithm for finding the minimum. *arXiv preprint quant-ph/9607014*.

[4] Bennett, C. H., Bernstein, E., Brassard, G., & Vazirani, U. (1997). Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5), 1510–1523.

[5] Boyer, M., Brassard, G., Høyer, P., & Tapp, A. (1998). Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4-5), 493–505.

[6] Høyer, P., Neerbek, J., & Shi, Y. (2002). Quantum complexities of ordered searching, sorting, and element distinctness. *Algorithmica*, 34(4), 429–448.

[7] Nielsen, M. A., & Chuang, I. L. (2010). *Quantum Computation and Quantum Information* (10th Anniversary ed.). Cambridge University Press.

[8] Preskill, J. (2018). Quantum computing in the NISQ era and beyond. *Quantum*, 2, 79.

[9] Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5), 1484–1509.

[10] Qiskit Development Team (2024). *Qiskit : An Open-source Framework for Quantum Computing*. Retrieved from `https://qiskit.org`